

RAUL SIDNEI WAZLAWICK

# HISTÓRIA *da* COMPUTAÇÃO



LTC

SÉRIE  
**SBC**  
SOCIEDADE  
BRASILEIRA DE  
COMPUTAÇÃO

# **História da Computação**



RAUL SIDNEI WAZLAWICK

## **História da Computação**

© 2016 Elsevier Editora Ltda.

Todos os direitos reservados e protegidos pela Lei n. 9.610, de 19/02/1998.  
Nenhuma parte deste livro, sem autorização prévia por escrito da editora, poderá ser reproduzida ou transmitida sejam quais forem os meios empregados:  
eletrônicos, mecânicos, fotográficos, gravação ou quaisquer outros.

*Copidesque: Vanessa Raposo Revisão tipográfica: Tássia Hallais* *Editoração Eletrônica: SBNigri Artes e Textos Ltda.*

*Produção de ePub: SBNigri Artes e Textos Ltda.*

Elsevier Editora Ltda.

Conhecimento sem Fronteiras

Rua Sete de Setembro, 111 – 16º andar 20050-006 – Centro – Rio de Janeiro – RJ – Brasil

Rua Quintana, 753 – 8º andar 04569-011 – Brooklin – São Paulo – SP – Brasil

Serviço de Atendimento ao Cliente 0800-0265340

sac@elsevier.com.br

ISBN: 978-85-352-8545-1

ISBN versão eletrônica: 978-85-352-8546-8

Consulte nosso catálogo completo, os últimos lançamentos e os serviços exclusivos no site [www.elsevier.com.br](http://www.elsevier.com.br)

**Nota:**

Muito zelo e técnica foram empregados na edição desta obra. No entanto, podem ocorrer erros de digitação, impressão ou dúvida conceitual. Em qualquer das hipóteses, solicitamos a comunicação ao nosso serviço de Atendimento ao Cliente para que possamos esclarecer ou encaminhar a questão.

Para todos os efeitos legais, nem a editora, nem os autores, nem os editores, nem os tradutores, nem os revisores ou colaboradores, assumem qualquer responsabilidade por qualquer efeito danoso e/ou malefício a pessoas ou propriedades envolvendo responsabilidade, negligência etc. de produtos, ou advindos de qualquer uso ou emprego de quaisquer métodos, produtos, instruções ou ideias contidos no material aqui publicado.

CIP-BRASIL. CATALOGAÇÃO-NA-FONTE  
SINDICATO NACIONAL DOS EDITORES DE LIVROS, RJ

W372h Wazlawick, Raul Sidnei  
História da computação / Raul Sidnei Wazlawick. – 1. ed. – Rio de Janeiro: Elsevier, 2016.  
il. ; 24 cm.

ISBN 978-85-352-8545-1

1. Informática. 2. Informática – História. 3. Computadores – História. I. Título.

16-33277

CDD: 004  
CDU: 004

# Prefácio

Em primeiro lugar é necessário esclarecer que sou um cientista da computação; não sou historiador nem jornalista. Por outro lado, acho que posso considerar que sou também um “contador de histórias”, no bom sentido. E foi esse o tom que tentei dar a este livro, que é um apanhado em ordem mais ou menos cronológica de uma série de pequenas histórias que foram construindo a grande História da computação ao longo dos séculos.

Este livro atende àquelas pessoas que querem ter uma visão do todo, sem se perderem em detalhes. Ele não enfatiza algumas poucas pessoas ou empresas, por mais importantes que tenham sido, em detrimento de outras que também contribuíram para essa história. Mais importante de tudo, procurei relatar os fatos interessantes, curiosos e até engraçados que ocorreram, sempre que possível.

Outro diferencial deste livro em relação a outros do tipo está no fato de que, sempre que possível, tento explicar de forma simples e clara o funcionamento do princípio sobre o qual estou apresentando dados históricos. Creio que de nada adiantaria, por exemplo, falar que a comutação de pacotes foi inventada por tal pessoa em tal

data sem explicar minimamente ao leitor o que é “comutação de pacotes”.

Acredito que este livro será bastante interessante para o aluno de computação ou áreas afins que conhece muitos conceitos, mas não sabe como eles se desenvolveram ao longo do tempo. Os professores e profissionais mais experientes poderão também descobrir neste livro que alguns fatos de que ouviram falar ou que aprenderam anos atrás foram revistos. Por exemplo, diferente do que era propagado nas décadas passadas, oficialmente não se considera mais Alexander Graham Bell o inventor do telefone, o ENIAC não foi o primeiro computador eletrônico, assim como o UNIVAC não foi o primeiro computador comercializado.

Toda a pesquisa para este livro foi realizada sem que uma única folha de papel fosse gasta. As fontes são integralmente eletrônicas, embora nem todas disponíveis gratuitamente. Usei bastante a Wikipédia, pois ela contém, em geral, excelentes referências e pontos de vista abrangentes; mas utilizei também artigos científicos acessíveis pelo Portal Capes, livros eletrônicos, gratuitos ou não, artigos de jornais disponíveis em meio eletrônico, muitos dos quais são registros da própria época em que os fatos históricos ocorreram. Usei blogs, especialmente nos casos em que são de responsabilidade da pessoa que criou o conceito analisado em questão, para obter uma informação em primeira mão. Usei também extensivamente documentos históricos originais, os quais, com muita paciência e prazer, estudei para poder falar sem intermediários sobre minhas impressões do trabalho de pioneiros como Ada Lovelace, Claude Shannon, Vannevar Bush e Ivan Sutherland. Tivesse escrito este livro nos anos em que esses documentos históricos ainda não estavam disponíveis na Internet, eu possivelmente não teria acesso a eles e teria que escrever baseado em fontes secundárias, ou seja, pessoas que escreveram sobre tais autores, com o viés de sempre.

Quanto às imagens, todas são licenciadas pelo projeto Wikimedia Commons, ou são de domínio público, ou seu uso foi expressamente

autorizado pelos autores especialmente para a utilização nesta obra. Este livro não deve ser citado como fonte original de nenhuma dessas imagens a não ser aquelas para as quais não constam créditos, porque estas eu mesmo desenhei.

Não inventei nenhuma das histórias contadas aqui. Assim, o leitor vai encontrar algumas delas contadas com outras palavras em sites, artigos e livros. Minha contribuição pessoal está na forma da organização, no destaque aos fatos curiosos e na escolha e correlação entre os diferentes fatos, mas os fatos são sempre os mesmos.

Acredito que o livro seja também bastante interessante para o leitor leigo, ou seja, aquele que não é profissional da área de computação, mas está curioso para saber como a humanidade chegou ao ponto em que onde se encontra em relação à tecnologia. Esse leitor poderá não só aprender sobre os fatos históricos, mas também sobre o funcionamento básico dessas tecnologias. Acredito que este livro possa ser usado não só para ensinar História da Computação, mas também como uma introdução aos Fundamentos da Computação. Falaremos um pouco sobre lógica, aritmética binária, funcionamento de calculadoras mecânicas, circuitos etc., mas sempre de forma que qualquer pessoa com instrução média possa entender.

Enfim, desejo uma boa leitura a todos! Se gostar do livro visite nosso site em [www.facebook.com/HistoriaDaComputacao](http://www.facebook.com/HistoriaDaComputacao). Sempre que possível contarei novas histórias por lá.

**Raul Sidnei Wazlawick**

Florianópolis, Maio de 2016

# O Autor

**RAUL SIDNEI WAZLAWICK** nasceu em 1967. Seu primeiro contato com computadores foi através dos filmes *Jornada nas estrelas* com o Capitão Kirk, por volta de 1974, e das histórias do Professor Pardal nas revistinhas da Disney, o qual volta e meia construía cérebros eletrônicos que enlouqueciam.

Em 1979, um de seus colegas da sexta série ganhou um telejogo com *Pong*. A engenhoca era fascinante, mas muito cara. Por meses ele tentou convencer seu pai a comprar um telejogo ou pelo menos um kit de montar, que seria mais barato, mas os tempos eram difíceis e não foi possível. Eventualmente, conseguia ir a um arcade jogar *F-1* ou *Space Invaders*.

Porém, ali nasceu a paixão pela eletrônica. Os anos seguintes foram dedicados à construção de pequenos circuitos, como um grilo eletrônico e um pequeno piano, com capacitores, transistores etc. Nessa época, revistas de eletrônica apresentando projetos que podiam ser montados em casa com peças que se comprava facilmente eram populares.

O primeiro computador de verdade que usou foi uma calculadora programável HP que pertencia à sua tia, Eda (Sabka) Kunde, que era analista de sistemas. Isso deve ter sido no final dos anos 1970

ou início dos anos 1980. Era fascinante ver os números piscando rapidamente na tela enquanto a calculadora fazia repetidos cálculos aproximando funções que ele não sabia para que serviam.

Gostava muito de jogos e chegou a criar alguns misturando *War II*, *Volta ao Mundo*, *Banco Imobiliário*, dentre outros. Naquela época, apenas cartolina e lápis de cor eram os materiais disponíveis. Ainda assim ele sonhava em um dia programar esses jogos em computadores.

Durante a terceira série do então Segundo Grau, em 1984, outro colega, que depois também seria colega no curso de graduação, Nazareno Philippi Lehmkul, lhe ensinou os princípios de programação em BASIC. Ninguém tinha computador, mas faziam programas que talvez até funcionassem. Ele ainda tem um caderno com vários desses programas, a maioria jogos, no estilo *Space Invaders*.

Ainda neste ano ganhou seu primeiro computador pessoal, um TK83, com 2 kB de memória. Infelizmente seus programas com centenas de linhas nem cabiam na máquina: lá pela vigésima linha a memória se esgotava!

Colocou as mãos em um minicomputador Edisa na empresa onde trabalhava seu primo Gilberto Jandt, mas para sua frustração o dialeto BASIC daquela máquina era diferente do que ele conhecia. Com ajuda do primo, entretanto, conseguiu implementar um jogo de golfe na máquina; pena que a trajetória da bola era plotada de cabeça para baixo devido a um sistema de coordenadas diferente.

Durante 1984 estudou para o vestibular – Ciência da Computação, obviamente – e aproveitava para usar seu pequeno TK83 para programar e plotar funções matemáticas como seno, tangente etc. Apesar de ter feito dezenas desses programas, até hoje não sabe o que é uma cotangente hiperbólica.

Passou no vestibular e iniciou o curso em 1985. Para sua frustração, a universidade não contava com nenhum computador pessoal, apenas uma máquina de grande porte IBM 4341 com quatro

terminais acessíveis para todos os 400 alunos do curso de computação, um dos quais estava sempre com defeito. Assim, era normal naqueles tempos passar as madrugadas na universidade esperando por uma vaga no terminal para implementar e rodar os programas exigidos pelos professores.

Conseguiu trocar o TK83 por um CP200 usado. Com 16 kB de memória era bem mais fácil programar. Criou alguns jogos de cabeça, como o da cobrinha que crescia à medida que comia ratos na tela.

Em 1986, entrou para o laboratório EDUGRAF, onde, trabalhando sob orientação do Professor Luiz Fernando Bier Melgarejo, aprendeu Pascal, Smalltalk e Graforth. Ali teve contato com os primeiros computadores COBRA 305 e 210, Apple II e finalmente Itautec PC. Seu projeto de final de curso foi um ambiente de programação ao estilo Forth, com cerca de 8.000 comandos em linguagem de máquina. Era um sistema operacional, editor de texto e ambiente de programação com uma linguagem capaz de concorrência e características de orientação a objetos.

Fez mestrado em Porto Alegre, na UFRGS, terminando em 1990 e defendendo sua tese em 1991, quando já estava cursando doutorado na UFSC em Engenharia de Produção. Esse doutorado foi concluído com louvor em 1993, quando apresentou uma tese sobre aplicação das teorias epistemológicas de Jean Piaget à aprendizagem de máquina.

Ainda em 1992, fez concurso e foi contratado como professor assistente da UFSC. Trabalhou com inteligência artificial, realidade virtual, engenharia de software, informática na educação e muitas outras áreas, mas sua preferência sempre foi trabalhar com sistemas orientados a objetos. Seu primeiro livro, hoje na 3<sup>a</sup> edição, foi voltado a esse assunto. Publicado pela primeira vez em 2004 e lançado nos Estados Unidos pela Elsevier em 2014, sua 3<sup>a</sup> edição hoje se chama *Análise e projeto orientados a objetos para sistemas de informação: Modelagem com UML, OCL e IFML*.

Seu segundo livro foi *Metodologia de pesquisa para ciência da computação*, de 2009, hoje na 2<sup>a</sup> edição, e o terceiro foi *Engenharia de software: Conceitos e práticas*, de 2013, todos publicados no Brasil pela Elsevier.

Coordenou vários eventos da Sociedade Brasileira de Computação, como o SBIE, SBES, SBBD, WIE, SBSI, SVR e CSBC. Foi membro do Conselho da SBC por 10 anos. Durante 11 anos representou o Brasil junto ao comitê técnico 3 da IFIP (Educação) e foi coordenador do grupo de trabalho 3.2 (Educação Superior) por dois anos.

É professor titular da UFSC desde 2015 e atualmente supervisiona o projeto e-SUS do Laboratório bridge\_, que está informatizando a saúde básica no Brasil a pedido do Ministério da Saúde, com sistemas instalados em mais de 40 mil postos e vários outros programas atendidos. Com uma equipe de quase 100 pessoas desenvolvendo software, esse projeto tem sido também um excelente laboratório para aprendizagem de engenharia de software por parte de seus alunos na UFSC, além de ser um dos maiores sistemas de informação na área de saúde hoje existente no mundo.

Em 2012 decidiu escrever uma trilogia de livros para a disciplina de Informática e Sociedade. Este, *História da computação*, é o primeiro. Espera concluir um livro sobre ética e legislação em informática e outro sobre planos de negócio em informática assim que possível.

# Sumário

## **PARTE I - Pedra, Madeira e Ideias**

- [1.1 Contagem com Dedos](#)
- [1.2 Vara de Contagem – c. 20.000 a.C.](#)
- [1.3 Ábaco – c. 2.700 a. C.](#)
- [1.4 Silogismos – c. 400 a.C.](#)
- [1.5 Astrolábio Planisférico – c. 150 a.C.](#)
- [1.6 Mecanismo de Anticítera – c. 125 a.C.](#)
- [1.7 O Carro Programável de Heron de Alexandria – 60 d.C.](#)
- [1.8 O Zero – 628](#)
- [1.9 Relógio Mecânico – 724](#)
- [1.10 Algoritmo – 825](#)
- [1.11 Música Mecânica – 850](#)
- [1.12 Criptografia – c. 850](#)
- [1.13 Autômatos de Al-Jazari – 1206](#)
- [1.14 Ars Magna de Ramon Llull – 1305](#)
- [1.15 A Controversa “Calculadora” de Leonardo da Vinci – 1499](#)
- [1.16 Logaritmos – 1614](#)
- [1.17 Bastões de Napier – 1617](#)
- [1.18 Régua de Cálculo – 1622](#)
- [1.19. Até Aqui...](#)

## **PARTE II - Surgimento das Calculadoras Mecânicas**

- [2.1 Relógio Calculador de Schickard – 1623](#)
- [2.2 Pascalina – 1642](#)
- [2.3 Leviatã de Thomas Hobbes – 1651](#)
- [2.4 O Método Llullístico de Athanasius Kircher – 1669](#)
- [2.5 Calculadora de Samuel Morland – 1673](#)
- [2.6 Sistema Binário – 1679](#)
- [2.7 Contador Mecânico de Leibniz – 1694](#)
- [2.8 Computadores Humanos – 1758](#)
- [2.9 Calculadora e Demonstrador de Charles Stanhope – 1775](#)
- [2.10 Calculadora e Máquina Diferencial de Johann H. Müller – 1783](#)
- [2.11 O Tear de Jacquard – 1801](#)
- [2.12 Arithmometer – 1820](#)
- [2.13 Até Aqui...](#)

## **PARTE III - De Babbage a Hollerith**

- [3.1 Máquina Diferencial de Babbage – 1821](#)
- [3.2 Semyon Korsakov e o Registro de Informação em Cartões Perfurados – 1832](#)
- [3.3 Calculadora com Teclado de Torchi – 1834](#)
- [3.4 Relê Eletromecânico de Joseph Henry – 1835](#)
- [3.5 Máquina Analítica de Babbage – 1837](#)
- [3.6 A Primeira Programadora: Ada Lovelace – 1842](#)
- [3.7 A Máquina Relacional de Alfred Smee – 1851](#)
- [3.8 Telefone de Antonio Meucci – 1856](#)
- [3.9 Álgebra Booleana – 1854](#)
- [3.10 Calculadora de Coluna Única de Caroline Winter – 1859](#)
- [3.11 A Máquina de Escrever – 1867](#)
- [3.12 O Piano Lógico de Jevons – 1869](#)
- [3.13 Multiplicação Direta de Edmund Barbour – 1872](#)
- [3.14 Lógica de Predicados de Frege – 1879](#)
- [3.15 NE e NOU – 1880](#)

[3.16 A Calculadora Brasileira de Azevedo Coutinho – 1884](#)

[3.17 Máquina Tabuladora de Hollerith – 1884](#)

[3.18 Até Aqui...](#)

## **PARTE IV - Surgimento da Eletrônica**

[4.1 Burroughs – 1886](#)

[4.2 Calculadora Eletromagnética de Charles Weiss – 1886](#)

[4.3 Máquina Lógica de Allan Marquand – 1886](#)

[4.4 Telautograph – 1888](#)

[4.5 Gravador Magnético de Oberlin Smith – 1888](#)

[4.6 Calculadoras Analógicas de Leonardo Torres y Quevedo – 1893](#)

[4.7 TMC: Tabulating Machine Company – 1896](#)

[4.8 O Monitor CRT – 1897](#)

[4.9 Telegraphone – 1898](#)

[4.10 Autarith de Alexander Rechnitzer – 1900](#)

[4.11 Máquina Analítica de Percy Ludgate – 1903](#)

[4.12 Válvula-Diodo de John Fleming – 1904](#)

[4.13 Cristal Líquido \(LCD\) – 1904](#)

[4.14 CTR: Computing-Tabulating-Recording Company – 1911](#)

[4.15 O Enxadrista de Torres y Quevedo – 1912](#)

[4.16 Máquina Analítica de Torres y Quevedo – 1913](#)

[4.17 Flip-Flop – 1918](#)

[4.18 IBM – 1924](#)

[4.19 OCR de Gustav Tauschek – 1929](#)

[4.20 Tambor Magnético – 1932](#)

[4.21 Até Aqui...](#)

## **PARTE V - Os Primeiros Computadores**

[5.1 Máquina de Turing – 1936](#)

[5.2 Rapid Arithmetical Machine de Vannevar Bush – 1936](#)

[5.3 Z1 de Konrad Zuse: O primeiro computador binário programável – 1936](#)

[5.4 Placa de Circuito Impresso – 1936](#)

[5.5 Modelo K de George Stibitz – 1937](#)

- [5.6 Louis Pierre Couffignal – 1938](#)
- [5.7 Z2 – 1939](#)
- [5.8 Enigma e a Bomba Criptológica de Turing – 1939](#)
- [5.9 ABC: O primeiro computador a válvula – 1939](#)
- [5.10 Complex Number Computer ou Modelo 1 de Stibitz – 1940](#)
- [5.11 Claude Shannon – 1940](#)
- [5.12 Z3 – 1941](#)
- [5.13 RDA2: Rockefeller Differential Analyzer II – 1942](#)
- [5.14 Colossus Mark I – 1943](#)
- [5.15 IBM ASCC ou Harvard Mark I – 1944](#)
- [5.16 MEMEX – 1945](#)
- [5.17 ENIAC – 1945](#)
- [5.18 Plankalkül – 1946](#)
- [5.19 Transistor – 1947](#)
- [5.20 ACM - Association for Computing Machinery – 1948](#)
- [5.21 Z4 – 1948](#)
- [5.22 Cibernética – 1948](#)
- [5.23 IBM SSEC – 1948](#)
- [5.24 Manchester SSEM \(Baby\) – 1948](#)
- [5.25 EDVAC e a Arquitetura von Neumann – 1949](#)
- [5.26 EDSAC e o Assembly – 1949](#)
- [5.27 MADM ou Manchester Mark I – 1949](#)
- [5.28 Pilot ACE – 1950](#)
- [5.29 Teste de Turing – 1950](#)
- [5.30 Até Aqui...](#)

## **PARTE VI - Da Válvula ao Transistor**

- [6.1 Human Use of Human Beings – 1950](#)
- [6.2 Simon, o Primeiro Kit de Computador Pessoal – 1950](#)
- [6.3 Ferranti Mark I, o Primeiro Computador Comercial – 1951](#)
- [6.4 UNIVAC – 1951](#)
- [6.5 LEO I – 1951](#)

- [6.6 CSIRAC – 1951](#)
- [6.7 MIT Whirlwind, o Primeiro Computador de Tempo Real – 1951](#)
- [6.8 OXO – 1952](#)
- [6.9 Compilador A-0 – 1952](#)
- [6.10 IBM 701 – 1952](#)
- [6.11 IAS – 1952](#)
- [6.12 Trackball – 1952](#)
- [6.13 IBM 650 e 704 – 1953](#)
- [6.14 Primeiros Sistemas Operacionais – 1954](#)
- [6.15 Harwell Cadet, o Primeiro Computador Totalmente Transistorizado – 1955](#)
- [6.16 Disco Magnético – 1956](#)
- [6.17 Logic Theorist – 1956](#)
- [6.18 MUSIC – 1957](#)
- [6.19 Scanner – 1957](#)
- [6.20 FORTRAN – 1957](#)
- [6.21 ALGOL – 1958](#)
- [6.22 Tênis para Dois – 1958](#)
- [6.23 LISP – 1958](#)
- [6.24 Até Aqui...](#)

## **PARTE VII - Do Circuito Integrado à Lua**

- [7.1 Circuito Integrado – 1960](#)
- [7.2 COBOL – 1960](#)
- [7.3 DEC PDP-1, o Primeiro Minicomputador – 1960](#)
- [7.4 IFIP – International Federation for Information Processing – 1960](#)
- [7.5 Compartilhamento de Tempo – 1961](#)
- [7.6 Videogame Spacewar! – 1961](#)
- [7.7 Zezinho e Lourinha – 1961](#)
- [7.8 ReserVec, Sistema de Reserva de Passagens Aéreas – 1962](#)
- [7.9 Supercomputadores Atlas e Titan – 1962](#)
- [7.10 Mouse – 1962](#)
- [7.11 SAGE e os Maiores Computadores de Todos os Tempos – 1963](#)

- [7.12 Sketchpad, o Início da Computação Gráfica – 1963](#)
- [7.13 BASIC – 1964](#)
- [7.14 IBM 360 – 1964](#)
- [7.15 Comutação de Pacotes – 1965](#)
- [7.16 CDC 6600 de Seymour Cray – 1965](#)
- [7.17 Touch Screen – 1965](#)
- [7.18 Memória DRAM – 1966](#)
- [7.19 Chatterbot ELIZA – 1966](#)
- [7.20 OS/360 – 1967](#)
- [7.21 MUMPS – 1967](#)
- [7.22 Simula – 1967](#)
- [7.23 Logo – 1967](#)
- [7.24 Forth – 1968](#)
- [7.25 BESM 6, o Computador Soviético – 1968](#)
- [7.26 Espada de Dâmocles de Sutherland e o Início da Realidade Virtual – 1968](#)
- [7.27 Oki Wiredot, a Primeira Impressora Matricial – 1968](#)
- [7.28 HP 9100A, a Primeira Calculadora Programável Pessoal – 1968](#)
- [7.29 Dynabook de Alan Kay – 1968](#)
- [7.30 A Arte de Programação de Computadores de Knuth – 1968](#)
- [7.31 Crise do Software – 1968](#)
- [7.32 Microprocessador – 1968](#)
- [7.33 MULTICS – 1969](#)
- [7.34 ARPANET – 1969](#)
- [7.35 Arkay CT-650, o Computador de Clipe de Papel – 1969](#)
- [7.36 RS-232C – 1969](#)
- [7.37 Impressora Laser – 1969](#)
- [7.38 Apollo Guidance Computer – 1969](#)
- [7.39 Até Aqui...](#)

## **PARTE VIII - O Caminho para o Computador Pessoal**

- [8.1 Memória Intel 1103 – 1970](#)
- [8.2 Pascal – 1970](#)

- [8.3 Imlac PDS-1, o Computador Gráfico – 1970](#)
- [8.4 Bancos de Dados Relacionais – 1970](#)
- [8.5 PDP-11 – 1970](#)
- [8.6 Modelo Cascata – 1970](#)
- [8.7 Unix – 1970](#)
- [8.8 INTEL 4004, o Primeiro Microprocessador – 1971](#)
- [8.9 E-Mail – 1971](#)
- [8.10 Projeto Gutenberg, a Primeira Biblioteca Digital – 1971](#)
- [8.11 The Creeper, o Primeiro Vírus – 1971](#)
- [8.12 Computador Pessoal Kenbak-1 – 1971](#)
- [8.13 Disco Flexível – 1971](#)
- [8.14 Smalltalk – 1971](#)
- [8.15 C – 1972](#)
- [8.16 PROLOG – 1972](#)
- [8.17 Pato Feio – 1972](#)
- [8.18 Magnavox Odyssey – 1972](#)
- [8.19 Calculadora HP 9830A – 1972](#)
- [8.20 Pong – 1972](#)
- [8.21 Ethernet – 1973](#)
- [8.22 Xerox Alto – 1973](#)
- [8.23 Micral – 1973](#)
- [8.24 Salas de Chat Talkomatic – 1973](#)
- [8.25 Telefone Celular Pessoal – 1973](#)
- [8.26 Community Memory, o Primeiro BBS – 1973](#)
- [8.27 Intel 8080 – 1974](#)
- [8.28 Altair 8800 – 1974](#)
- [8.29 TCP-IP e o Surgimento da Internet – 1974](#)
- [8.30 COBRA, Computadores Brasileiros – 1974](#)
- [8.31 The Mythical Man-Month – 1975](#)
- [8.32 Computador Portátil IBM 5100 – 1975](#)
- [8.33 Sphere – 1975](#)
- [8.34 Circuito MOS 6502 – 1975](#)

- [8.35 Supercomputadores CRAY – 1976](#)
- [8.36 Primeiro Sistema Operacional para Microprocessador: CP/M – 1976](#)
- [8.37 Apple I – 1976](#)
- [8.38 Zilog Z80 – 1976](#)
- [8.39 Sol 20 – 1976](#)
- [8.40 Compucolor 8001, o Primeiro Computador a Cores – 1976](#)
- [8.41 Modem 80-103A – 1977](#)
- [8.42 DEC VAX 11 – 1977](#)
- [8.43 Apple II – 1977](#)
- [8.44 Tandy TRS80 – 1977](#)
- [8.45 Commodore PET – 1977](#)
- [8.46 Console de Videogame Atari 2600 – 1977](#)
- [8.47 Matlab – 1978](#)
- [8.48 Intel 8086 – 1978](#)
- [8.49 Xerox NoteTaker – 1978](#)
- [8.50 Planilha Eletrônica VisiCalc – 1978](#)
- [8.51 SBC - Sociedade Brasileira de Computação – 1978](#)
- [8.52 Editor de Texto WordStar – 1979](#)
- [8.53 Motorola 68000 – 1979](#)
- [8.54 SQL – 1979](#)
- [8.55 Até Aqui...](#)

## **PARTE IX - Evolução dos Computadores Pessoais e sua Interconexão**

- [9.1 dBASE II – 1980](#)
- [9.2 Linguagem Ada – 1980](#)
- [9.3 VIC-20 – 1980](#)
- [9.4 Apple III – 1980](#)
- [9.5 Sinclair ZX80 – 1980](#)
- [9.6 UseNet – 1980](#)
- [9.7 MS-DOS – 1980](#)
- [9.8 Xerox 8010 Star – 1981](#)

- [9.9 IBM PC – 1981](#)
- [9.10 Placa de Video CGA – 1981](#)
- [9.11 Revista Micro Sistemas – 1981](#)
- [9.12 Osborne 1 – 1981](#)
- [9.13 Rational Machines – 1981](#)
- [9.14 Sun Microsystems Inc. – 1982](#)
- [9.15 Commodore 64 – 1982](#)
- [9.16 Arquitetura RISC – 1982](#)
- [9.17 BBC Micro – 1982](#)
- [9.18 GRiD Compass 1101, o Primeiro Notebook – 1982](#)
- [9.19 80286 – 1982](#)
- [9.20 Compac Portable – 1982](#)
- [9.21 ZX Spectrum – 1982](#)
- [9.22 Computadores de Quinta Geração – 1982](#)
- [9.23 C++ – 1983](#)
- [9.24 Turbo Pascal – 1983](#)
- [9.25 Apple IIe – 1983](#)
- [9.26 Apple Lisa – 1983](#)
- [9.27 IBM PC-XT – 1983](#)
- [9.28 IBM PCjr – 1983](#)
- [9.29 Word – 1983](#)
- [9.30 Impressora 3D – 1984](#)
- [9.31 GNU – 1984](#)
- [9.32 Apple Macintosh – 1984](#)
- [9.33 Mac OS – 1984](#)
- [9.34 IBM AT – 1984](#)
- [9.35 Tetris – 1984](#)
- [9.36 Windows – 1985](#)
- [9.37 Atari ST – 1985](#)
- [9.38 Commodore Amiga – 1985](#)
- [9.39 PageMaker – 1985](#)
- [9.40 Acorn ARM – 1985](#)

- [9.41 CD-ROM – 1985](#)
- [9.42 Connection Machine – 1986](#)
- [9.43 Eiffel – 1986](#)
- [9.44 Amstrad PC-1512 – 1986](#)
- [9.45 Compaq Deskpro 386 – 1986](#)
- [9.46 Pixar – 1986](#)
- [9.47 Acorn Archimedes – 1987](#)
- [9.48 IBM PS/2 e OS/2 – 1987](#)
- [9.49 Perl – 1987](#)
- [9.50 VGA – 1987](#)
- [9.51 Photoshop – 1987](#)
- [9.52 Hypercard – 1987](#)
- [9.53 Worm – 1988](#)
- [9.54 IRC – 1988](#)
- [9.55 Archie – 1989](#)
- [9.56 Poqet PC – 1989](#)
- [9.57 World Wide Web – 1989](#)
- [9.58 Até Aqui...](#)

## **PARTE X - O Mundo se Conecta na World Wide Web**

- [10.1 PowerPoint – 1990](#)
- [10.2 Python – 1991](#)
- [10.3 Linux – 1991](#)
- [10.4 NCSA Mosaic – 1993](#)
- [10.5 Pentium – 1993](#)
- [10.6 Wanderer – 1993](#)
- [10.7 MP3 – 1993](#)
- [10.8 Yahoo! – 1994](#)
- [10.9 Simon Personal Communicator – 1994](#)
- [10.10 Netscape – 1994](#)
- [10.11 Amazon – 1994](#)
- [10.12 AltaVista – 1995](#)

- [10.13 Delphi – 1995](#)
- [10.14 Java – 1995](#)
- [10.15 Wiki – 1995](#)
- [10.16 USB – 1995](#)
- [10.17 PHP – 1995](#)
- [10.18 Windows 95 – 1995](#)
- [10.19 Ruby – 1995](#)
- [10.20 JavaScript – 1995](#)
- [10.21 eBay – 1995](#)
- [10.22 Palm Pilot – 1996](#)
- [10.23 Google – 1996](#)
- [10.24 Deep Blue – 1997](#)
- [10.25 Blog – 1997](#)
- [10.26 Clippy – 1997](#)
- [10.27 iMac – 1998](#)
- [10.28 Windows 98 – 1998](#)
- [10.29 Wi-Fi – 1999](#)
- [10.30 Napster – 1999](#)
- [10.31 Flash Drive – 1999](#)
- [10.32 Bug do Milênio – 1999](#)
- [10.33 Até Aqui...](#)

## **PARTE XI - Computação Móvel**

- [11.1 C# – 2000](#)
- [11.2 Manifesto Ágil – 2001](#)
- [11.3 Windows XP – 2001](#)
- [11.4 BitTorrent – 2001](#)
- [11.5 Wikipédia – 2001](#)
- [11.6 Skype – 2003](#)
- [11.7 Fundação Eclipse – 2004](#)
- [11.8 Facebook – 2004](#)
- [11.9 YouTube – 2005](#)

- [11.10 Twitter – 2006](#)
- [11.11 Computador Quântico – 2007](#)
- [11.12 iPhone – 2007](#)
- [11.13 Android – 2008](#)
- [11.14 WhatsApp – 2009](#)
- [11.15 Waze – 2009](#)
- [11.16 Até Aqui...](#)

## **PARTE XII - O Presente e o Futuro**

- [12.1 Chrome OS – 2010](#)
- [12.2 Instagram – 2010](#)
- [12.3 PewDiePie – 2010](#)
- [12.4 Watson – 2011](#)
- [12.5 Siri – 2011](#)
- [12.6 Google Glasses – 2012](#)
- [12.7 Michigan Micro Mote – 2014](#)
- [12.8 Apple Watch – 2015](#)
- [12.9 Previsões até 2020](#)
- [12.10 Previsões até 2030](#)
- [12.11 Previsões até 2040](#)
- [12.12 Previsões até 2050](#)
- [12.13 Previsões até 2100](#)
- [12.14 Previsões até 2200](#)
- [12.15 Previsões Finais](#)

## **Referências**

# PARTE I

## Pedra, Madeira e Ideias

Esta primeira parte do livro apresenta a história das primeiras formas de contar, registrar e calcular. Sua evolução começa na Pré-História e passa pela invenção dos primeiros mecanismos de contagem, especialmente nas civilizações babilônica, fenícia, grega e romana. Durante a Idade Média, a influência dos matemáticos árabes e indianos, com a criação do conceito de algoritmo, da criptografia e do zero como dígito, também foi importante para a evolução futura da computação. A Idade Antiga e Média testemunharam a invenção de mecanismos de computação analógica, ou seja, mecanismos que não calculam diretamente com números, mas com grandezas físicas. O período em análise estende-se desde a Pré-História até 1622, quando a régua de cálculo foi inventada.

### 1.1 Contagem com Dedos

Quem nunca contou com os dedos? Essa prática antiquíssima tem até nome: dactilonomia. A forma mais simples de usar os dedos para contar consiste em representar quantidades diretamente com os dedos esticados. A [Figura 1.1](#) mostra os números de 1 a 5 representados pelos dedos de uma das mãos.



**Figura 1.1: Dactilonomia decimal.<sup>1</sup>**

Esse sistema de contagem explica porque nossa cultura adotou a base 10 para representar números, afinal temos 10 dedos nas mãos (os dedos dos pés são difíceis de usar para contar).

Mas essa forma de contagem simples nem sempre foi a técnica mais usada. Suspeita-se de que os antigos babilônios utilizassem outra forma de calcular com os dedos que os levou a adotar o sistema sexagesimal em vez do decimal. A vantagem do sistema babilônico é que pode-se contar até 60 com as duas mãos (e não apenas até 10) e, com isso, negociar mais cabras e odres de vinho.

Como funcionava esse sistema de contagem? A ideia era que uma das mãos representasse os números de 1 até 12 fazendo o polegar tocar em uma das três falanges dos quatro dedos restantes ([Figura 1.2](#)). Assim, se o polegar tocasse a ponta do dedo mindinho, isso representava “1”, se ela tocasse o meio do dedo mindinho era “2” e se tocasse a base era “3”. A contagem prosseguia pelo dedo anular, médio e indicador, cuja base representava o “12”. Então com a segunda mão podia ser representada a magnitude do número usando a contagem simples: um dedo para uma dúzia, dois dedos para duas dúzias e assim por diante. Com 5 dedos se podia representar até 5 dúzias, ou seja, 60 quantidades diferentes.

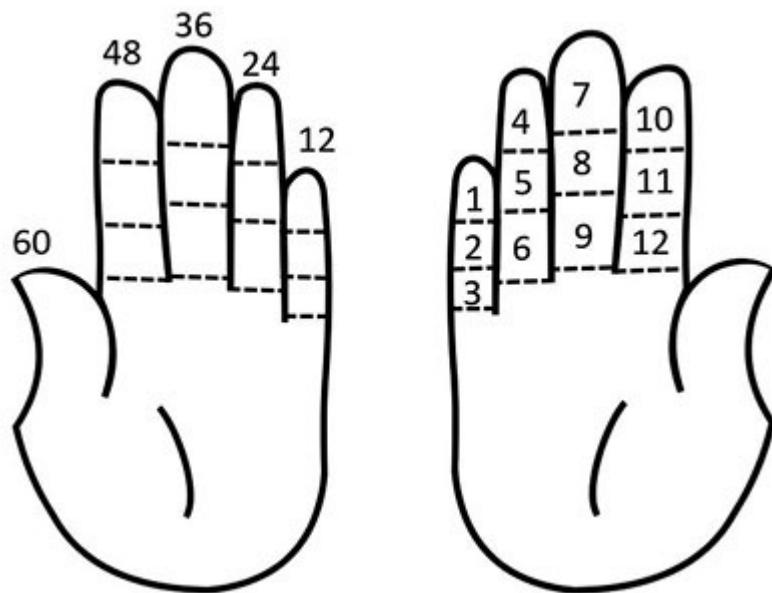


Figura 1.2: Dactilonomia sexagesimal.<sup>2</sup>

Poder-se-ia argumentar que esse sistema de contagem teria base 72, caso a mão usada para representar a magnitude contasse também o zero, além dos números de 1 a 5. Dessa forma, representando os números de 0 a 5 teríamos seis dúzias em vez de cinco. Porém, não há evidências de que os seres humanos houvessem concebido o zero como número na época dos babilônicos. Isso provavelmente explica porque a base 60 foi adotada em vez de 72.

A escolha do número 60 como base para contagem era muito apropriada para uma época em que a matemática ainda era muito incipiente. O número 60 podia ser facilmente dividido por vários outros números: 2, 3, 4, 5, 6, 10, 12, 15, 20 e 30, e, portanto, trabalhar com essa base era muito prático.

Tal sistema de contagem ainda é usado nos dias de hoje: uma hora se divide em 60 minutos e um minuto, 60 segundos. É uma base fácil de operar se você precisa de intervalos como meia hora, quarto de hora etc.

## **1.2 Vara de Contagem – c. 20.000 a.C.**

Quem observa hoje um computador com terabytes de memória não imagina que a necessidade de registrar números e outros tipos de informação já vem sendo suprida desde a Pré-História por instrumentos conhecidos como varas de contagem.

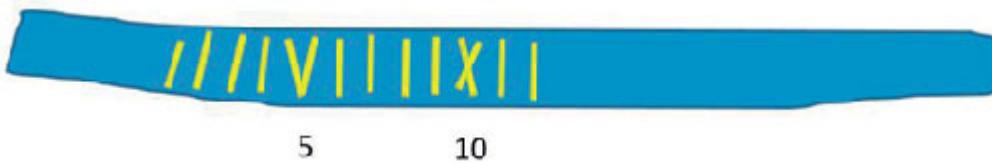
O objetivo principal da vara de contagem é registrar uma quantidade. Quem viveu na época em que os meninos brincavam com estilingues, deve lembrar que cada vez que um deles cometia a proeza de eliminar a vida de uma pobre ave, fazia uma pequena marca com canivete ou faca no cabo do seu estilingue. O total de marcas seria então o registro do total de pequenas vidas eliminadas pelo garoto (meninas usualmente não apreciavam a atividade).

Mas a vara de contagem apareceu em nossa história há muito tempo. Arqueólogos muitas vezes discordam das datas, mas parece ser consenso de que há pelo menos 20 mil anos já existiam estes instrumentos. Uma das mais antigas e famosas varas de contagem é o Osso de Ishango ([Figura 1.3](#)), um osso de babuíno com três colunas de marcas assimetricamente distribuídas, o qual foi encontrado próximo ao Rio Nilo na divisa entre os atuais Congo e Uganda.



**Figura 1.3: Osso de Ishango.<sup>3</sup>**

Há uma teoria que propõe que os numerais romanos ou seus antecessores possam ter surgido a partir de marcas em varas de contagem. A cada cinco marcas se fazia uma em forma de “V” e a cada 10 marcas uma em “X” para ajudar a visualizar a contagem, conforme mostrado na [Figura 1.4](#).



**Figura 1.4: Vara de contagem com protonumerais romanos.**

Com o tempo, segundo essa teoria, os numerais passaram a ser abreviados. Por exemplo, bastava escrever “XI” para indicar o onze ou escrever “IV” para indicar o quatro. Daí talvez, a ambiguidade com que normalmente se tratam estes números, já que em algumas situações quatro é grafado como “IIII” e em outras é “IV”.

Uma variação importante da vara de contagem é a vara dividida (*split tally*). Este instrumento foi usado até o século XIX em alguns lugares do mundo para registrar dívidas. Tomava-se uma vara de contar e anotava-se nela o valor da dívida com sulcos. Usualmente sulcos mais grossos indicavam magnitudes mais altas e sulcos mais finos as magnitudes mais baixas. Após a vara ser devidamente sulcada, ela era dividida longitudinalmente em duas partes, ficando uma com o credor e outra com o devedor ([Figura 1.5](#)). Qualquer tentativa de adulterar as marcações em uma das partes poderia ser facilmente rechaçada pela simples reunião das duas partes, visto que os sulcos deveriam coincidir, e não era possível apagar as marcas, que eram feitas em baixo relevo. Assim, essas varas foram um dos primeiros instrumentos de prevenção de fraudes em sistemas de informação.



Figura 1.5: Vara de contar dividida.<sup>4</sup>

O império britânico usou estas varas como registro de valores por centenas de anos. Curiosamente, elas foram responsáveis pela destruição do prédio do parlamento em 1843 quando foram postas fora de uso. Uma grande quantidade de antigas varas foi queimada na fornalha da Casa dos Lordes, e produziram tanto calor que o fogo fugiu do controle e destruiu o prédio.

### 1.3 Ábaco – c. 2.700 a. C.

O ábaco talvez tenha sido um dos mais antigos instrumentos de cálculo desenvolvido pelo ser humano. Sua origem conhecida remonta ao período entre 2.700 e 2.300 antes de Cristo na antiga Suméria (região na qual ficava a cidade de Babilônia).

Como quase todos os instrumentos criados pela engenhosidade humana, o ábaco veio a atender uma necessidade: representar e manipular números. Como os primeiros computadores modernos tinham como principal objetivo a manipulação de números, é natural que o ábaco sempre seja lembrado como uma das primeiras tentativas de construção de máquinas de computar.

Possivelmente os primeiros ábacos foram feitos com sulcos na terra nos quais pedrinhas representando quantidades eram colocadas. Cada sulco podia representar uma magnitude diferente do número,

a qual dependeria da base adotada. Os babilônios, como já vimos, adotavam uma base 60, composta por cinco dúzias.

Atribui-se aos romanos, milênios mais tarde, o aperfeiçoamento do ábaco, que passou a utilizar a base 10 e se tornou mais compacto permitindo que fosse mais facilmente manipulado. O ábaco romano seria, assim, a primeira calculadora de mão (*handheld*, ou portátil) de que se tem notícia ([Figura 1.6](#)).

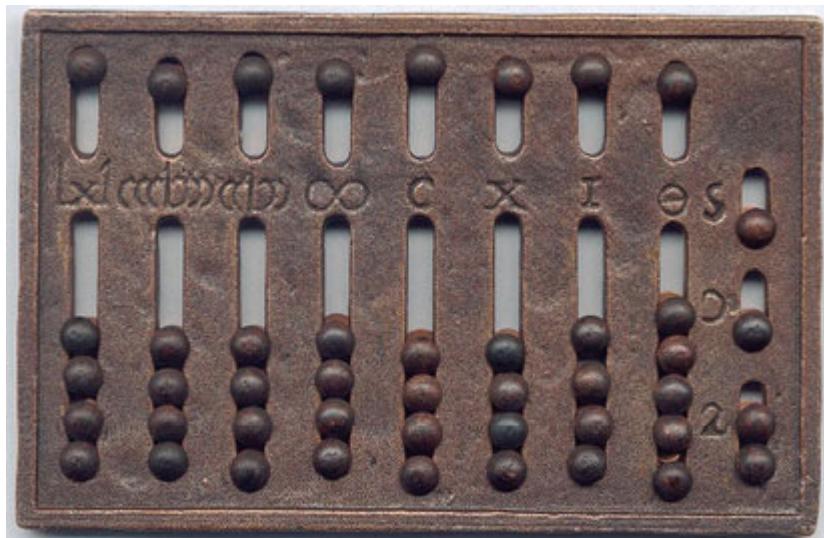


Figura 1.6: Ábaco romano.<sup>5</sup>

No ábaco romano cada linha representa uma ordem de magnitude. A coluna marcada com “I” representa as unidades (0-9), a marcada com “X” simboliza as dezenas (10-90) e assim por diante. Para representar um número de 1 a 4 move-se o número de bolinhas correspondente da parte inferior do ábaco para o centro. Para representar números entre 5 e 9 move-se a bolinha da parte superior para o centro e soma-se a ela a quantidade necessária de bolinhas da parte inferior.

Para somar um número a um número que já esteja representado no ábaco, move-se uma quantidade correspondente de bolinhas em cada coluna. Se o algarismo somado ao que já estava representado passar de 9, então soma-se 1 na coluna imediatamente a esquerda,

indicando o “*carry*” ou “vai um”. Este é um conceito muito importante que foi automatizado nas primeiras calculadoras mecânicas inventadas pela humanidade e depois nos computadores eletrônicos.

O ábaco também foi muito usado na antiga China, onde é chamado de *suanpan*. Não há consenso entre os pesquisadores sobre se o ábaco chinês foi desenvolvido de forma independente ou se foi inspirado no ábaco europeu. Presume-se que tenha havido contato entre as civilizações oriental e ocidental desde a Antiguidade, mas não há provas concretas de que a tecnologia do ábaco tenha sido importada pelos chineses a partir dessa eventual comunicação.

Uma particularidade da maioria dos ábacos chineses é que eles usam 5 contas mais 2 em vez de 4 mais 1, como os romanos. Com esta quantidade de contas é possível fazer operações também com números hexadecimais, ou seja, de base 16. Curiosamente, a base hexadecimal se tornou muito importante para a computação com o surgimento dos primeiros computadores binários, visto que era mais fácil representar códigos de máquina usando números hexadecimais do que através de binários ou decimais. Os primeiros computadores pessoais realmente populares no final dos anos 1970 permitiam que o usuário programasse instruções de máquina diretamente na memória do computador usando o comando “*poke*”, cujos parâmetros eram um endereço de memória e um valor, ambos em hexadecimal. Como os algarismos arábicos só vão até 9, os números entre 10 e 15 eram representados pelas seis primeiras letras do alfabeto: A, B, C, D, E e F. Assim, o número  $10_h$  na verdade vale 16 em decimal, e  $1B_h$  corresponde ao número 27 ( $=16+11$ ).

Um aprimoramento tecnológico importante do ábaco foi quando se passou a usar contas presas em um arame em lugar das pedrinhas em sulcos. Com isso, possivelmente, os cálculos passaram a ser feitos muito mais rapidamente, pois as contas são mais fáceis de controlar e não ficam caindo para fora das canaletas toda hora.

## 1.4 Silogismos – c. 400 a.C.

Os silogismos se configuram como uma das primeiras tentativas reconhecidas da Humanidade de organizar e entender o raciocínio como uma ciência formal, ou seja, as conclusões a serem tomadas em um raciocínio dependeriam apenas da sua forma e não do seu conteúdo. Um exemplo de raciocínio deste tipo é: “Se todos os X são Y e todos os Y são Z, então todos os X são Z”; não importa o que sejam X, Y e Z – o raciocínio é sempre verdadeiro.

Aristóteles (Grécia, 384 a.C. – 322 a.C.) foi um dos filósofos que mais contribuiu para a formalização do silogismo. Ele combateu os sofistas de seu tempo, os quais, segundo consta, se interessavam mais em convencer as pessoas a qualquer custo do que efetivamente buscar a verdade.

As afirmações, ou proposições categóricas, nos silogismos podem ser de quatro tipos:

- A: Todos os X são Y.
- E: Nenhum X é Y.
- I: Alguns X são Y.
- O: Alguns X não são Y.

Os dois primeiros tipos são universais e os dois últimos são particulares.

Os silogismos são sempre formados por três afirmações, cada qual classificada em um dos quatro tipos acima:

- Premissa maior.
- Premissa menor.
- Conclusão.

A conclusão necessariamente deve conter um dos elementos da premissa maior e um elemento da premissa menor. Considere novamente o exemplo: “Se todos os X são Y (premissa maior) e todos os Y são Z (premissa menor), então todos os X são Z

(conclusão)." Neste caso, as duas premissas e a conclusão são do tipo A.

Considerando apenas a forma das afirmações, e ignorando a ordem das premissas (que, como diz o ditado, não afeta a conclusão), existem 256 combinações possíveis de premissas e conclusões, mas apenas 24 delas são raciocínios verdadeiros.

Para facilitar a memorização dos silogismos verdadeiros, passou-se a utilizar palavras-chave, ou mnemônicos, formados com as quatro vogais que identificam os tipos de proposições combinadas três a três. Seguem alguns exemplos de silogismos verdadeiros:

- AAA – *Barbara*: todos os homens são mortais, todos gregos são homens, portanto, todos gregos são mortais.
- EAE – *Celarent*: nenhum réptil tem pelos, todas cobras são répteis, portanto, nenhuma cobra tem pelos.
- All – *Darii*: todos coelhos têm pelos, alguns animais são coelhos, portanto alguns animais têm pelos.
- EIO – *Ferio*: nenhum dever é divertido, algumas leituras são dever, portanto algumas leituras não são divertidas.
- AOO – *Baroco*: toda informação é útil, alguns websites não são úteis, portanto alguns websites não são informação.
- OAO – *Bocardo*: alguns gatos não têm cauda, todos gatos são mamíferos, portanto alguns mamíferos não têm cauda.
- AAI – *Barbari*: todos os homens são mortais, todos gregos são homens, portanto alguns mortais são gregos.
- EAO – *Celaront*: nenhum réptil tem pelos, todas cobras são répteis, portanto algumas cobras não têm pelos.
- AEO – *Camestros*: todos cavalos têm cascos, nenhum humano tem cascos, portanto alguns humanos não são cavalos.
- EAO – *Felapton*: nenhuma flor é animal, todas flores são plantas, portanto algumas plantas não são animais.

- AAI – *Darapti*: todos os quadrados são retângulos, todos retângulos são polígonos, portanto alguns polígonos são retângulos.

Por outro lado, a maioria das combinações é falaciosa, ou seja, as premissas chegam a conclusões falsas, ou ainda podem até chegar a conclusões verdadeiras, mas que não seguem das premissas. Por exemplo: “Alguns cães são vira-latas, alguns cães são grandes, portanto, alguns vira-latas são grandes” – a conclusão é verdadeira porque sabemos que alguns vira-latas são grandes, mas isso não segue das premissas, pois a conclusão poderia ser falsa mesmo que as premissas fossem verdadeiras. Há vários padrões de premissas falaciosas, dentro os quais pode-se citar, por exemplo:

- *O meio não distribuído*: todos os gregos são homens, todos os romanos são homens, portanto todos os gregos são romanos.
- *Tratamento ilícito da premissa maior*: todos os cães são animais, nenhum gato é cão, portanto nenhum gato é animal.
- *Tratamento ilícito da premissa menor*: todos os gatos são felinos, todos gatos são mamíferos, portanto, todos os mamíferos são felinos.
- *Premissas exclusivas*: nenhum gato é cão, alguns cães não são animais, portanto alguns animais não são gatos.
- *Conclusão afirmativa de premissas negativas*: nenhum peixe é cão, nenhum cão pode voar, portanto todos os peixes podem voar.
- *Conclusão negativa de premissas positivas*: todos os coronéis são oficiais, todos oficiais são soldados, portanto nenhum coronel é soldado.

Os silogismos foram extremamente importantes na história do raciocínio humano. Apenas no final do século XIX, com o trabalho de Gottlob Frege (Alemanha, 1848-1925), eles foram sucedidos por uma forma mais completa de raciocínio, conhecida como lógica de primeira ordem. Algumas áreas da computação e mesmo

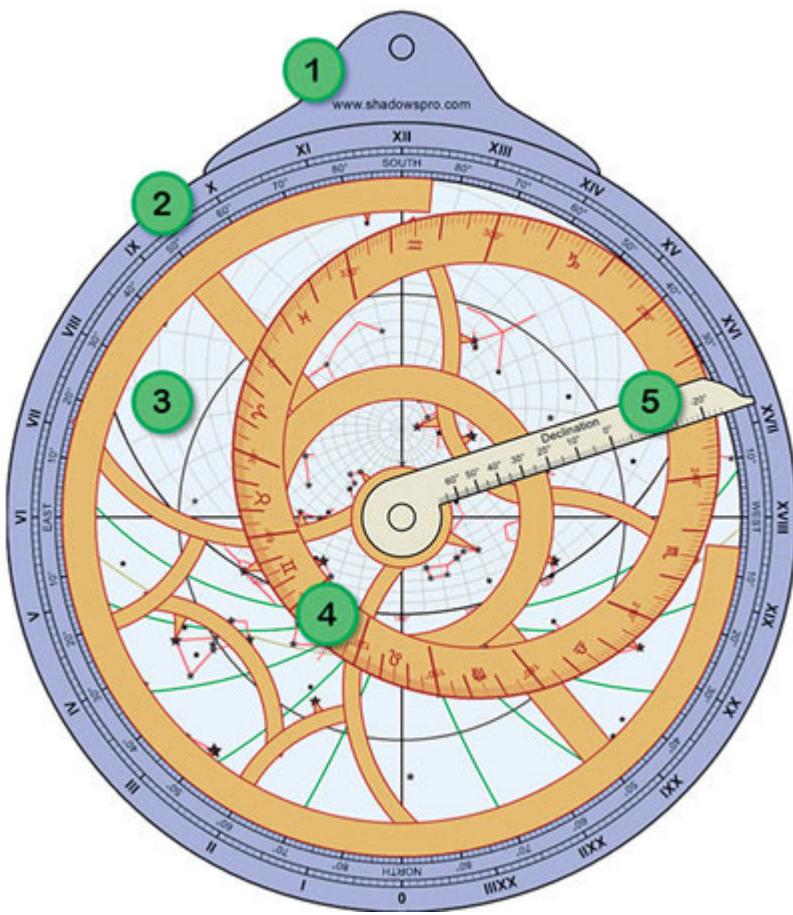
linguagens de programação atuais baseiam-se fortemente nos conceitos desta lógica.

## 1.5 Astrolábio Planisférico – c. 150 a.C.

O astrolábio grego ou planisférico era um instrumento de medição que podia ser usado para calcular a hora do dia ou da noite a partir da observação da posição do Sol ou de uma dada estrela. Ele não deve ser confundido com o astrolábio náutico, bem mais simples, pois o astrolábio grego foi feito para ser usado em terra firme e não no balanço do convés de um barco. Sua concepção ou aperfeiçoamento são atribuídos a Hiparco de Niceia (Grécia, c. 190 a.C. - 120 a.C.), astrônomo e matemático considerado criador da trigonometria e descobridor da precessão dos equinócios.<sup>6</sup>

O astrolábio grego foi construído como a combinação de dois outros instrumentos: o planisfério e a dióptrica. Basicamente a dióptrica é usada para medir um ângulo, usualmente aquele formado pela linha imaginária que liga o instrumento a uma estrela ou planeta e a linha imaginária que liga o instrumento ao ponto no horizonte que fica logo abaixo desse astro. Já o planisfério é uma representação esquemática do céu em um disco.

A [Figura 1.7](#) apresenta os principais componentes de um astrolábio planisférico. O anel de sustentação (1) era usado para que o peso do próprio astrolábio o mantivesse na posição correta. A madre (2) é a parte principal e contém uma escala dada em horas e graus. Os tímpanos (3) são gravados com um sistema de coordenadas adequado para uma determinada latitude, mostrando os círculos correspondentes aos trópicos e equador. A aranha (4) era colocada sobre o tímpano, podendo rodar sobre o eixo central do instrumento e apontar para as principais estrelas. Finalmente, a régua (5) permitia determinar a posição de uma determinada estrela no céu com base na data corrente e outros cálculos.



**Figura 1.7: Um astrolábio planisférico.<sup>7</sup>**

O usuário deve segurar o astrolábio pela sua argola de sustentação e alinhar a régua entre sua posição e o astro em questão. Dessa forma, o ângulo poderá ser visualizado na placa principal. A representação do planisférico no astrolábio permite então associar tais ângulos com a posição de estrelas, planetas, latitude e época do ano.

O site <http://www.shadowsp.com/help/en/astrolabeusages.html> apresenta na prática como resolver uma série de problemas utilizando um astrolábio planisférico. Entre outros problemas pode-se citar:

- Determinar a hora e direção do nascer e pôr do Sol em uma determinada data.

- Determinar o instante em que o Sol estará em um determinado azimute<sup>8</sup> em uma determinada data.
- Determinar o instante em que o Sol estará em um dado azimute e altitude, sem conhecer a data.
- Determinar a hora do nascer ou pôr de uma estrela na aranha em uma determinada data.
- Determinar o instante de culminação<sup>9</sup> de uma estrela na aranha em uma determinada data.
- Determinar a altitude máxima do Sol em um determinado ano e uma determinada localização.
- Determinar a altitude máxima de uma estrela em uma determinada data para uma determinada localização.
- Determinar a hora corrente a partir da altitude de uma estrela para uma determinada localização em uma determinada data.

Desta forma, o astrolábio pode perfeitamente ser considerado como um computador analógico, já que permite realização de cálculos com seu mecanismo móvel. Esse tipo de mecanismo de cálculo analógico possivelmente atingiu seu auge com a criação das réguas de cálculo, muito usadas por engenheiros e outros profissionais das ciências exatas até meados do século XX.

## 1.6 Mecanismo de Anticítera – c. 125 a.C.

Um dos mais surpreendentes exemplos da tecnologia do mundo antigo foi encontrado por volta de 1.900 em um naufrágio próximo à ilha grega de Anticítera ([Figura 1.8](#)). Sua construção foi atribuída aos gregos e data do segundo século antes de Cristo. O mecanismo formado por pelo menos 30 engrenagens é considerado um complexo computador analógico.



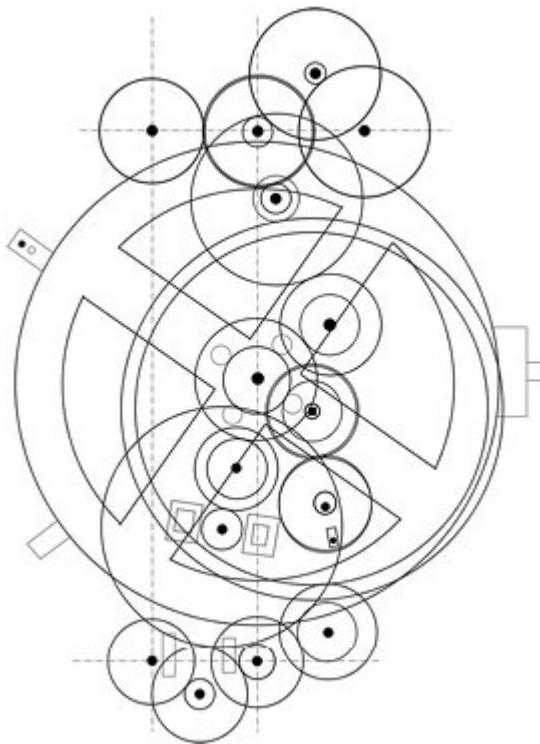
**Figura 1.8: Um dos fragmentos do mecanismo de Anticítera.<sup>10</sup>**

O mecanismo podia ser usado, dentre outras coisas, para prever a posição de planetas e eclipses. Nada parecido com ele foi encontrado no mundo até o final da Idade Média, cerca de 1.400 anos depois, quando relógios astronômicos começaram a ser construídos na Europa.

O mecanismo ficava alojado em uma caixa de madeira de 34 x 18 x 9 cm. A maior das engrenagens, claramente vista na [Figura 1.8](#) tinha 14 cm de diâmetro e 233 dentes. A qualidade e complexidade deste mecanismo sugere que outros devam ter sido construídos antes dele, mas até hoje, nenhum deles foi encontrado.

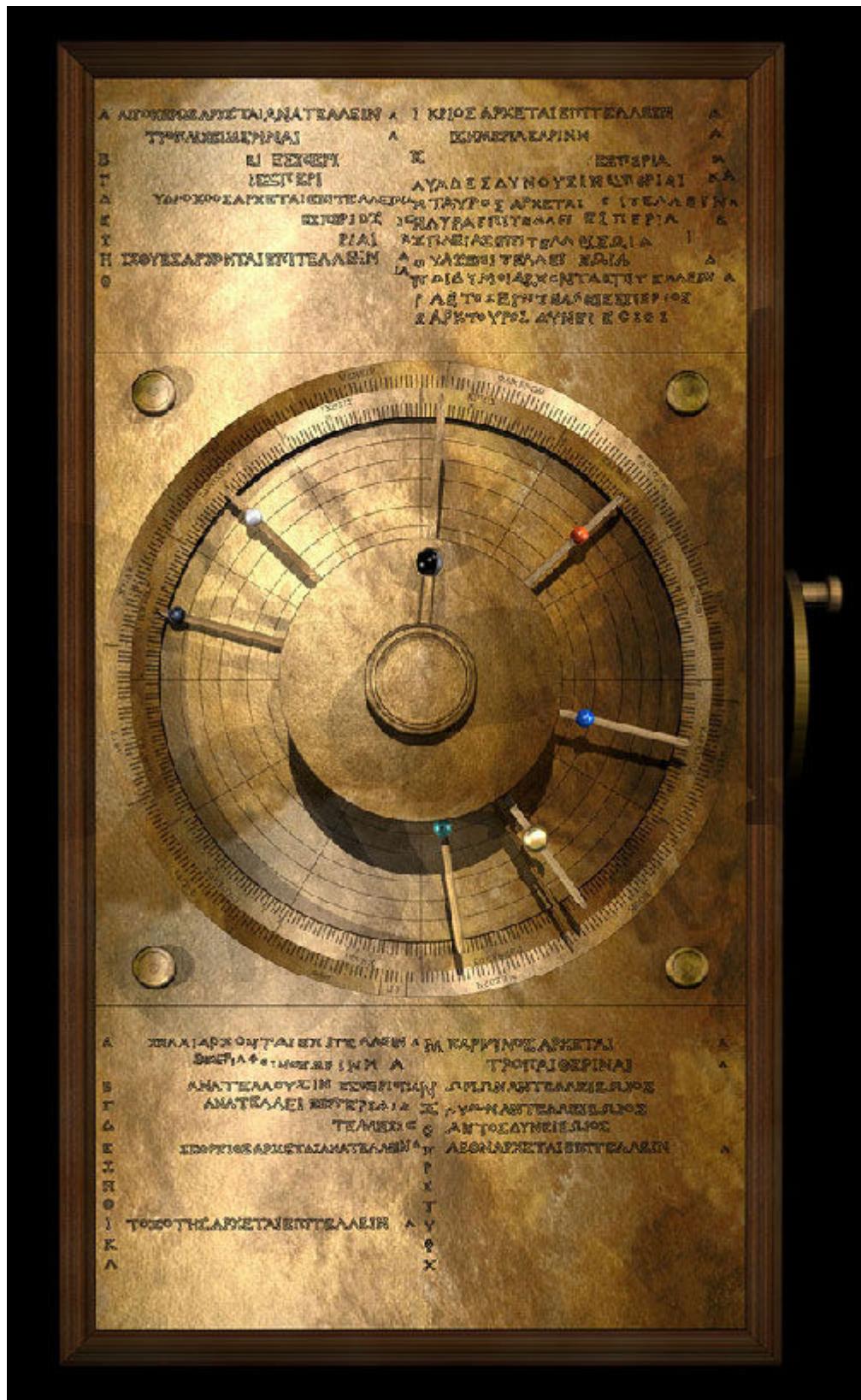
Ele é considerado um computador analógico por conta de sua capacidade de computar (calcular), dentre outras coisas, a posição dos planetas conhecidos na época e das constelações do zodíaco a partir do giro de suas engrenagens.

A [Figura 1.9](#) apresenta esquematicamente a posição das engrenagens do mecanismo.



**Figura 1.9: Esquema do mecanismo de Anticítera.<sup>11</sup>**

A [Figura 1.10](#) apresenta uma reprodução da parte frontal do mecanismo, onde se vê um anel fixo representando a eclíptica (linha imaginária por onde passa o Sol, a Lua e os planetas no céu). A eclíptica é dividida em 12 setores de 30 graus, cada um representando uma das 12 constelações do zodíaco. Por fora deste anel há outro, móvel, marcado com os 12 meses de 30 dias mais os 5 dias intercalados do calendário sótico, usado no antigo Egito.



**Figura 1.10:** Parte frontal do mecanismo de Anticítera.<sup>12</sup>

O funcionamento do mecanismo possivelmente era realizado a partir do giro de uma manivela (mostrada na parte direita da [Figura 1.10](#)). Ajustando-se os mostradores à data desejada, os ponteiros no painel estariam apresentando a posição dos vários planetas e constelações no céu.

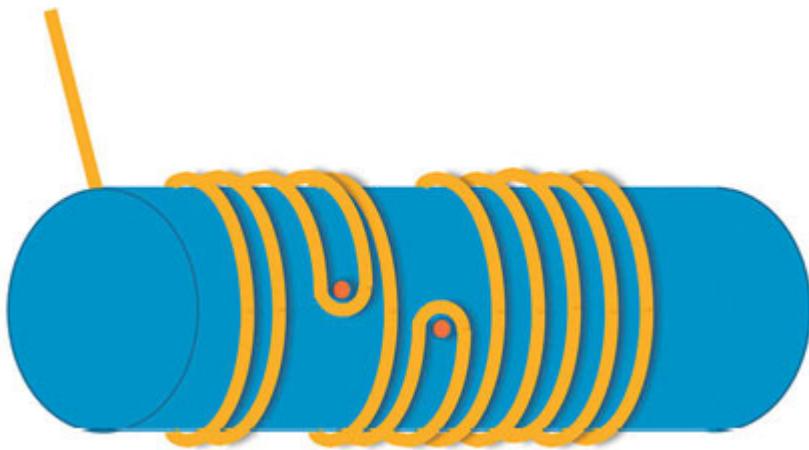
Especula-se quais seriam as razões pelas quais os gregos, com tal nível de conhecimento e engenharia, não foram capazes de desenvolver uma revolução industrial, e por que esse conhecimento se perdeu por tanto tempo, para só ser redescoberto mais de mil anos depois.

## 1.7 O Carro Programável de Heron de Alexandria – 60 d.C.

Atribui-se a Heron de Alexandria (Egito, 10-70 d.C.) a construção do primeiro carro robô programável. O carro de Heron tinha três rodas e podia ser programado para andar para a frente ou para trás, e também para girar para a direita ou esquerda.

O modelo era genial, mas bem simples, e as duas rodas traseiras tinham eixos independentes. Uma corda era enrolada em cada um dos eixos e as duas cordas eram presas a um peso que ficava suspenso por uma haste amarrada ao carro. À medida que o peso descia, os dois cabos iam puxando o eixo e fazendo o carrinho se mover.

Até aí nada demais. A grande sacada de Heron foi a utilização de pinos nos eixos que permitiam que se enrolasse a corda não apenas em um sentido, mas também no sentido inverso ([Figura 1.11](#)). Dessa forma quando a corda que estava sendo esticada atingisse um dos pinos, ela automaticamente fazia com que o sentido do giro do eixo se invertesse. Como os eixos eram independentes e podiam ter seus pinos configurados também de forma autônoma, isso permitia que o carrinho fosse literalmente programado para executar movimentos bastante complexos por um tempo razoável.



**Figura 1.11: Detalhe do primeiro robô programável de que se tem notícia.**

Pode-se dizer que esse robô programável antecedeu em quase 2.000 anos o conceito de Turtlegraphics, usado em ambientes de programação como Logo e Forth.

## 1.8 O Zero – 628

Após a queda do Império Romano e, com ele, de grande parte da civilização ocidental, a cultura na Europa entrou em declínio. Coube aos cientistas árabes, indianos e chineses a maior parte das descobertas científicas ocorridas durante o período da Idade Média.

Uma destas contribuições, importantíssima para a história dos computadores, foi a invenção (ou descoberta) do número zero. De fato, em 628, um matemático indiano chamado Brahmagupta (Índia, 598-670) publicou um livro: o *Brāhmasphutasiddhānta* (*Tratado Completo sobre a Escola de Brahma*). Neste livro houve o primeiro registro conhecido do zero como número. A [Figura 1.12](#) mostra uma concepção artística de Brahmagupta traçando o símbolo que hoje representa o zero.



Figura 1.12: Brahmagupta e o zero.<sup>13</sup>

Anteriormente, a noção de zero já era conhecida, mesmo durante o tempo dos romanos ou babilônios. Mas o zero era considerado como o “nada”, como a “ausência de números”. Até hoje, a maioria dos matemáticos considera que os números naturais são  $1, 2, 3, \dots$ , excluindo o zero deste conjunto. Um romano não teria meios para representar o zero. Ele apenas deixaria um espaço em branco. Isso, claro, dificultaria as operações aritméticas com números negativos.

Brahmagupta, por outro lado, além de considerar o zero como um algarismo, definiu todo um conjunto de regras aritméticas envolvendo-o e os números positivos (que ele chamava de “fortunas”) e negativos (chamados por ele de “débitos”).

Uma de suas regras dizia: “A soma de dois positivos é positiva, e de dois negativos é negativa. A soma de um positivo e um negativo é sua diferença; e, se eles são iguais, ela é igual a zero. A soma de

um negativo e zero é negativa, de um positivo e zero é positiva, e de dois zeros é zero.”

Sobre a multiplicação, ele escreveu: “O produto de um negativo e um positivo é negativo, de dois negativos é positivo, e de dois positivos é positivo. O produto de zero e um negativo é zero, de zero e um positivo é zero, e de dois zeros é zero.”

Apenas a sua interpretação sobre a divisão por zero difere da que temos hoje. Hoje, qualquer número dividido por zero tem resultado indefinido. Para Brahmagupta, zero dividido por zero é zero. Em relação a outros números divididos por zero, ele não chega a estabelecer nenhuma regra, postulando apenas que a divisão de um número  $n$  por zero é uma fração na qual zero é o divisor, ou seja  $n/0$ .

A existência do zero também foi importante para o surgimento da computação mecânica porque a partir daí os números poderiam ser representados pelo sistema de posicionamento decimal, ou seja, depois do 9 vem 10 (um seguido de zero); depois de 99 vem 100, e assim por diante. Esse tipo de notação permitiu simplificar os cálculos mecânicos realizados pelas primeiras calculadoras do século XVII, como a de Pascal, Schickard e Leibniz.

## 1.9 Relógio Mecânico – 724

Relógios são importantes para a computação não apenas por ajudarem no processo de sincronização de complexos mecanismos eletrônicos, mas também pelo fato de que os primeiros computadores mecânicos, ou calculadoras, foram construídos a partir do século XVII efetivamente por relojoeiros. Ou seja, embora as primeiras calculadoras e computadores mecânicos tivessem sido inventados por cientistas como Pascal, Schickard, Leibniz e Babbage, sua efetiva elaboração só foi possível graças à tecnologia dos relógios, já que essas máquinas ancestrais eram baseadas na movimentação de engrenagens.

A invenção do primeiro relógio mecânico efetivo é atribuída a Liang Lingzan e Yi Xing (China, 683-727). Ele era movido a água e era capaz de medir com exatidão, para os padrões da época, a duração de um dia e as fases da Lua. Além disso, a cada hora cheia ele tocava uma campainha e a cada quarto de hora executava uma batida de tambor.

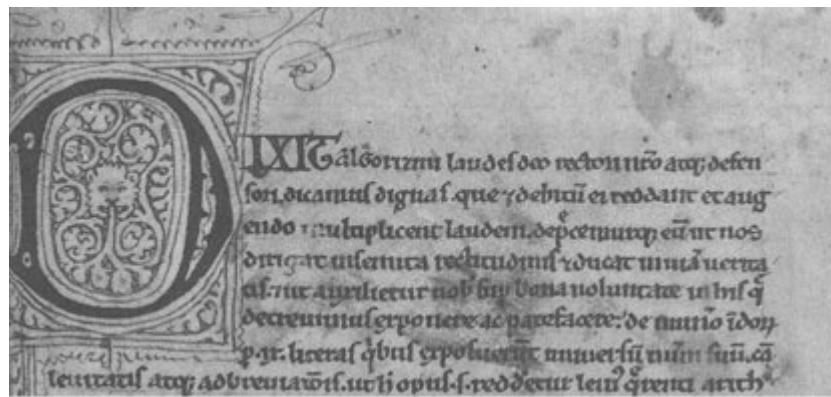
Infelizmente, mesmo considerando que tal equipamento deveria ter sido construído a partir de diagramas bem elaborados, muito poucos registros de sua existência restaram nos dias de hoje. Sendo assim, seu mecanismo efetivo é alvo de especulação.

Essa tecnologia só começou a ser explorada na Europa a partir do século XIV quando algumas catedrais passaram a construir seus próprios relógios astronômicos.

## 1.10 Algoritmo – 825

A programação de computadores segue uma técnica formal conhecida como “algoritmo”. Esse termo surgiu a partir do nome de um matemático chamado Abdullah Muhammad bin Musa al-Khwarizmi (Pérsia, 780 - c. 850), que escreveu um livro sobre cálculos com os algarismos indianos no ano 825.

Esta obra foi responsável pela popularização dos algarismos indianos (hoje conhecidos como arábicos ou indo-arábicos) na Europa e Oriente Médio, os quais eram bem mais fáceis de operar do que os números romanos, até então usados. O título do livro foi traduzido para o latim como “*Algoritmi de numero Indorum*” e assim a tradução para o latim do nome deste matemático passou a significar genericamente qualquer procedimento mecânico para realização de cálculos. A [Figura 1.13](#) mostra o início da primeira página da tradução latina do livro, onde se lê “*Dixit algorizmi...*”.



**Figura 1.13: Início do livro de al-Khwarizmi.<sup>14</sup>**

Entre outras contribuições, al-Khwarizmi mostrou como realizar adições e divisões com números longos, um procedimento (ou algoritmo) até hoje ensinado nas escolas, e que foi a base para a construção das primeiras calculadoras mecânicas.

Observe, por exemplo, a dificuldade para se somar números romanos como  $XIX + XXI = XL$ . A notação romana tinha regras relativamente complicadas para a soma:

- Primeiramente os números deviam ser descompactados, trocando-se IV por IIII, IX por VIIII etc.: XVIIII+XXI.
- Depois eles são concatenados, ou seja, juntados: XVIIIIXXI.
- Depois deve-se ordenar as letras da mais significativa para a menos significativa: XXXVIIIIII.
- Finalmente o número resultante é recomposto: XXXVIIIIII = XXXV = XXXX = XL.

Bem, a soma até que não era tão difícil, mas a divisão...

Com a notação indo-árabica, baseada em posições decimais, as somas podiam ser feitas de forma mais automática, da direita para a esquerda, quando os dois algarismos somados resultavam em um número maior do que nove, somava-se um (*carry*) aos dois algarismos imediatamente à esquerda. Esse algoritmo era bem mais fácil de representar mecanicamente.

## 1.11 Música Mecânica – 850

O século IX experimentou, talvez pela primeira vez na história, a música mecânica, ou seja, música tocada automaticamente, sem intervenção humana. Isso aconteceu na Pérsia, atual Irã, mais especificamente na cidade de Bagdá. Ali, três irmãos, conhecidos como os irmãos Banū Mūsā (filhos de Moisés), publicaram um livro no ano 850, chamado *Livro dos Dispositivos Engenhosos*, no qual descreviam, dentre muitas outras invenções, um órgão que podia ser tocado automaticamente a partir de um comando passado por um cilindro com pinos ([Figura 1.14](#)). O cilindro girava lentamente e cada vez que um dos pinos passava sob um dos tubos do órgão, uma válvula se abria e fazia o respectivo tubo tocar o seu som.

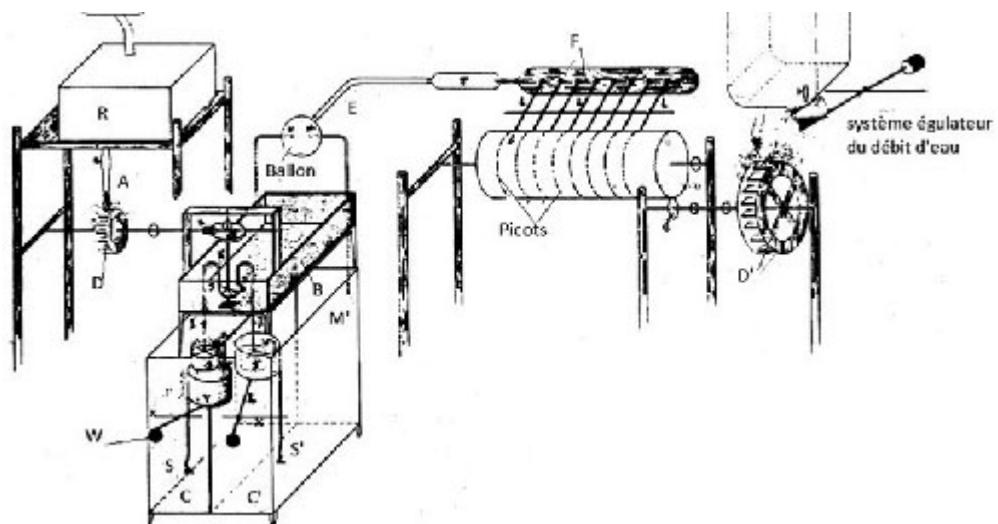


Figura 1.14: Esquema do órgão mecânico dos irmãos Banū Mūsā.<sup>15</sup>

Os irmãos se chamavam Abū Ja'far, Muḥammad ibn Mūsā ibn Shākir, Abū al-Qāsim, Aḥmad ibn Mūsā ibn Shākir e Al-Ḥasan ibn Mūsā ibn Shākir, sendo todos acadêmicos que viveram e trabalharam em Bagdá.

O princípio usado em seu órgão mecânico ainda hoje é utilizado em pequenas caixas de música e até meados do século XIX era a forma

dominante de execução de música mecânica, quando foi substituído por rolos de papel perfurado, usados em pianolas no século XIX e finalmente pela música eletrônica tocada por computadores nos dias de hoje.

## 1.12 Criptografia – c. 850

Embora o termo “criptografia” tenha sido cunhado apenas em 1920, um de seus métodos clássicos foi definido mais de mil anos antes. No século IX, por volta de 850, o matemático árabe Al-Kindi (Iraque, 801-853), conhecido no Ocidente como Alkindus, publicou um manuscrito sobre a decifração de mensagens criptográficas.

O manuscrito incluía uma descrição do método de análise de frequência para a decifração de mensagens criptográficas simples. Suponha, por exemplo, uma criptografia simples de substituição de símbolos na qual cada letra da mensagem original é substituída pela letra seguinte do alfabeto e o Z substituído pelo A. Com essa técnica, uma palavra escrita, por exemplo como “dsquphsbgbj” seria decifrada se substituíssemos cada letra pela letra imediatamente anterior no alfabeto, resultando assim na palavra decifrada: “criptografia”.

Agora imagine que as letras fossem substituídas de forma não tão lógica, como por exemplo, “A” por “N”, “B” por “%”, “C” por “9” etc. Como você decifraria uma mensagem como “HGS&JW%@KKS”? Sem saber o método de criptografia fica muito difícil traduzir a mensagem, pois cada caractere na frase pode ser, em princípio, qualquer uma das letras, números ou sinais especiais utilizados na escrita.

O método descrito por Al-Kindi baseia-se em uma análise estatística da frequência de cada letra numa determinada língua. Em português, por exemplo, a letra mais frequente em textos é a letra “A”, que aparece em média 14,63%, seguida da letra “E” com uma média de 12,57%. Assim, em uma mensagem criptografada suficientemente longa, que se saiba estar escrita originalmente em

português, um caractere que apareça por volta de 14,63% das vezes tem grande chance de ser a letra “A”. Já um caractere que apareça cerca de 12,57% das vezes tem chance de ser a letra “E”. E assim por diante.

Algumas letras têm probabilidade muito parecida, como “T” e “U” com 4,74% e 4,63% de chance. Nestes casos, o método puro e simples de análise de frequência pode não ser efetivo. Então, usa-se o conhecimento da língua para identificar a letra em questão; por exemplo, se o símbolo “#” aparece com uma distribuição de 4,69% (entre as distribuições de “T” e “U”), mas ocorre com muita frequência após consoantes que já foram decifradas, como “Q” e “G”, então esse símbolo mais provavelmente é uma vogal, no caso “U” e não uma consoante como “T”.

## 1.13 Autômatos de Al-Jazari – 1206

Badi’al-Zaman Abū al-’Izz ibn Ismā’īl ibn al-Razāz al-Jazarī, ou simplesmente al-Jazari (Turquia, 1136-1206) foi um cientista responsável por várias invenções que tiveram impacto na história da computação. Em 1206 ele publicou o *Livro do Conhecimento dos Dispositivos Mecânicos Engenhosos*, no qual descreve mais de 100 dispositivos mecânicos. Esse livro se tornou muito popular em sua época e nos séculos seguintes.

Uma destas invenções foi o eixo excêntrico, usado até hoje nos comandos de válvulas dos automóveis e que permitia imprimir movimento rítmico a outros elementos de um projeto mecânico. No caso dos automóveis, esse comando permite abrir e fechar as válvulas do motor a partir do giro do eixo. Esse mecanismo foi utilizado também na Máquina Analítica de Babbage, o primeiro computador de propósito geral projetado. Para uma melhor compreensão do conceito, sugere-se visualizar a animação disponível

[https://en.wikipedia.org/wiki/Camshaft#/media/File:Nockenwelle\\_ani.gif](https://en.wikipedia.org/wiki/Camshaft#/media/File:Nockenwelle_ani.gif).

Mais impressionante ainda, al-Jazari foi responsável pela criação de vários autômatos de forma humana, movidos principalmente a água e capazes de executar diversas tarefas. Os autômatos já eram conhecidos dos gregos antigos (por exemplo, Heron de Alexandria), mas enquanto os gregos os construíam aparentemente apenas pela curiosidade ou entretenimento, al-Jazari projetou autômatos que realmente prestariam serviços úteis aos seres humanos, antecipando assim em mais de 700 anos funções que só seriam plenamente automatizadas a partir do século XX como, por exemplo, portas que se abrem automaticamente e máquinas para servir bebidas.

Consta que os desenhos de al-Jazari, um dos quais mostrado na [Figura 1.15](#), teriam influenciado ou inspirado muitas das invenções de Leonardo da Vinci, mais de duzentos anos depois.

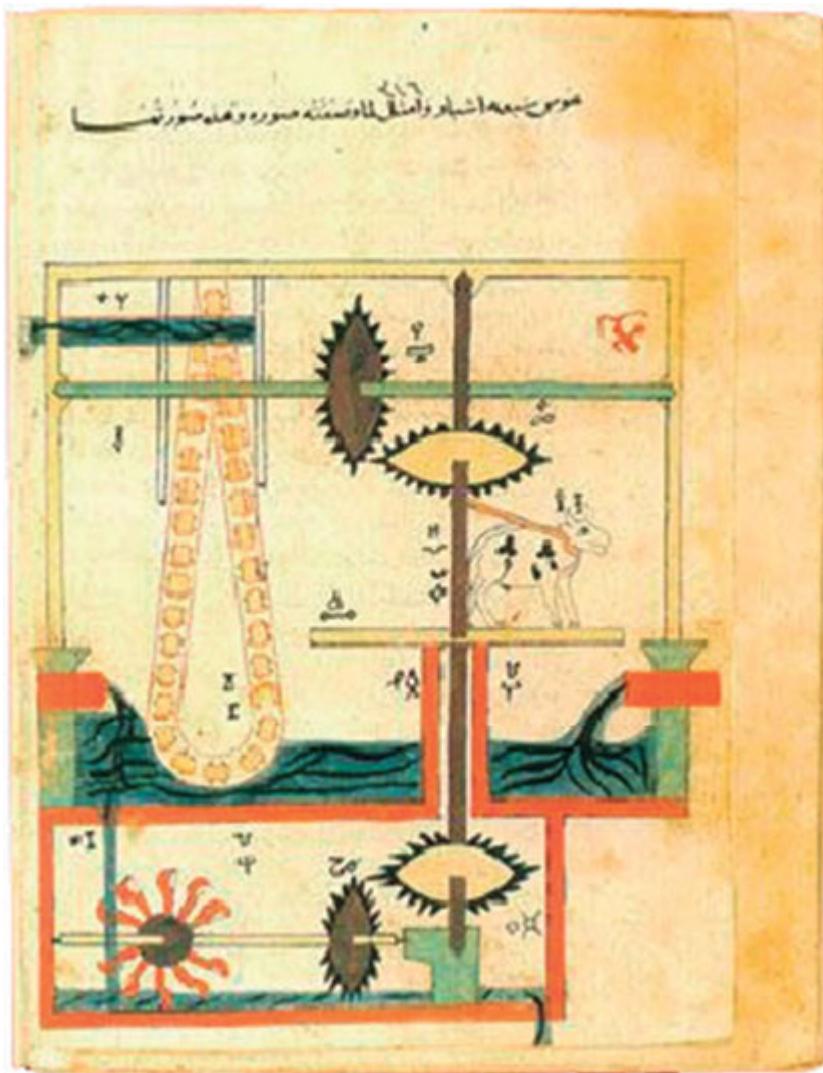


Figura 1.15: Esquema de um dos autômatos de Al-Jazari.<sup>16</sup>

## 1.14 Ars Magna de Ramon Llull – 1305

O monge catalão Ramon Llull (Mallorca, 1232-1315) projetou a partir de 1275 e publicou finalmente em 1305, na obra *Ars Magna* ou *Ars Generalis Ultima*, um mecanismo lógico cujo objetivo era combinar ideias religiosas e filosóficas. Esse mecanismo seria usado para debater com muçulmanos e, vencendo-os no debate, convertê-los à fé cristã.

A inovação radical introduzida por Llull foi a construção de uma máquina de papel que combinava elementos de raciocínio e linguagem. Seguindo o caminho traçado por figuras geométricas inscritas em seu mecanismo, Llull acreditava ser capaz de reproduzir todas as possíveis declarações lógicas que a mente humana fosse capaz de conceber (uma ideia que antecedeu em mais de 600 anos a Máquina de Turing, definida como a máquina capaz de computar tudo o que pode ser efetivamente computável).

Llull entrou para a história da computação por ter sido o primeiro a tentar fazer deduções lógicas usando um mecanismo físico e não a mente. Ele demonstrou que o pensamento humano pode ser imitado por uma máquina. Sua máquina não operava com números, mas com palavras, como se pode observar na [Tabela 1.1](#).

**Tabela 1.1: Elementos combináveis de raciocínio da Ars Magna**

	Princípios Absolutos	Princípios relativos	Questões	Sujeitos	Virtudes	Vícios
<b>B</b>	Bondade	Diferença	Se	Deus	Justiça	Avareza
<b>C</b>	Grandezza	Concorda	O que	Anjo	Prudência	Gula
<b>D</b>	Eternidade	Oposição	De que lugar	Céu	Fortaleza	Luxúria
<b>E</b>	Poder	Prioridade	Quais	Homem	Temperança	Orgulho
<b>F</b>	Sabedoria	Central	Quantos	Imaginação	Fé	Preguiça
<b>G</b>	Vontade	Finalidade	Que tipo	Sensação	Esperança	Inveja
<b>H</b>	Virtude	Maioria	Quando	Vegetativo	Caridade	Fúria
<b>I</b>	Verdade	Igualdade	Onde	Elemento	Paciência	Falsidade
<b>K</b>	Glória	Minoria	Como	Instrumento	Piedade	Inconstância

Llull publicou quatro figuras, cada uma das quais indicava diferentes formas de usar os discos para combinar palavras. A primeira figura ([Figura 1.16](#)) combinava os princípios absolutos, como por exemplo, “sabedoria é verdade”. A segunda figura combinava dois sujeitos por um princípio relativo, como por exemplo “o céu se opõe aos elementos”. A terceira figura permitia criar perguntas combinando

questões, princípios absolutos e relativos, como por exemplo, “qual poder é prioritário?”. Já a quarta figura permitia ler asserções como por exemplo “o homem concorda com a justiça”.



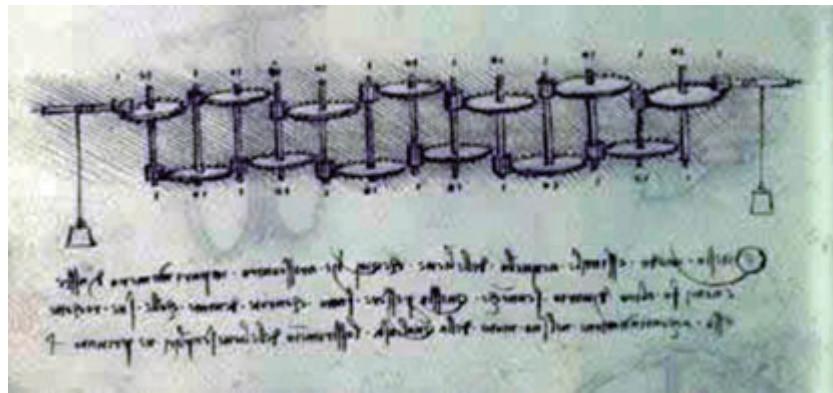
Figura 1.16: Primeira figura da Ars Magna de Ramon Llull.<sup>17</sup>

Acredita-se que Llull possa ter se inspirado nas combinações místicas praticadas na cabala hebraica e também na zairja árabe para construir seu mecanismo.

## 1.15 A Controversa “Calculadora” de Leonardo da Vinci – 1499

Um dos muitos manuscritos de Leonardo da Vinci (Itália, 1452-1519) ficou desconhecido até 1967 porque tinha sido armazenado no local errado na Biblioteca Nacional da Espanha. Por puro acaso, o professor americano Dr. Julius Piccus, que estava à procura de baladas medievais, deparou-se com dois manuscritos, compostos por quase 700 páginas, hoje conhecidos como *Codex Madri I* e *Codex Madri II*. O verso da página 36 do *Codex Madri I* mostrava uma curiosa figura que muito se parecia com uma calculadora

mecânica baseada em engrenagens. Esse desenho é reproduzido na [Figura 1.17](#).



**Figura 1.17: Suposto projeto de uma calculadora diagramado por Leonardo da Vinci.<sup>18</sup>**

O desenho mostra 13 eixos, cada um dos quais contendo duas engrenagens, sendo que a menor tem  $1/10$  da circunferência da maior. Da forma como estão arranjadas, a cada volta da engrenagem à direita, a engrenagem seguinte faria 10 voltas, a terceira 100, a quarta 1000 e assim por diante, até a 13<sup>a</sup>, que faria  $10^{13}$  voltas. Se construída, tal máquina poderia ser usada para realizar cálculos de forma semelhante à calculadora de Schickard, projetada mais de 100 anos depois.

Em 1951, o Dr. Roberto A. Guatelli (Itália, 1904-1993) foi contratado pela IBM para construir réplicas de mecanismos de Leonardo da Vinci. Em 1967, após a descoberta dos *Codex Madri*, ele construiu uma réplica para o suposto mecanismo calculador.

Ela foi exibida pela IBM em 1968. A legenda indicava que se tratava de uma calculadora projetada por Leonardo da Vinci. Houve muita controvérsia e afirmou-se que o Dr. Guatelli usou de muita imaginação para concluir que tal máquina fosse mesmo uma calculadora. Entre outras objeções pode-se citar:

- Os desenhos de Leonardo não indicavam quaisquer dígitos ou números, o que seria obrigatório para uma calculadora.
- Não foi indicada por da Vinci a forma como a máquina deveria operar.
- Não foi indicado um meio de fazer as engrenagens pararem em posições discretas, o que seria obrigatório para uma calculadora.
- O desenho de Leonardo mostra dois pesos amarrados nas duas pontas do sistema, o que seria inútil em uma calculadora.
- Cálculos com 13 dígitos seriam absurdos na Renascença. Nenhum problema real da época exigia números tão grandes.

Some-se a isso o fato de que devido à enorme fricção entre as engrenagens, tal mecanismo jamais poderia ter sido construído com a tecnologia disponível no tempo de Leonardo. Claro que para o inventor do helicóptero e do paraquedas isso não teria sido problema. Mas, no final, a conclusão foi de que de fato o projeto de Leonardo era de um multiplicador de força e não uma calculadora de números.

## 1.16 Logaritmos – 1614

Com o renascimento da cultura e ciência europeia no século XVI, e também devido às grandes navegações, os astrônomos passaram a dedicar mais atenção aos fenômenos celestiais e sua medição precisa. Essa precisão fazia com que necessitassem operar com números muito grandes. Fazer contas manualmente com valores extensos pode ser demorado, além de sujeito a erros. Assim, ferramentas foram criadas para ajudar a simplificar ou mesmo mecanizar parte desses cálculos.

Uma destas ferramentas, já usada no século XVI era a prostaférese, uma técnica que permitia simplificar o cálculo de uma multiplicação fazendo-se apenas duas somas, uma subtração, uma divisão por 2

e algumas olhadas em uma tabela com valores de cossenos. A técnica se baseava na seguinte identidade:

$$\cos(x) \times \cos(y) = \frac{\cos(x+y) + \cos(x-y)}{2}$$

Assim, se você quisesse multiplicar dois números quaisquer  $x$  e  $y$ . Bastaria executar os seguintes passos:

1. Encontrar na tabela de cossenos os valores  $A$  e  $B$  tal que  $x=\cos(A)$  e  $y=\cos(B)$ .
2. Calcular  $p=A+B$  e  $q=A-B$ .
3. Encontrar na tabela os valores de  $\cos(p)$  e  $\cos(q)$ .
4. Somar  $\cos(p)+\cos(q)$  e dividir por 2 para obter o resultado (aproximado) de  $x \times y$ .

Outra ferramenta que simplificava ainda mais as multiplicações e divisões foram os logaritmos, criados de forma independente quase que simultaneamente por John Napier (Escócia, 1550-1617) e Joost Bürgi (Suíça, 1552-1632). Eles descobriram uma forma ainda mais simples de multiplicar números grandes a partir da seguinte identidade:

$$\log(x \times y) = \log(x) + \log(y)$$

Isso reduziria o procedimento de multiplicação de dois números  $x$  e  $y$  aos seguintes passos:

1. Encontre na tabela de logaritmos os valores  $A=\log(x)$  e  $B=\log(y)$ .
2. Some  $A$  e  $B$ .
3. Encontre na tabela o valor do qual  $A+B$  é o logaritmo. Esse valor equivale a  $x \times y$ .

O logaritmo de um número é o expoente ao qual outro número (uma base fixa) deve ser elevado para gerar o número. Por exemplo, considerando uma base 10, o logaritmo de 1.000 é 3 porque  $10^3=1.000$ .

A identidade usada acima para transformar multiplicações em adições baseia-se no fato de que  $x^n \times x^m = x^{n+m}$ ; por exemplo,  $3^2 \times 3^3 = 3^{2+3}$ . Resolvendo a fórmula obtemos  $9 \times 27 = 3^5 = 243$ .

Essa técnica foi implementada fisicamente nas régulas de cálculo, conforme explicado adiante. Além disso, a construção de tabelas numéricas como as de logaritmos foi se tornando mais importante à medida que cálculos cada vez mais complexos se tornavam necessários para resolver problemas de engenharia, astronomia, economia e navegação. O cálculo confiável de tais tabelas gerou a necessidade de máquinas que as pudesse calcular automaticamente, o que levou, em última análise, à invenção dos computadores.

## 1.17 Bastões de Napier – 1617

John Napier (Escócia, 1550-1617) inventou um mecanismo de cálculo baseado em varetas, que permitia fazer multiplicações e divisões de forma quase automática. Esse mecanismo é conhecido como “bastões de Napier” ou “ossos de Napier”, visto que a versão original era feita de marfim.

O instrumento consiste de dez bastões, cada um deles com os valores correspondentes à tabuada de 1 a 10, conforme pode ser visto na [Figura 1.18](#).

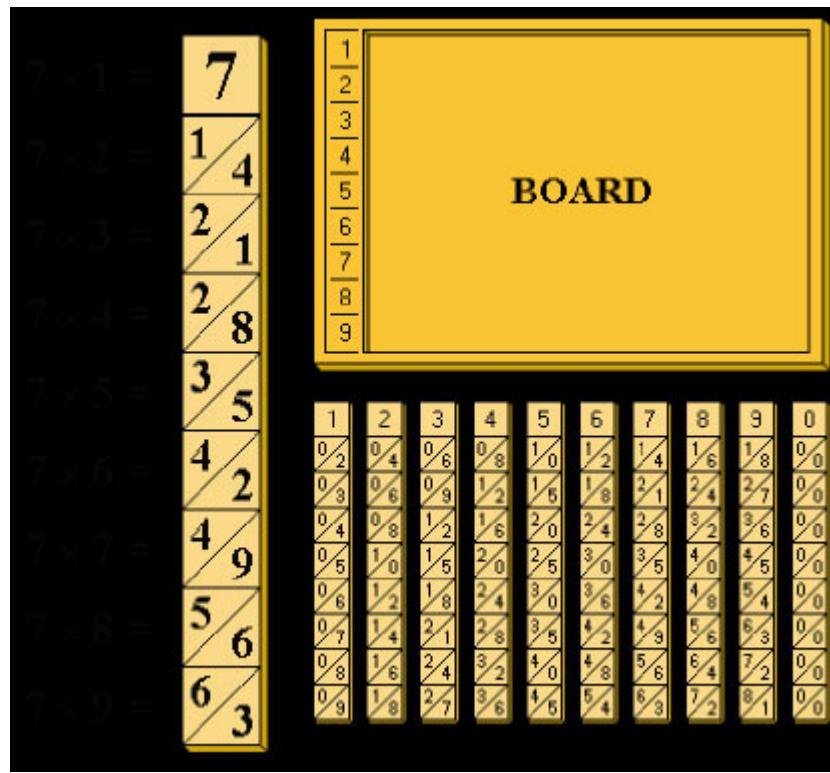
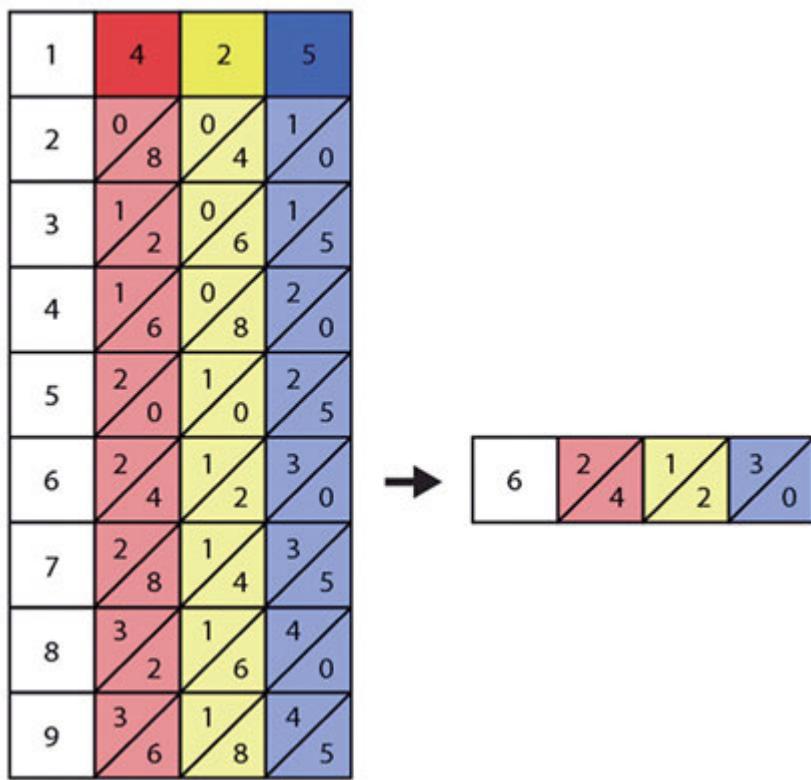


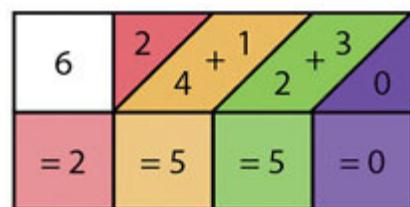
Figura 1.18: Bastões de Napier.<sup>19</sup>

O procedimento para executar uma multiplicação é muito parecido com o que até hoje se aprende nas escolas, mas dispensa saber a tabuada de cor: basta alinhar os bastões e somar os números nas diagonais. A título de exemplo, mostramos como multiplicar 425 por 6. A [Figura 1.19](#) mostra que as varetas correspondentes aos dígitos 4, 2 e 5 foram tomadas e alinhadas. Neste ponto pode-se multiplicar 425 por qualquer número entre 0 e 9, mas escolhemos a linha correspondente ao número 6 para multiplicar 425 por 6.

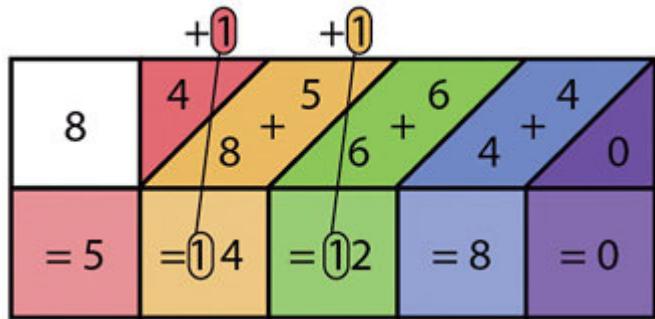


**Figura 1.19:** Alinhamento dos bastões de Napier para multiplicação do número 425 por 6.<sup>20</sup>

Na sequência, basta somar os números nas diagonais, conforme mostrado na [Figura 1.20](#), onde se chega à conclusão de que o resultado é 2550. Se alguma das somas passar de 9, o “vai um” ou “carry”, é somado na diagonal imediatamente à esquerda, conforme mostrado na [Figura 1.21](#).



**Figura 1.20:** Soma-se os números nas diagonais para obter o produto de 425 por 6.<sup>21</sup>



**Figura 1.21:** Multiplicação com “vai um”:  $6785 \times 8$ .<sup>22</sup>

Segundo consta, Napier criou estes bastões para ajudá-lo a realizar os cálculos tediosos que eram necessários para gerar tabelas de logaritmos, as quais, por sua vez, seriam utilizadas para simplificar operações matemáticas como multiplicações, que a partir de então poderiam ser feitas quase que instantaneamente com réguas de cálculo.

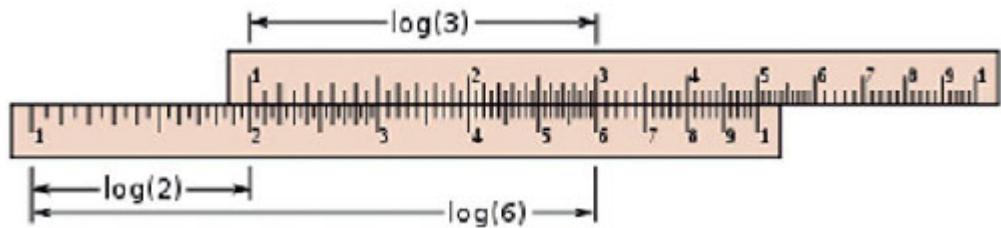
Os bastões de Napier também fizeram parte do projeto da primeira calculadora mecânica conhecida: a de Schickard. E eram usados em conjunto com um mecanismo que operava mecanicamente as adições indicadas pelos bastões.

## 1.18 Régua de Cálculo – 1622

A invenção da régua de cálculo é atribuída a William Oughtred (Inglaterra, 1574-1660), e isso ocorreu logo após a invenção dos logaritmos por Napier e da escala logarítmica por Edmund Gunter (Inglaterra, 1581-1626). Oughtred teria sido a primeira pessoa a fazer multiplicações e divisões simplesmente deslizando uma régua com escala logarítmica sobre outra.

A primeira régua de cálculo teria sido circular. Poucos anos mais tarde, a forma retangular mais conhecida passou a ser também empregada. A [Figura 1.22](#) mostra uma régua de cálculo posicionada para mostrar o resultado da multiplicação de 2 por 3. Posiciona-se o 1 da régua superior para coincidir com o 2 da régua inferior. Assim,

a posição da régua inferior que corresponder ao 3 será o resultado do produto de 2 por 3.



**Figura 1.22:** Forma simplificada de uma régua de cálculo mostrando a multiplicação de 2 por 3.<sup>23</sup>

Já a [Figura 1.23](#) mostra como realizar uma divisão, no caso  $5,5/2$ . Posiciona-se o valor 5,5 na régua inferior coincidindo com o valor 2 da régua superior. O ponto da régua inferior que coincidir com o 1 da régua superior apresentará o resultado: no caso 2,75.



**Figura 1.23:** Régua de cálculo mostrando a divisão de 5,5 por 2.<sup>24</sup>

Ao longo da história, as réguas de cálculo foram constantemente aperfeiçoadas para permitir a realização de operações cada vez mais complexas. Elas estiveram em uso intenso por parte de profissionais como engenheiros, físicos e matemáticos até os anos 1970, quando foram finalmente substituídas pelas calculadoras eletrônicas.

## 1.19. Até Aqui...

Até este ponto, em 1622, a Humanidade desenvolveu algumas formas primitivas de representação de informação e certos

instrumentos simples para cálculo com números. Enquanto a matemática era predominantemente usada para transações comerciais, a maioria dos problemas podiam ser resolvidos com poucas operações aritméticas simples. Mas com o crescimento do interesse pela astronomia, engenharia e economia, e a necessidade cada vez maior de construir tabelas de números para aplicações nessas áreas, a quantidade de cálculos necessários para dar conta dessas exigências cresceu de forma astronômica.

A dificuldade na realização de cálculos repetitivos com precisão e correção levou alguns pesquisadores a pensar se não seria possível automatizar a tarefa. A invenção do zero, dos algarismos arábicos e dos autômatos, aliados ao conceito de algoritmo para realização de cálculos, acabou possibilitando essa evolução. Isso nos leva à era seguinte, com a construção e aperfeiçoamento das primeiras calculadoras mecânicas.

---

<sup>1</sup> By Sannita - Own work by uploader. This vector image was created with Inkscape. CC BY-SA 3.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=7108704, 7108771, 7108796, 7108816, 7108831>

<sup>2</sup> Adaptado a partir da fonte da Figura 1.1.

<sup>3</sup> Adaptado (recortado) By Ben2 - Own work, CC BY-SA 3.0, Disponível em: <https://commons.wikimedia.org/w/index.php?curid=3166443>

<sup>4</sup> Adaptado (recortado): By Own photograph by Sandstein, CC BY 3.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=8725556>

<sup>5</sup> ©Jörn Lütjens. Cortesia. Disponível em: <http://www.joernluetjens.de/sammlungen/abakus/rom-abakus-en.htm>

<sup>6</sup> Precessão dos equinócios: movimento do eixo da Terra que faz a orientação do eixo polar do planeta apontar para uma constelação zodiacal diferente aproximadamente a cada dois mil anos.

<sup>7</sup> © François Blateyron. Cortesia. Disponível em:  
<http://www.shadowspro.com/en/planispheric.html>

<sup>8</sup> Azimute: o ângulo formado entre a projeção de uma posição no céu até o horizonte e a direção do norte geográfico.

<sup>9</sup> Culminação: o momento em que um astro passa pelo meridiano local.

<sup>10</sup> “NAMA Machine d’Anticythère 1”. Licensed under CC BY 2.5 via Wikimedia Commons – Disponível em: [https://commons.wikimedia.org/wiki/File:NAMA\\_Machine\\_d%27Anticyth%C3%A8re\\_1.jpg#/media/File:NAMA\\_Machine\\_d%27Anticyth%C3%A8re\\_1.jpg](https://commons.wikimedia.org/wiki/File:NAMA_Machine_d%27Anticyth%C3%A8re_1.jpg#/media/File:NAMA_Machine_d%27Anticyth%C3%A8re_1.jpg)

<sup>11</sup> “Antikythera mechanism” by Lead Holder - File: Meccanismo\_di\_Antikytera.jpg. Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Antikythera\\_mechanism.svg#/media/File:Antikythera\\_mechanism.svg](https://commons.wikimedia.org/wiki/File:Antikythera_mechanism.svg#/media/File:Antikythera_mechanism.svg)

<sup>12</sup> ©2012 Tony Freeth, Images First. Cortesia. Disponível em: <http://www.images-first.com/>

<sup>13</sup> “Brahmagupta” by anonymus – Disponível em: [http://pioneros.puj.edu.co/biografias/edad\\_media/560\\_860/brahmagupta.html](http://pioneros.puj.edu.co/biografias/edad_media/560_860/brahmagupta.html). Licensed under Public Domain via Wikimedia Commons – <https://commons.wikimedia.org/wiki/File:Brahmagupta.jpg#/media/File:Brahmagupta.jpg>

<sup>14</sup> “Dixit algorizmi” por Muhammad ibn Musa al-Khwarizmi - scanned from facsimile (1963). Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Dixit\\_algorizmi.png#/media/File:Dixit\\_algorizmi.png](https://commons.wikimedia.org/wiki/File:Dixit_algorizmi.png#/media/File:Dixit_algorizmi.png)

<sup>15</sup> Muslim Heritage. Cortesia. Fonte: Chaarani, M. S. L’orgue hydraulique des Banu Mûsa. Disponível em: <http://www.muslimheritage.com/article/l%CE%80%99orgue-hydraulique-des-banu-m%C3%BBsa-hydraulic-organ-banu-musa>

<sup>16</sup> “Al-Jazari Automata 1205”. Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Al-Jazari\\_Automata\\_1205.jpg#/media/File:Al-Jazari\\_Automata\\_1205.jpg](https://commons.wikimedia.org/wiki/File:Al-Jazari_Automata_1205.jpg#/media/File:Al-Jazari_Automata_1205.jpg)

<sup>17</sup> “Ramon Llull - Ars Magna Fig 1”. Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Ramon\\_Llull\\_-\\_Ars\\_Magna\\_Fig\\_1.png#/media/File:Ramon\\_Llull\\_-\\_Ars\\_Magna\\_Fig\\_1.png](https://commons.wikimedia.org/wiki/File:Ramon_Llull_-_Ars_Magna_Fig_1.png#/media/File:Ramon_Llull_-_Ars_Magna_Fig_1.png)

<sup>18</sup> “Máquina de somar de Leonardo da Vinci” by Leonardo da Vinci - Disponível em: [http://www.thcopc.net/hardware/da\\_vinci\\_replica.htm](http://www.thcopc.net/hardware/da_vinci_replica.htm). Licenced under Public Domain via Wikimedia Commons. <https://commons.wikimedia.org/wiki/>

[File:M%C3%A1quina\\_de\\_sumar\\_de\\_Leonardo\\_da\\_Vinci.jpg#/media/File:M%C3%A1quina\\_de\\_sumar\\_de\\_Leonardo\\_da\\_Vinci.jpg](#)

<sup>19</sup> “Bones of Napier (board and rods)”. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:Bones\\_of\\_Napier\\_\(board\\_and\\_rods\).png#/media/File:Bones\\_of\\_Napier\\_\(board\\_and\\_rods\).png](#)

<sup>20</sup> “Napier’s Bones ex 1 pic 2” by Mopedmeredith - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Napier%27s\\_Bones\\_ex\\_1\\_pic\\_2.png#/media/File:Napier%27s\\_Bones\\_ex\\_1\\_pic\\_2.png](#)

<sup>21</sup> “Napier’s Bones ex 1 pic 3” by Mopedmeredith - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Napier%27s\\_Bones\\_ex\\_1\\_pic\\_3.png#/media/File:Napier%27s\\_Bones\\_ex\\_1\\_pic\\_3.png](#)

<sup>22</sup> “Napier’s Bones ex 2 pic 3” by Mopedmeredith - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Napier%27s\\_Bones\\_ex\\_2\\_pic\\_3.png#/media/File:Napier%27s\\_Bones\\_ex\\_2\\_pic\\_3.png](#)

<sup>23</sup> “Slide rule example2 with labels” by Jakob.scholbach - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Slide\\_rule\\_example2\\_with\\_labels.svg#/media/File:Slide\\_rule\\_example2\\_with\\_labels.svg](#)

<sup>24</sup> “Slide rule example4” by Wrtlprnft, original image made by Benjamin Crowell - This is an SVG version of en:Image:Slide rule example4.jpg. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Slide\\_rule\\_example4.svg#/media/File:Slide\\_rule\\_example4.svg](#)

## **PARTE II**

### **Surgimento das Calculadoras Mecânicas**

Os séculos XVII, XVIII e XIX viram o surgimento e aperfeiçoamento das calculadoras mecânicas. A partir de trabalhos inovadores como os de Schickard, Pascal e Leibniz, máquinas capazes de realizar as quatro operações aritméticas com o simples girar de uma alavanca se tornaram realidade. Essas máquinas, bem como o tear mecânico, que usava cartões perfurados já no início do século XIX, foram fundamentais para a concepção posterior dos computadores de propósito geral – ou seja, máquinas programáveis para executar qualquer função computável e não apenas as quatro operações. Esse mesmo período também testemunhou o desenvolvimento da aritmética binária e o surgimento dos primeiros computadores humanos, ou seja, pessoas cuja profissão era executar cálculos repetitivos à mão.

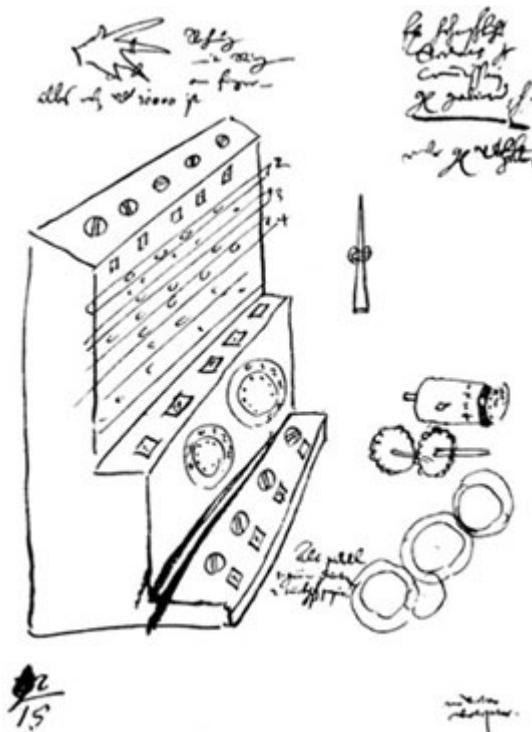
#### **2.1 Relógio Calculador de Schickard – 1623**

Durante quase 300 anos a calculadora de Pascal foi considerada como o primeiro projeto bem-sucedido de calculadora mecânica automática. Porém, quando duas cartas de William Schickard (Alemanha, 1592-1635) para Johannes Kepler (Alemanha, 1571-1630) mencionando o projeto de uma máquina calculadora (*rechenuhr*) foram redescobertas entre os pertences de Kepler, em pleno século XX, percebeu-se que o projeto de Schickard antecedia o de Pascal em cerca de 20 anos. Essas cartas de 1623 e 1624 já haviam sido já mencionadas em publicações nos séculos XVIII e

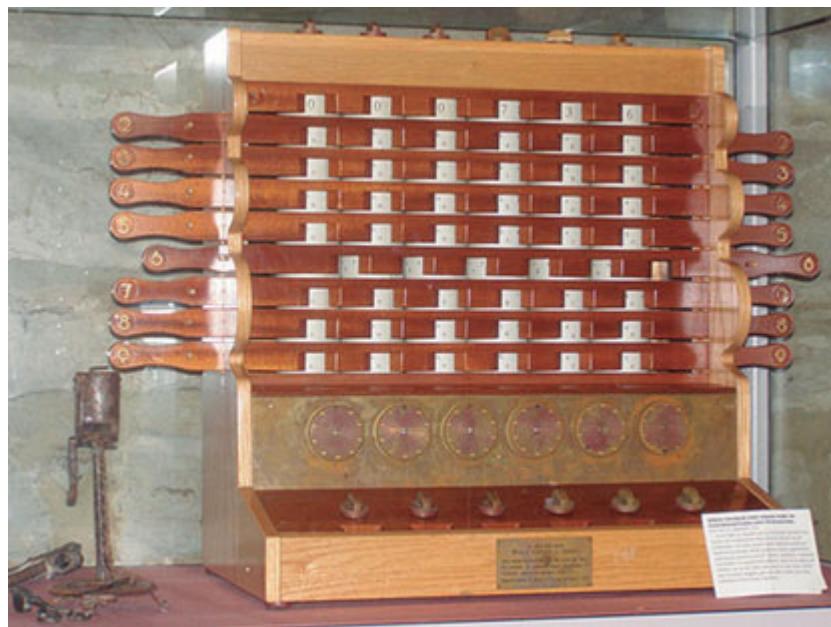
XIX, mas até 1957 ninguém havia percebido esse fato histórico tão importante.

Assume-se, pelo descrito nas cartas, que Schickard teria construído duas máquinas: um protótipo para uso próprio e outra mais bem-acabada que ele daria de presente a Kepler. A segunda máquina foi encomendada a um artesão, mas pouco antes de ser concluída e entregue acabou destruída em um incêndio noturno. Já a máquina pessoal de Schickard possivelmente também foi perdida, visto que a região onde morava foi envolvida em uma guerra e sua casa e todos os bens foram queimados após sua morte, causada pela peste bubônica.

Nada restou do trabalho de Schickard a não ser alguns papéis mantidos na biblioteca de Stuttgart e as duas cartas a Kepler. Um diagrama da máquina incluído em uma das cartas é apresentado na [Figura 2.1](#), e uma reprodução física da mesma na [Figura 2.2](#).



**Figura 2.1:** Um esboço da calculadora de Schickard em uma das cartas a Kepler.<sup>1</sup>



**Figura 2.2: Reprodução da calculadora de Schickard.<sup>2</sup>**

Assim como o astrolábio e o mecanismo de Anticítera, a máquina de Schickard foi planejada para fazer cálculos astronômicos (daí sua relação com Kepler, famoso astrônomo). Em 1623, ele escreveu o seguinte para o amigo: “O que você fez com cálculos eu consegui através da mecânica. Eu concebi uma máquina consistindo de onze engrenagens completas e seis incompletas; ela calcula instantaneamente e automaticamente adições, subtrações, multiplicações e divisões, a partir dos números dados. Você iria adorar ver como a máquina faz acumulações e transporta espontaneamente as dezenas e centenas para a esquerda e em reverso, como ela faz o oposto quando subtrai (...).”<sup>3</sup>

A máquina de Schickard era composta por duas partes. A parte superior era uma versão modificada dos bastões de Napier. Os números correspondentes a cada bastão eram engenhosamente gravados em seis cilindros giratórios na frente dos quais 10 varetas com uma pequena abertura eram colocadas. As varetas podiam ficar em duas posições: à direita, não permitiam visualizar nenhum número, mas à esquerda a abertura permitia visualizar um número

correspondente à multiplicação da posição correspondente no cilindro e na vareta. Por exemplo, para multiplicar 532 por 7 o usuário deveria girar os três cilindros mais à direita para as posições “5”, “3” e “2”, e mover a vareta número 7 para a esquerda. As aberturas permitiriam então visualizar os valores “35”, “21” e “14”, correspondendo aos produtos de 5, 3 e 2 por 7. Assim, a parte superior da calculadora simplificava o processo de multiplicação pelo uso da técnica dos bastões de Napier. Restava a soma das parcelas.

Para realizar a soma das parcelas, Schickard construiu um sistema de engrenagens que permitia que números fossem representados através do giro de discos. Cada disco permitia representar os dígitos de 0 a 9. Quando um disco girava além da posição “9” para a posição “0”, uma engrenagem de um único dente faria a engrenagem à esquerda girar uma posição (por exemplo, se estivesse em 3, ela iria para 4). Assim, a cada dez giros, o disco imediatamente à esquerda girava uma vez (realizando desta forma o famoso “carry” ou “vai um” das operações aritméticas).

Assim, para somar, por exemplo 35 com 67, o usuário deveria girar o disco mais à direita para 5 e o seu vizinho da esquerda para 3; depois deveria girar o disco mais à direita mais 7 posições, o que deixaria este disco na posição “2” e, na passagem de 9 para 0, isso faria o disco imediatamente à esquerda girar mais uma posição, ficando assim, portanto, na posição “4”. Para finalizar, o usuário deveria girar o disco da esquerda mais seis posições, ficando assim na posição “0”, e fazendo seu vizinho da esquerda girar uma posição, indo para “1”. O resultado 102 seria então visualizado nos mostradores da máquina.

Desta forma, seria relativamente fácil somar várias sequências de números, apenas girando discos, e a máquina acumularia o resultado em seus mostradores. Schickard teve até o cuidado de instalar uma campainha no disco mais à esquerda, que tocaria caso houvesse *overflow*, ou seja, se a capacidade da máquina fosse

excedida. Com seis discos, a calculadora seria limitada a representar números de 0 a 999.999.

A subtração é obtida girando os discos no sentido inverso. A divisão deveria seguir o processo proposto por Napier. A máquina ainda tinha um conjunto independente de discos para que resultados intermediários pudessem ser armazenados pelo usuário enquanto este continuasse a fazer novas operações.

Uma das maiores desvantagens do sistema de Schickard é que quando várias engrenagens precisassem realizar o “vai um”, por exemplo, ao somar 99.999 com 1, o esforço necessário para mover o disco mais à direita seria bastante grande, exceto se o mecanismo utilizado estivesse muito bem lubrificado. Mesmo assim, a escalabilidade desse sistema para números mais longos seria inviável com a tecnologia da época.

## 2.2 Pascalina – 1642

Blaise Pascal (França, 1623-1662) provavelmente nunca teve acesso ao trabalho de Schickard. Entretanto, cerca de 20 anos depois e com apenas 19 anos de idade, ele conseguiu reinventar o conceito de calculadora mecânica e foi considerado por quase 300 anos como pioneiro na área. De fato, ele teve o mérito de ter criado a primeira calculadora que funcionava de forma robusta, além de ter obtido um registro equivalente a uma patente moderna (um “privilégio real”, concedido por Luiz XIV em 1649).

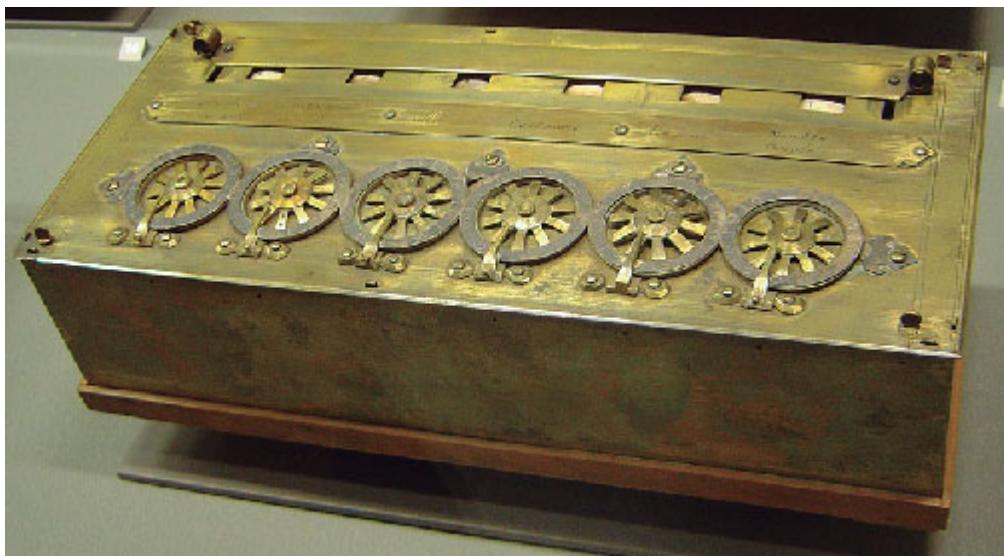


Figura 2.3: Uma Pascalina.<sup>4</sup>

Pascal teria construído 20 dessas máquinas após testar cerca de 50 protótipos com diferentes designs. Ele foi capaz de vender algumas, mas o alto custo não permitiu que o projeto fosse muito longe. Consta que nove dessas calculadoras ainda existem e estão em exposição em museus.

A máquina de Pascal tinha muitas semelhanças com a parte inferior da máquina de Schickard. Porém, Pascal resolveu o problema do atrito do “vai um” em cadeia inserindo no seu projeto uma pequena mola que fazia mover a engrenagem da esquerda quando fosse necessário passar o dígito de *carry*. Como cada engrenagem tinha a própria mola que gerava o movimento, o problema do atrito não se propagava de uma engrenagem para outra. Isso permitiria que não apenas cinco ou seis engrenagens fossem usadas, mas qualquer quantidade delas já que cada uma teria sua própria força motriz sem depender das anteriores.

Infelizmente esse mecanismo também impedia que as engrenagens girassem no sentido inverso, o que dificultava as subtrações. Pascal solucionou a questão utilizando a técnica de subtrair pela soma do número complementar. A janela que mostrava os valores na

máquina podia ser movida de forma que fossem exibidos os números normais ou seus complementares.

O complemento de um número depende de quantos dígitos se considera. Com uma máquina de cinco dígitos, o complemento de  $n$  seria  $99.999-n$ . Por exemplo, o complemento de 89 seria  $99.999 - 89 = 99.910$ . Assim, para subtrair, por exemplo,  $89 - 35$  poderíamos somar  $99.910 + 35 = 99.945$ . O complemento deste número ( $99.999 - 99.945$ ) é o resultado da subtração: 54.

Para somar dois números ou mais a operação da máquina era muito simples. Primeiramente o operador devia se certificar de que a máquina estava zerada (todos os mostradores na posição “0”). Em seguida o primeiro número podia ser inserido. Cada botão giratório tinha 10 raios, como uma roda de carroça. Ao redor dos botões estavam inscritos os dígitos de 0 a 9. Havia um ponto de parada, semelhante aos antigos telefones de discar. Então, por exemplo, para inserir um “5” o operador colocava o dedo ou um apontador na posição “5” e girava a roda até a posição de parada. Nesse momento, o mostrador da máquina estaria exibindo “00005”. Se em seguida o operador quisesse adicionar 6, deveria colocar o apontador na posição “6” e girar a roda até a posição de parada. Neste momento o mostrador estaria exibindo “00011”.

Como as engrenagens não podiam ser giradas no sentido inverso, para zerar a máquina era necessário colocar todos os mostradores na posição 9 (ou seja, fazer a máquina exibir “99.999”) e em seguida somar 1 girando o botão mais à direita em uma posição.

Um pequeno filme disponível em [www.youtube.com/watch?v=3h71HAJWnVU](https://www.youtube.com/watch?v=3h71HAJWnVU) ilustra com perfeição o funcionamento desta máquina.

## 2.3 Leviatã de Thomas Hobbes – 1651

O filósofo Thomas Hobbes (Inglaterra, 1588-1679) é considerado por muitos como o fundador da inteligência artificial. Em seu livro

*Leviatā* ([Figura 2.4](#)), de 1651, ele afirma o seguinte: “*By ratiocination, I mean computation.*”<sup>5</sup>

Esta frase é particularmente importante por trazer à discussão filosófica duas grandes ideias:

- Pensar consiste em realizar operações simbólicas, ou seja, é como trabalhar cálculos com lápis e papel, exceto que isso é feito internamente.
- O pensamento é mais claro e racional quando ele segue regras rígidas como as que os contadores usam quando fazem cálculos, ou seja, o raciocínio é um processo mecânico, como a operação de um ábaco mental.



Figura 2.4: Frontispício do livro *Leviatã* de Hobbes.<sup>6</sup>

As ideias de Hobbes aparentemente tiveram influência sobre o pensamento de Leibniz e também sobre teorias mais modernas que tentam explicar o funcionamento da mente como um processo de computação.

Hobbes afirma que pensar é como fazer somas e subtrações com a mente. Ele mesmo exemplifica isso argumentando que com a adição da ideia de quadrilátero (um polígono de quatro lados) com a ideia de polígono regular (um polígono com lados iguais) pode-se conceber a ideia de quadrado (um polígono regular com quatro lados).

Ele também afirma que os silogismos nada mais são do que a soma de duas proposições gerando um resultado. Adicionalmente, proposições são a adição de um predicado a um sujeito, por exemplo, somando o sujeito “neve” com o predicado “branca”, tem-se a proposição “a neve é branca”. Ao sugerir que tais combinações podem ser feitas sem a necessidade de uma palavra intermediária na frase (no exemplo, o verbo “é”), ele lança as bases para a notação matemática que mais tarde se tornaria a fundamentação de linguagens de programação como Prolog, na qual se escreveria este predicado como “*branca(neve)*”.

## 2.4 O Método Llullístico de Athanasius Kircher – 1669

Athanasius Kircher (Alemanha, 1602-1680) de certa forma aprimorou o sistema de combinação de ideias de Ramon Llull ao utilizar símbolos no lugar de palavras latinas ([Figura 2.5](#)). Com essa abordagem, Kircher buscava a matematização do método combinatório de Llull.

## LIBRI TERTII.

### PARS I.

#### De Alphabetis eorumque applicatione.

##### CAPUT I.

###### *De Alphabeto Artis Lulliane, ejusque Combinationibus & usu.*

**Q**uamvis Alphabetum Artis nostrae fuisse in primo Libro exposuerimus, quia tamen omnia, quæ tum ad terminos ejus combinandos, tum ad usum ejus insignem & maximum pertinent, ibidem omisimus, hic singula ordine ad majorem Lectoris instructionem per-

nitiū ad mentem Lulli exponenda duximus. Nota Lector, sequens Alphabetum non Lulli, sed nostrum esse. Quomodo vero id regulis Artis à Lullo traditæ applicari possit & debeat ad habitum scientificum faciliori methodo adquirendam, hac parte demonstrabitur.

###### *Alphabetum Artis Magnæ.*

I.	II.	III.	IV.	V.	VI.	VII.	VIII.	IX.
Principia absoluta.	B.	M.	D.	P.	S.	Vo.	Vi.	G.
Respecti- va.	=	♡	∞	α	◎	ω	M	Æ
Quæstio- nes.	An.	Quid.	De quo.	Cur.	Quan- tum.	Quale.	Quomo- do.	Ubi. Qui cumquo.
Subiecta u- niversalia.	△	∅	◎	□	†	🐴	✿	⋮
Virtutes.	Justitia.	Pruden- tia.	Fortitu- do.	Tempe- rantia.	Fides.	Spes.	Charitas.	Patiens.
Vicia.	Avaritia.	Gula.	Luxuria.	Super- bia.	Acedia.	Invidia.	Ira.	Menda- cium.

###### *De Combinatione Alphabeti, & de Tabulis ex eo concinnandis.*

Nota, dupliciter Alphabeti terminos in-  
ter se combinari posse, vel expansè, vel

contradicte. Combinationem expansam ita  
adorieris.

Cum in Alphabeto novem sint literæ: B.M.  
D.P.S.Vo.Vi.Ve. G. quæ novem principia  
absoluta per initiales suas literas significant.

Figura 2.5: O alfabeto simbólico de Kircher.<sup>7</sup>

Kircher publicou em 1669 um vasto tratado sobre a arte de Llull, o *Ars Magna Sciendi, Sive Combinatoria*. No capítulo terceiro do livro ele apresenta essa nova e universal versão do método de Llull para combinação de conceitos.

Entre outras coisas, Kircher tentou calcular todas as possíveis combinações de alfabetos formados com 1, 2, 3, ..., 50 letras diferentes. Com esse conhecimento ele pretendia decifrar os hieróglifos egípcios.

## 2.5 Calculadora de Samuel Morland – 1673

Samuel Morland (Inglaterra, 1625-1695) deixou uma importante contribuição à computação mecânica do século XVII – não tanto pelas três máquinas calculadoras que inventou, mas pela publicação de um extenso tratado sobre elas. Em 1673, Morland publicou um livro ([Figura 2.6](#)) sobre duas de suas calculadoras, uma máquina de somar e uma de multiplicar. Esse foi o primeiro livro sobre calculadoras mecânicas publicado na história, visto que de Schickard restaram apenas as cartas, e que Pascal publicou apenas um pequeno panfleto sobre sua própria calculadora. De fato, até as publicações de Charles Babbage, mais de 150 anos depois, muito pouco foi publicado no mundo sobre esse assunto.

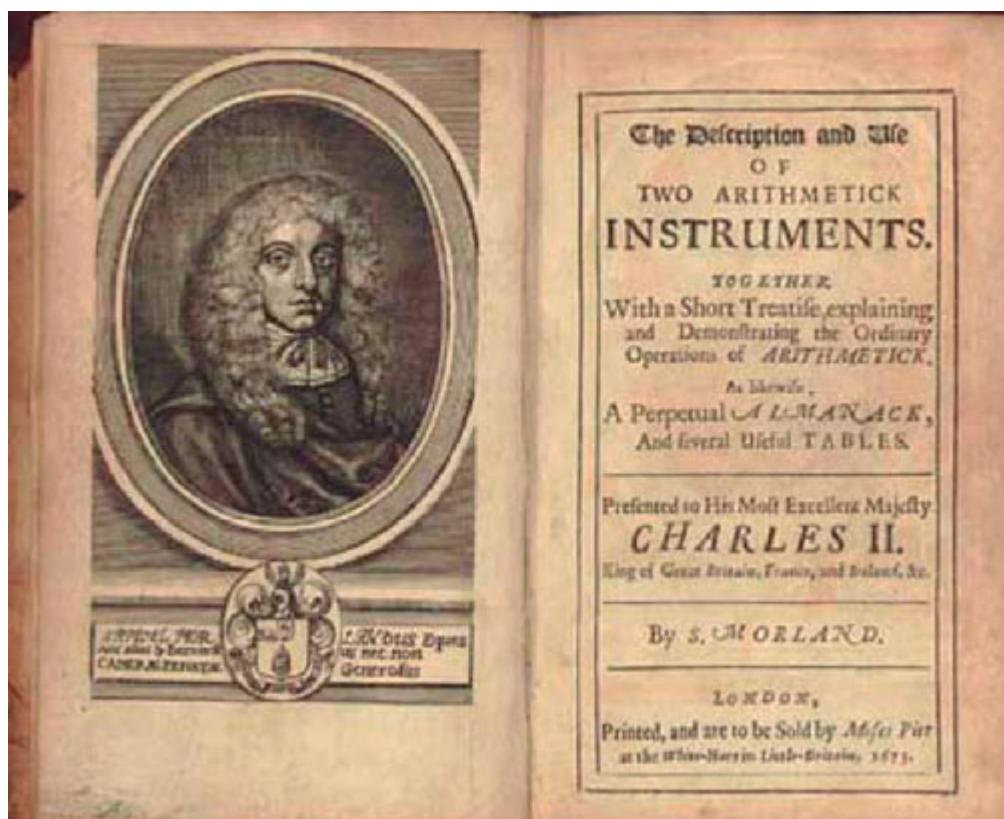


Figura 2.6: O primeiro livro escrito no mundo sobre calculadoras mecânicas.<sup>8</sup>

Menciona-se que Morland conheceu uma pascalina enquanto visitava a corte da rainha da Suécia, o que o inspirou a criar e construir suas próprias máquinas. Seu aparelho de somar era na verdade praticamente do tamanho de uma calculadora de bolso, medindo 12 x 7 cm e possuindo menos de 1 cm de espessura. Ela era construída e podia ser encomendada ao relojoeiro Humphrey Adamson de Londres. A máquina era capaz de lidar com escalas diferentes da decimal, mas ainda assim necessárias naquela época, visto que as moedas inglesas do século XVII não seguiam a divisão decimal: 1 pound valia 20 shillings, 1 shilling valia 12 pences e 1 pence 4 farthings.

Porém, a principal desvantagem da máquina de Morland, especialmente se comparada à mais complexa máquina de Pascal, é que ela não fazia automaticamente a operação de transposição do *carry*, que devia ser realizada manualmente pelo operador quando necessário. Isso simplificava a calculadora em relação à Pascalina e, inclusive, permitia que realizasse subtrações com o giro dos discos no sentido oposto ao da soma, mas também a fazia perder muito do apelo prático, pois deixava de ser realmente automática.

O outro aparelho era uma máquina de multiplicar chamada “ciclológica” ([Figura 2.7](#)), a qual se baseava no método de Napier. Ela era basicamente uma execução mecânica dos bastões do matemático. Provida com discos contendo números de Napier, eles deviam ser colocados em posições específicas (para cada cálculo) e cobertos por uma tampa que se deixava visualizar apenas um dígito de cada vez. Então, uma alavanca era girada fazendo a cada volta os discos avançarem uma posição. Um indicador mostrava quantas vezes ela tinha sido acionada. Assim, por exemplo, para multiplicar 365 por 7, devia-se posicionar os discos de número 3, 6 e 5 na posição apropriada e girar a alavanca sete vezes. Os mostradores exibiriam os números de Napier correspondentes ao produto de 3 por 7, 6 por 7 e 5 por 7, ou seja, 21, 42 e 35. Para obter o resultado, o usuário precisaria somar à mão os dígitos em discos contíguos, isso é  $2[1+4][2+3]5 = 2.555$ .

Essa máquina também não dispunha de mecanismo de *carry*, o que dificultava seu uso prático. Por exemplo, ao multiplicar 28 por 9 os discos vão apresentar  $2 \times 9 = 18$  e  $8 \times 9 = 72$ . Somando-se os dígitos dos discos contíguos obtém-se 1[8+7]2. Neste caso, o resultado fica 1[15]2, e o *carry* precisa ser transferido manualmente para a esquerda, resultando em 252.

A terceira máquina de Morland não era uma calculadora mecânica propriamente dita, mas um instrumento que permitia fazer cálculos trigonométricos e medir o resultado com o uso de réguas.

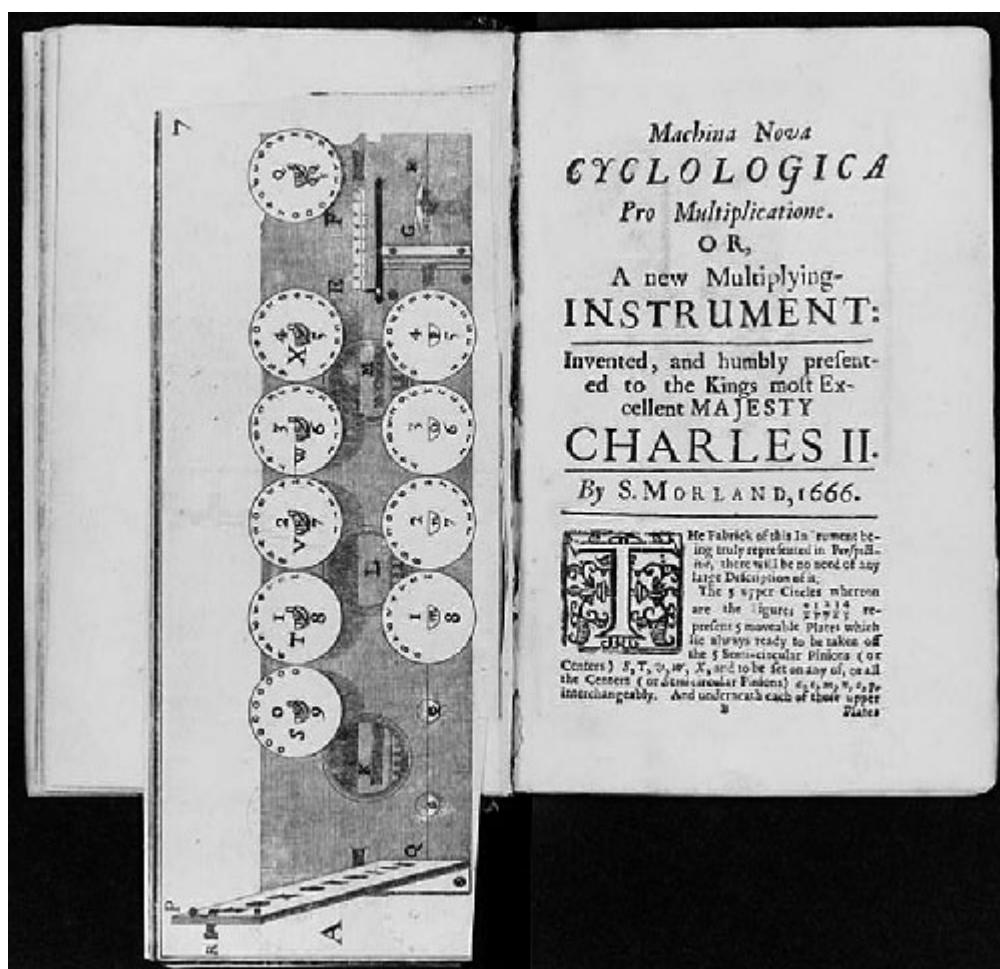


Figura 2.7: Máquina de multiplicar de Morland.<sup>9</sup>

## 2.6 Sistema Binário – 1679

A invenção, ou descoberta, do moderno sistema binário é atribuída a Gottfried Leibniz (Alemanha, 1646-1716). Em seu artigo “*Explication de l’Arithmétique Binaire*” (1679) ele apresenta o sistema binário dizendo que estamos acostumados a representar os números no sistema decimal: progredimos de 0 a 9 e em seguida iniciamos de novo, ou seja, depois de 9, vem 10. Da mesma forma, após 99 temos 100, e assim por diante. Leibniz propõe trabalhar com o sistema mais simples de todos, que tem apenas 2 algarismos: 0 e 1. Depois do 1, começamos de novo. Assim, depois de 1 vem 10, depois 11, depois 100, e assim por diante. Então o 10 binário corresponde ao 2 decimal e o 100 binário, ao 4 decimal.

Leibniz afirma que essa é a forma mais simples e mais pura de entender e manipular os números já que se pode fazer as quatro operações aritméticas de forma muito mais simples, sem a necessidade de saber tabuadas ou resultados de somas de algarismos quaisquer como 6 e 9. Para fazer as quatro operações com aritmética binária, basta saber os fatos representados na [Tabela 2.1](#).

**Tabela 2.1: Os fatos básicos da aritmética binária**

Soma	Subtração	Multiplicação	Divisão
$0+0=0$	$0-0=0$	$0\times0=0$	Se o divisor $y$ tiver $n$ dígitos significativos, tome os primeiros $n$ dígitos do dividendo e chame de $x$ . Assim, se $x \geq y$ então $x \div y = 1$ , senão $x \div y = 0$ .
$0+1=1$	$10-1=1$	$0\times1=0$	
$1+0=1$	$1-0=1$	$1\times0=0$	
$1+1=10$	$1-1=0$	$1\times1=1$	

Por exemplo, para somar 12 e 6 em binário, primeiro construímos a representação binária dos dois números:  $12=1100_{\text{bin}}$  e  $6=110_{\text{bin}}$ . Para somar, segue-se o procedimento usual que aprendemos com o sistema decimal: alinharmos os dois números pelo dígito mais à direita e somarmos os pares um por um conforme a primeira coluna da [Tabela 2.1](#). Quando se somar 1 e 1, o resultado é zero e “vai um” para a coluna imediatamente à esquerda. Os passos são mostrados na [Figura 2.8](#).

Passo 1	Passo 2	Passo 3	Passo 4
		1	1
1100	1100	1100	1100
<u>+110</u>	<u>+110</u>	<u>+110</u>	<u>+110</u>
0	10	010	10010

Figura 2.8: Exemplo de soma binária.

Assim, o resultado  $10010_{\text{bin}}$  corresponde ao número 18.

A [Figura 2.9](#) mostra como fazer uma subtração binária, no caso  $13-6=7$ . Novamente, usa-se o mesmo processo que se usa com números decimais e unicamente as regras mostradas na segunda coluna da [Tabela 2.1](#).

Passo 1	Passo 2	Passo 3
	0	0
1101	1101	1101
<u>-110</u>	<u>-110</u>	<u>-110</u>
1	11	111

Figura 2.9: Exemplo de subtração binária.

A multiplicação binária também segue o mesmo procedimento da multiplicação decimal. Mas a tabuada é muito mais simples, conforme mostrado na terceira coluna da [Tabela 2.1](#). A Figura 2.10 mostra a operação  $13 \times 5 = 65$ . Nos passos em que o número de cima é multiplicado por 1, basta reproduzir o número no resultado; isso pode ser visto nos passos 1 e 3. Nos passos é que o número de cima é multiplicado por zero, basta preencher o resultado com zeros, como no passo 2. O passo 4 consiste em somar as parcelas, produzindo o resultado da multiplicação.

Passo 1	Passo 2	Passo 3	Passo 4
<b>1101</b>	<b>1101</b>	<b>1101</b>	1101
$\times 101$	$\times 101$	$\times 101$	$\times 101$
<b>1101</b>	1101	1101	<b>1101</b>
	<b>0000</b>	0000	<b>0000</b>
		<b>1101</b>	<b>+1101</b>
			<b>1000001</b>

Figura 2.10: Exemplo de multiplicação binária.

A divisão binária é também extremamente simples. Basta fazer como no método decimal, tomando uma parte do dividendo que tenha o mesmo comprimento do divisor da esquerda para a direita. Se a parte do dividendo for maior ou igual ao divisor, o resultado é 1 e o resto é a parte do dividendo menos o divisor. Se a parte do dividendo for menor do que o divisor, o resultado é 0 e o resto é igual ao dividendo.

A Figura 2.11 mostra como dividir 18 por 2 em binário, produzindo como resultado o número 9. No passo 1, tem-se divisão exata  $10 \div 10 = 1$  com resto 0. No passo 2, desce um 0 e a divisão é  $00 \div 10 = 0$  com resto 0. No passo 3, desce o 1 e a divisão é  $01 \div 10 = 0$  com resto 1. Finalmente, no passo 4 desce o último 0 e a divisão é  $10 \div 10 = 1$  com resto 0.

Passo 1	Passo 2	Passo 3	Passo 4
<b>10010 10</b>	<b>10010 10</b>	<b>10010 10</b>	<b>10010 10</b>
0    1	00    10	00    100	00    1001
	0	01	01
		1	10
			0

**Figura 2.11: Exemplo de divisão binária.**

O leitor que possivelmente entenda um pouco de circuitos eletrônicos já poderá perceber como seria muito mais fácil implementar tais operações com portas lógicas do que implementar as operações sobre números decimais. Este é um dos principais motivos pelos quais os computadores modernos em sua grande maioria operam no sistema binário e não decimal.

Mais interessante ainda é que Leibniz chegou a descrever um mecanismo de computação muito semelhante aos computadores modernos. Ele pensou e descreveu o que viria a ser o sistema de cartões perfurados para entrada e saída de dados.

## 2.7 Contador Mecânico de Leibniz – 1694

Conta a história que Leibniz, ao conhecer um pedômetro (um aparelho capaz de contar o número de passadas dadas por um ser humano), teve a ideia de construir um calculador mecânico em 1672. Tempos depois, ele acabou entrando em contato com os escritos de Pascal e conheceu a Pascalina. Leibniz ficou praticamente obcecado pela ideia de construir uma máquina capaz de fazer operações aritméticas e decidiu-se a aperfeiçoar a Pascalina de modo a conseguir realizar multiplicações e divisões de forma totalmente automatizada.

Inicialmente, Leibniz tentou projetar um mecanismo semelhante ao de Pascal, mas logo se deu conta de que para realizar multiplicações e divisões automaticamente ele precisaria criar um mecanismo totalmente novo, o que ele finalmente conseguiu ao criar o cilindro escalonado (*stepped-drum*).

A máquina de Leibniz (*Stepped Reckoner*) esteve em desenvolvimento durante cerca de 40 anos; vários modelos foram construídos e seu projeto final acabou sendo a base para as calculadoras mecânicas em uso até a década de 1970, quando foram finalmente substituídas pelos modelos eletrônicos. A [Figura](#)

[2.12](#) apresenta um desenho mostrando a máquina em sua visão frontal e superior.

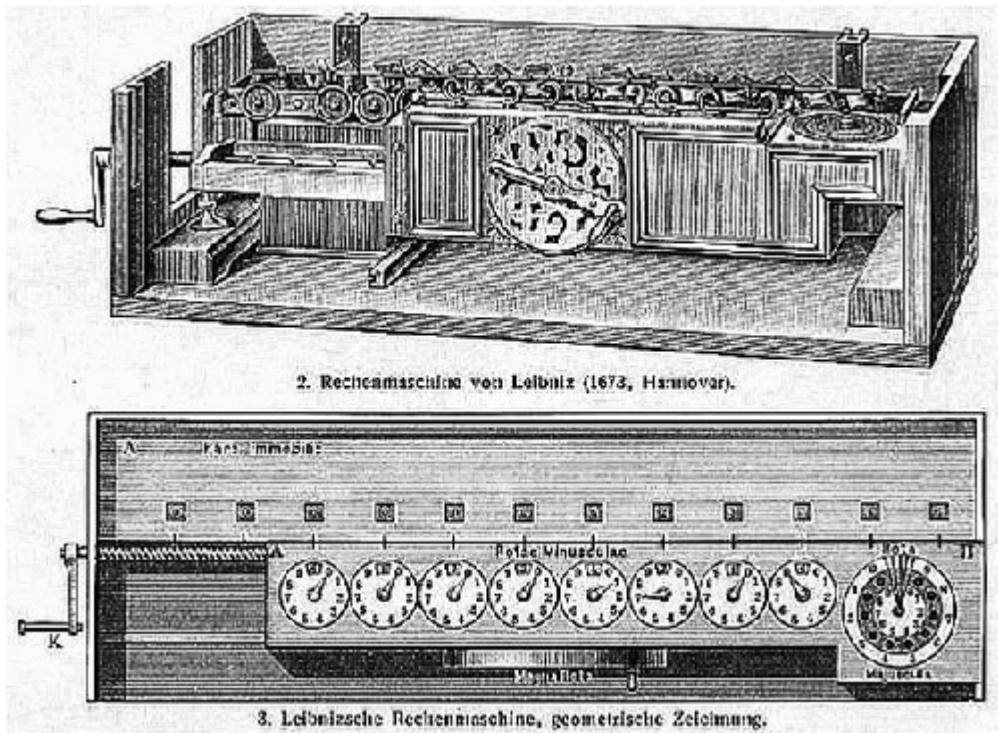


Figura 2.12: Desenho da calculadora de Leibniz.<sup>10</sup>

O primeiro protótipo de apenas dois dígitos foi construído em 1672 e foi apresentado na Academia Francesa de Ciências. Somente duas máquinas feitas na época de Leibniz e várias réplicas posteriores sobreviveram até os nossos dias. Uma delas ficou esquecida em um sótão na universidade de Göttingen até que em 1879 a necessidade de um conserto no telhado fez com que fosse encontrada.

A [Figura 2.13](#) mostra o esquema básico de funcionamento do cilindro escalonado de Leibniz. O cilindro em questão, exibido na parte de baixo da figura, tem dentes com diferentes extensões. A ideia é que a cada inserção de números, uma manivela faça o cilindro dar uma volta completa. Mas a quantidade de movimento colocada no acumulador (representado pela engrenagem na parte de cima na figura), dependerá da posição em que se encontrar o

seletor que move essa engrenagem para a direita ou para a esquerda. Quanto mais à esquerda a engrenagem estiver, menor o número de dentes que vai mover-la a cada giro completo do cilindro inferior.

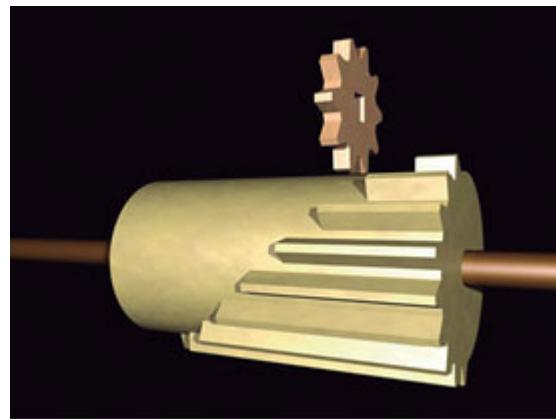


Figura 2.13: Esquema do cilindro escalonado de Leibniz.<sup>11</sup>

Um mecanismo como esse precisaria ser construído para cada dígito dos números que se pretende operar, bem como um mecanismo de *carry* (vai um) para os diferentes acumuladores. Desta forma, para efetuar somas e subtrações, basta posicionar os indicadores dos diferentes dígitos nos números desejados e girar uma volta completa na manivela. O sentido horário realiza somas e o sentido anti-horário, subtrações.

Para efetuar uma multiplicação, por exemplo,  $825 \times 3$ , bastaria posicionar os indicadores dos três últimos dígitos nas posições 8, 2 e 5 e girar a manivela 3 vezes, o que realizaria na prática a operação  $825 + 825 + 825 = 3 \times 825$ . Leibniz acrescentaria um visor no qual se poderia verificar quantas vezes a alavanca foi girada.

Contudo, como fazer multiplicações de números grandes como  $1873 \times 249$ ? Para evitar ter que girar a alavanca 249 vezes, Leibniz criou um mecanismo ainda mais engenhoso com o qual todo o conjunto seria movimentado da seguinte forma: primeiramente, coloca-se o conjunto na posição das centenas, o que faz com que

cada giro da alavanca some na casa das centenas, equivalendo a uma multiplicação por 100, depois nas dezenas e finalmente nas unidades. Assim, para multiplicar por 249, posiciona-se o conjunto na casa das centenas e se realizam dois giros, depois nas dezenas quatro giros e finalmente nas unidades, nove giros. Isso equivale a operação  $1873 \times (2 \times 100) + 1873 \times (4 \times 10) + 1873 \times (9 \times 1) = 1873 \times 249$ . O resultado 466.377 seria mostrado pela máquina.

Acredita-se que a máquina de Leibniz não tenha sido um sucesso completo em sua época primeiramente devido ao seu alto custo, mas também pelo fato de que ele aparentemente não resolveu completamente o problema do mecanismo de *carry* – em alguns casos, a intervenção manual do operador poderia ser necessária para realizar o *carry* com sucesso.

## 2.8 Computadores Humanos – 1758

Antes que os computadores fossem eletromecânicos ou puramente eletrônicos, eles eram humanos. A própria palavra “computador” até meados do século XX significava o nome de uma profissão, e não de uma máquina. Especialmente durante as duas guerras mundiais podia-se encontrar nos classificados dos jornais, nos países envolvidos, anúncios procurando por “computadores” para trabalhar nos cálculos necessários para diversas atividades relacionadas à guerra.

Pode-se dizer que a profissão em si nasceu a partir da necessidade criada pelo trabalho de Edmond Halley (Inglaterra, 1656-1742). Halley tentou calcular a órbita do cometa que depois acabou recebendo seu próprio nome, mas foi incapaz de conseguir algo mais que uma aproximação grosseira e deixou a tarefa para as gerações seguintes. A órbita era influenciada principalmente pela gravidade do Sol, mas também por Júpiter e Saturno e não havia uma equação conhecida que permitisse seu cálculo.

Em 1758, Alexis-Claude Clairaut (França, 1713-1765) criou um novo modelo matemático para prever a órbita do cometa, mas este

modelo só teria resolução numérica, ou seja, para calcular sua posição, uma série extensiva de cálculos teria que ser feita (e não a aplicação de uma simples fórmula). No mesmo ano, Clairaut contratou dois amigos para ajudá-lo nos cálculos. Eles se debruçaram sobre os números durante cinco meses e são considerados a primeira equipe de “computadores” do mundo. Completaram o trabalho mas, infelizmente, erraram por 31 dias a previsão de quando o Halley estaria em seu ponto mais próximo ao Sol. Isso fez com que alguns cientistas desmerecessem o trabalho dos computadores, mas logo depois outros grupos começaram a se formar.

Para que uma equipe de computadores pudesse funcionar, ela teria que ser adequadamente organizada. Assim, a divisão de trabalho inicialmente alocada a cada membro da equipe era peça fundamental. Cada pessoa não precisaria ser um matemático de mão cheia: na maioria dos casos bastaria saber as quatro operações aritméticas, mas deveriam ser pessoas bem disciplinadas e focadas, já que o trabalho envolvia executar muitos procedimentos puramente mecânicos e repetitivos com números. Ou seja, na prática as pessoas estariam seguindo mecanicamente os algoritmos estabelecidos para cada caso.

Durante o século XIX, verdadeiras fábricas com dezenas ou centenas de computadores humanos surgiram ao redor do mundo para calcular os mais diferentes tipos de tabelas de números. Não apenas a área da astronomia, mas também a engenharia, a construção, as finanças, os bancos e seguros dependiam de tabelas de números que precisavam ser previamente calculadas. Os primeiros trabalhos com computadores humanos foram considerados de baixo retorno, pois muito tempo e dinheiro se gastava com os cálculos e pouco se obtinha de retorno, mas após a Revolução Industrial, bens e dinheiro passaram a depender fortemente dessas inúmeras tabelas. Possivelmente por isso Charles Babbage, no início do século XIX, tenha obtido recursos

para a época considerados astronômicos em investimentos para mecanizar o processo de cálculo dessas tabelas.

A maioria dos computadores humanos eram mulheres. Entre outros motivos para isso, cita-se o fato de que, primeiramente, essa era uma das poucas formas de uma mulher se envolver efetivamente em ciência na época, e além disso o salário pago a elas era bem menor do que o pago aos homens (algo como 35 centavos de dólar para mulheres contra 50 centavos por hora para homens).

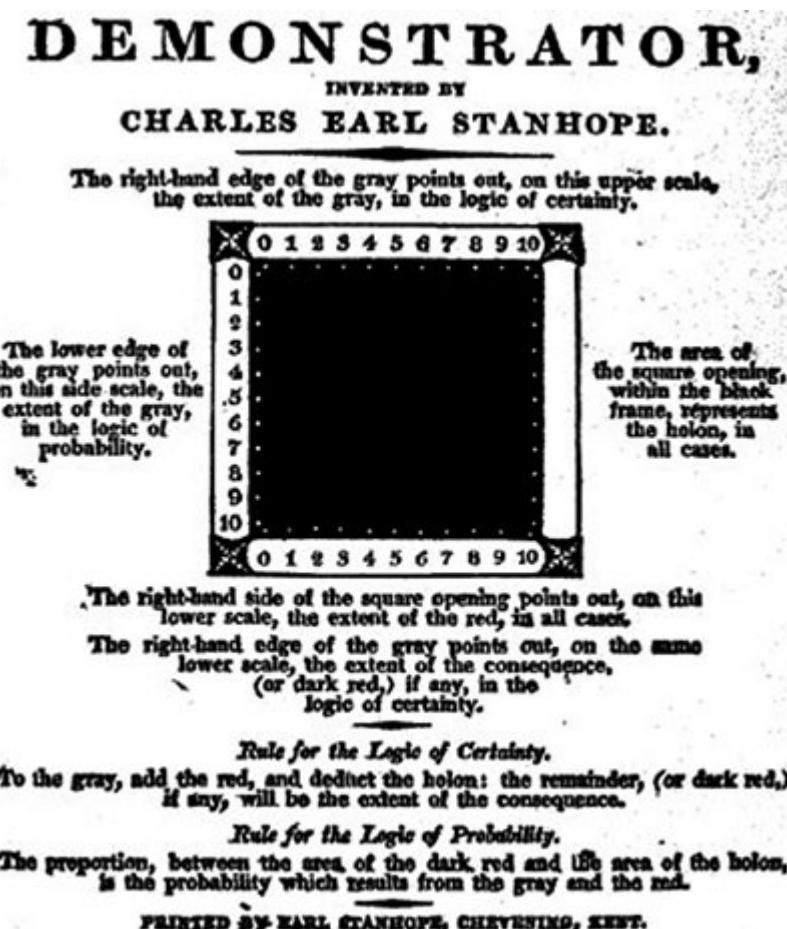
A era dos computadores humanos só terminou com o advento dos primeiros computadores eletrônicos, mais precisamente a partir do final dos anos 1940, quando eles começaram a entrar em operação em empresas e no âmbito governamental. A previsão da órbita do cometa de Halley para 1986 já foi calculada por um desses primeiros computadores eletrônicos, o UNIVAC.

## **2.9 Calculadora e Demonstrador de Charles Stanhope – 1775**

Durante os séculos XVIII e XIX, várias pessoas projetaram ou construíram diferentes tipos de máquinas de calcular. A maioria delas inspirou-se nos mecanismos de Leibniz e Pascal sem apresentar grandes inovações, isso quando não arquitetavam mecanismos menos sofisticados que os originais.

Uma exceção a essa regra foi a máquina de Charles Stanhope (Inglaterra, 1753-1816) que desenvolveu em 1775 uma forma inovadora de realizar a execução do “vai um”. De fato, ele idealizou e construiu um mecanismo que fazia o *carry* em duas fases: uma fase de preparação, quando o 9 passa para a posição 0, e uma fase de execução, quando o *carry* é efetivamente realizado na engrenagem seguinte. Isso evitava o problema da acumulação de esforço nas engrenagens, já abordado por Pascal e Leibniz. O mecanismo de Stanhope passou a ser utilizado nas máquinas calculadoras desde então por mais de 100 anos.

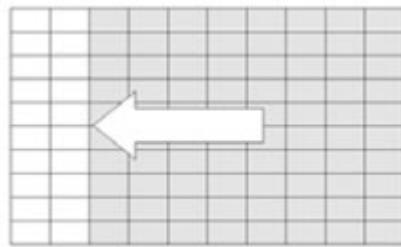
Além desta contribuição, em 1805 ele também construiu um mecanismo chamado “demonstrador”, que era capaz de mecanicamente resolver silogismos numéricos simples ou combinar probabilidades ([Figura 2.14](#)). Para realizar as demonstrações, ele movia alavancas nas laterais do dispositivo, que exibiam resultados a serem interpretados na tela central.



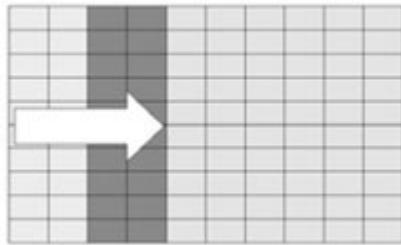
[Figura 2.14: O demonstrador de Charles Stanhope.](#)<sup>12</sup>

Por exemplo, se 8 em 10 As são B (80%) e 4 em 10 As são C (40%), então quantos B são C, no mínimo? Stanhope moveria uma alavanca para a esquerda oito unidades ([Figura 2.15](#)); em seguida ele moveria outra alavanca para a direita quatro unidades ([Figura 2.16](#)) e o resultado seria imediatamente visível na tela central,

indicado pela quantidade de colunas cobertas pela intersecção da área coberta pela primeira e pela segunda alavancas, ou seja, duas colunas; portanto a conclusão é que pelo menos 20% dos Bs são C.

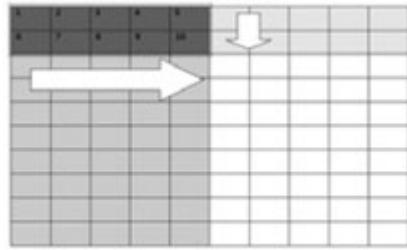


**Figura 2.15:** Inserindo a premissa maior “8 em 10 As são B”.



**Figura 2.16:** Inserindo a premissa menor “4 em 10 As são C” e concluindo que 2 em 10 Bs são C.

A máquina também era capaz de combinar (multiplicar) probabilidades. Para combinar duas probabilidades como  $1/2$  e  $1/5$ , ele moveria uma alavanca para baixo duas unidades e outra alavanca para a esquerda cinco unidades, o resultado seria lido na área da intersecção das duas áreas, no caso,  $1/10$ , conforme mostrado na [Figura 2.17](#).



**Figura 2.17: Combinando duas probabilidades:  $1/2$  e  $1/5 = 1/10$ .**

As demonstrações obtidas por este mecanismo eram bem simples e poderiam ser facilmente calculadas de cabeça. Porém, seu verdadeiro valor esteve em mostrar que raciocínios lógicos poderiam ser realizados de forma mecânica, coisa que os computadores modernos fazem muito bem.

## **2.10 Calculadora e Máquina Diferencial de Johann H. Müller – 1783**

Outro engenheiro que fez a diferença (literalmente, pois ele inventou uma máquina diferencial) foi Johann H. Müller (Alemanha, 1746-1830). Inicialmente, ele idealizou e construiu uma pequena, mas muito interessante, calculadora em formato circular (design ainda utilizado durante o século XX pelas calculadoras Curta). Um detalhe desse mecanismo é mostrado na [Figura 2.18](#).



**Figura 2.18: Detalhe da calculadora circular de Johann Helfrich Müller.<sup>13</sup>**

Por volta de 1780, seu patrão lhe pediu que conferisse e recalculasse algumas tabelas numéricas. Como outros antes e depois dele, Müller sentiu que seria mais produtivo construir uma máquina que calculasse tais números de forma automática. Após ler um artigo de Phillip M. Hahn (Alemanha, 1739-1790), que também tinha criado uma calculadora circular baseada nas ideias de Leibniz, ele resolveu desenvolver seu próprio projeto e tentar melhorar alguns aspectos do mecanismo elaborado por Hahn, especialmente no que se relacionava ao sempre problemático mecanismo de *carry*. A máquina de Hahn era capaz de realizar as quatro operações com números de até 14 dígitos e podia trabalhar tanto com o sistema decimal quanto com outros sistemas de numeração, o que na época era uma necessidade importante já que alguns sistemas monetários, para não mencionar o sistema de contagem de tempo usado até hoje (de base 60), não eram decimais.

Em relação à máquina de Hahn, as principais melhorias introduzidas por Müller foram:

- Na máquina de Hahn, os números eram inseridos pela movimentação de um bastão para a frente e para trás, o que podia ser difícil, pois exigia grande precisão; já na máquina de Müller, os números eram introduzidos pelo girar de um disco.
- As engrenagens da máquina de Müller podiam ser facilmente substituídas por engrenagens com números diferenciados de dentes, o que possibilitava que o mecanismo operasse em diferentes sistemas numéricos.
- A máquina de Müller adicionou uma campainha que tocava no caso de *overflow* de operação ou quando uma subtração produzia um número negativo (tentar subtrair um número maior de um menor).

Porém, mais importante do que esse trabalho, Müller teve o mérito de ter pela primeira vez concebido uma máquina diferencial, capaz de calcular automaticamente séries de polinômios usando o método das diferenças, que veremos mais adiante. Müller concebeu a máquina e publicou seu projeto em um livro em 1786, mas não foi capaz de obter financiamento para construí-la. Apenas em 1821 Charles Babbage conseguiu os recursos necessários para iniciar a construção de uma máquina do tipo. Assim, talvez por este motivo, o mérito pela invenção da máquina diferencial seja usualmente atribuído a Babbage, relegando a Müller uma posição relativamente esquecida na história. Porém, conforme veremos, mesmo após gastar uma fortuna, Babbage foi incapaz de terminar a máquina diferencial.

## 2.11 O Tear de Jacquard – 1801

Uma invenção que teve grande impacto no desenvolvimento da computação no século XX foi o cartão perfurado. Este foi o primeiro meio desenvolvido para dar entrada de dados em grande

quantidade e também para introduzir instruções nos computadores até meados dos anos 1980.

Mas o cartão perfurado tem uma história muito mais antiga do que a dos computadores eletrônicos. Um dos grandes passos nessa história foi dado por Joseph Marie Charles *dit* Jacquard (França, 1752-1834), que aperfeiçou projetos anteriores de teares automáticos com um sistema que permitia a programação automática dessas máquinas com o uso de cartões perfurados. Um desses teares é mostrado na [Figura 2.19](#).

Basicamente o tear serve para tramar fios para criar tecidos. A cada movimento do tear uma nova “linha” do tecido é produzida. Para criar desenhos pode-se selecionar quais fios serão tramados em cada linha. Uma sequência de decisões desse tipo leva à construção de desenhos ou padrões no próprio tecido. Obviamente que para um operador humano, o trabalho de decidir a cada instante quais fios utilizar ou não era altamente repetitivo e pouco produtivo.

Jacquard então aperfeiçou por volta de 1801 o sistema que permitia ao tear automaticamente tomar essa decisão em função de cartões perfurados que eram passados pela máquina a cada instante. A sequência de furos nos vários cartões previamente programados iria produzir o desenho ou padrão desejado.



**Figura 2.19: Tear baseado em cartões perfurados de Jacquard.<sup>14</sup>**

Jacquard com certeza conhecia os projetos de seus antecessores, Basile Bouchon, Jean Falcon e especialmente Jacques Vaucanson (França, 1709-1782). Em relação ao projeto de Vaucanson, Jacquard eliminou a fita de papel que este utilizava e retornou ao modelo de cartões de Falcon. Ele procurou também retirar os caros cilindros de metal de Vaucanson, substituindo-os por um mecanismo de cabeçote adaptável a uma série de teares.

No mecanismo de Jacquard, cada posição do cartão perfurado correspondia a um gancho que podia ser elevado ou ficar parado dependendo de a posição conter um furo ou não. Cada gancho podia ser conectado a mais de uma linha, permitindo que o padrão repetisse várias vezes ao mesmo tempo. Assim, um tear com um cabeçote de 500 ganchos poderia ter quatro linhas ligadas a cada gancho, resultando em um tecido de 2.000 fios de largura.

Para demonstrar o potencial de sua máquina, Jacquard produziu uma imagem de si próprio que foi tecida com fios brancos e pretos a partir de um programa formado com 10 mil cartões. A imagem resultante é mostrada na [Figura 2.20](#). Consta que Charles Babbage, idealizador do primeiro computador universal, tinha uma destas imagens em sua sala para servir de inspiração, visto que ele mesmo esperava que seu mecanismo fosse alimentado por cartões perfurados e, por vezes, o descrevia como uma máquina de tecer números.

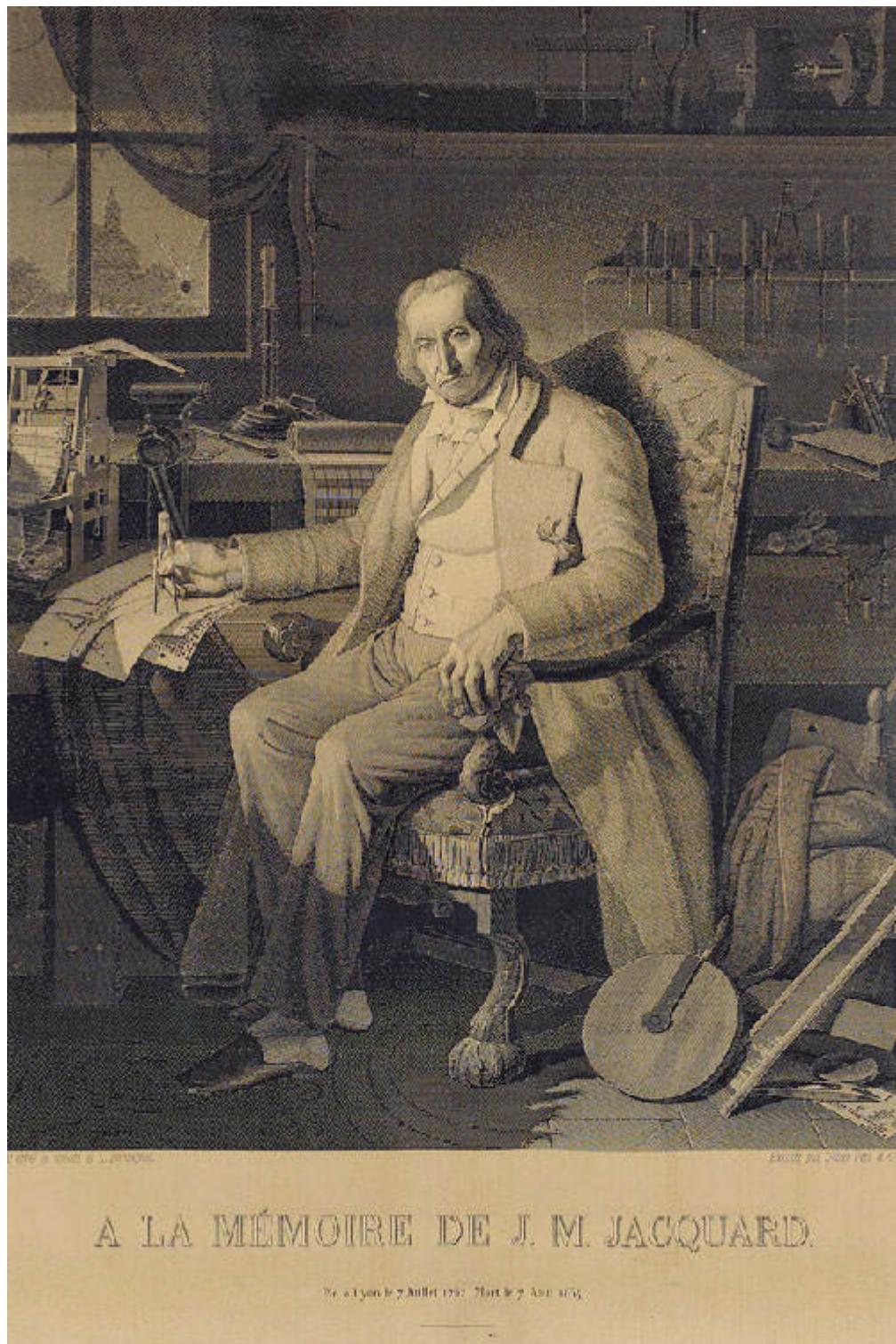


Figura 2.20: Imagem de Jacquard tecida a partir de 10 mil cartões perfurados.<sup>15</sup>

A invenção de Jacquard não teve um início muito satisfatório, não porque não funcionasse, mas pelo medo dos trabalhadores da indústria de tecelagem de perderem seus empregos. Em 1801, após uma apresentação exitosa em Paris, Jacquard levou seu tear para ser mostrado em Lyon. Lá ele foi atacado por uma multidão e o tear foi destruído. Assim, Jacquard por pouco não foi o primeiro mártir da computação. O problema do desemprego causado pela automatização continuou a ser um assunto em franca discussão durante os séculos XIX e XX.

Em 1805, o imperador Napoleão concedeu a Jacquard uma patente pela sua invenção. Em função disso ele recebeu uma grande pensão, bem como uma participação nos lucros da venda de teares Jacquard nos seis anos seguintes. Isso o deixou rico, pois consta que cerca de 11 mil de suas máquinas foram vendidas até 1811.

Nessa época, os conjuntos de cartões perfurados que poderiam ser usados para reproduzir os padrões pré-programados inúmeras vezes tinham um valor semelhante ao dos softwares nos dias de hoje. Surgiu então a pirataria, pois consta que algumas tecelagens roubavam conjuntos de cartões de outras para poder usar os mesmos padrões sem ter o trabalho de programá-los.

Teares semelhantes aos de Jacquard ainda são usados nos dias de hoje, mas sua operação é totalmente computadorizada, desde a captura da imagem até a execução do tecido.

## 2.12 Arithmometer – 1820

Durante o século XIX, especialmente na sua segunda metade e em plena Revolução Industrial, as calculadoras mecânicas tornaram-se mais do que simples curiosidade a ser colecionada por reis e nobres, e adquiriram status de ferramenta de trabalho produzida em escala industrial, suficientemente robusta para ser usada nas atividades diárias de empresas diversas.

A partir de 1820, Charles Xavier Thomas de Colmar (França, 1785-1870) desenhou e construiu uma calculadora que viria a ser o ícone

de uma era: a Arithmometer. De fato, ela chegou a ser produzida até 1915. Sua produção só parou devido à Primeira Guerra Mundial. Após a guerra, ela não foi retomada pela falta de pessoas habilitadas para produzi-la. A [Figura 2.21](#) mostra uma Arithmometer de 1875.

A Arithmometer baseava-se nos mesmos princípios do contador mecânico de Leibniz. Ela era capaz de fazer as quatro operações aritméticas usando os mesmos expedientes criados pelo cientista alemão. De fato, o grande mérito de Thomas foi o de investir em seu design sólido e robusto, permitindo maior confiabilidade e simplicidade no uso.

Tamanho foi o sucesso desta calculadora e tantas máquinas foram vendidas que ela chegou a ter vários clones ([Figura 2.22](#)) produzidos e comercializados em todo o mundo – para completar, seu nome passou a ser, na época, sinônimo de “calculadora”. Ela também teve manuais de usuário durante sua existência.



Figura 2.21: Arithmometer.<sup>16</sup>

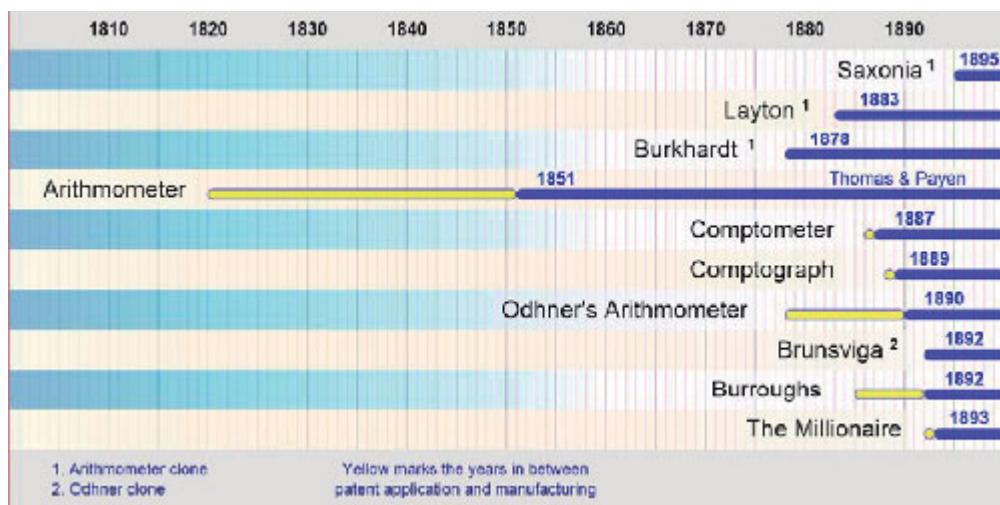


Figura 2.22: Linha do tempo da Arithmometer e seus concorrentes.<sup>17</sup>

O primeiro protótipo foi apresentado publicamente em 1820. Thomas recebeu uma patente por cinco anos, mas não iniciou imediatamente a produção e venda. Antes disso, ele aperfeiçoou o mecanismo. O protótipo de 1821, que ainda pode ser visto no museu Smithsonian tinha apenas três dígitos e, em vez de uma alavanca, possuía uma fita que devia ser puxada para girar os cilindros de Leibniz, claramente visíveis no lado esquerdo inferior da [Figura 2.23](#). Este protótipo possuía uma característica que depois foi abandonada: um dígito multiplicador que se fosse colocado, por exemplo, na posição 8, fazia os cilindros girarem oito voltas quando a fita fosse puxada, evitando assim que o usuário tivesse que fazer a operação oito vezes, como no caso da contadora mecânica de Leibniz ou da Pascalina.

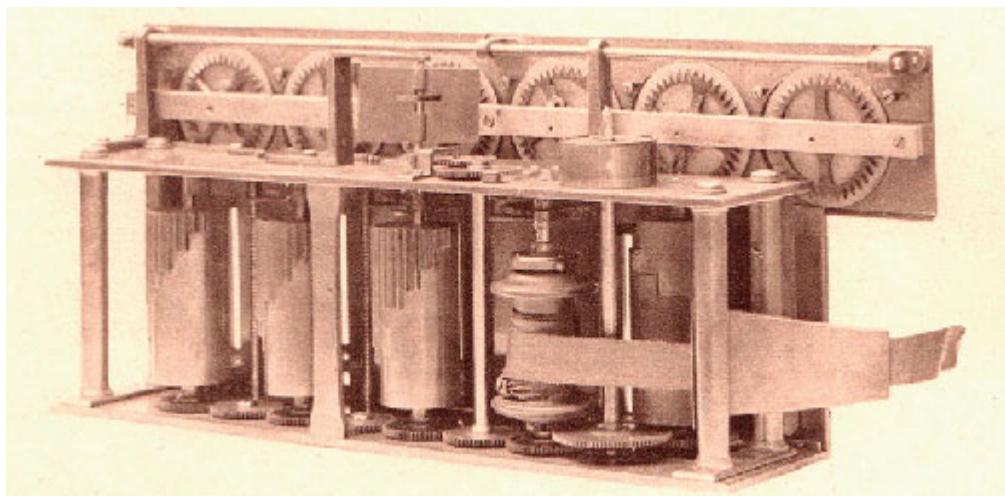


Figura 2.23: Visão interna da Arithmometer de 1821.<sup>18</sup>

Até 1850 pouco se sabe se Thomas trabalhou na concepção de sua máquina ou se simplesmente cuidou de outros projetos. Mas a partir daí novos protótipos começaram a surgir: máquinas maiores, porém mais simples e mais robustas.

A Arithmometer foi construída com vários tamanhos – os mais comuns eram de 10, 12, 16 e 20 dígitos. Entretanto, uma calculadora especial com 30 dígitos foi construída para a exposição universal de Paris. Ela era, portanto, capaz de cálculos de até um nonilhão menos um, operando de zero a 999.999.999.999.999.999.999.999.999. Essa máquina hoje pertence à coleção de calculadoras da IBM. Júlio Verne (França, 1828-1905) viu a máquina em Paris e ficou tão impressionado que mencionou em seu livro *Paris no século XX*, escrito em 1863, que haveria calculadoras do tamanho de pianos por volta de 1960, capazes de dar respostas instantâneas a quem pudesse operar seus teclados (uma visão quase exata dos computadores eletrônicos de 1960, obtida 100 anos antes).

## 2.13 Até Aqui...

Como vimos neste capítulo, os séculos de XVII a XIX viram a necessidade de realização de cálculos mais rápidos, pois diversos problemas, inicialmente na astronomia e mais tarde em outras áreas, exigiam a realização de cálculos repetitivos à exaustão.

Nesta época surgiu o conceito de “computador”, que era uma profissão, e o conceito de calculadora mecânica. Pascal e Leibniz foram as duas maiores influências na construção destas máquinas; o primeiro por divulgar amplamente um mecanismo capaz de somar e subtrair, e o segundo por criar um dispositivo para realizar multiplicações e divisões. Sua contribuição com a aritmética binária também foi importante, mas só seria relevante em meados do século XX com a chegada dos computadores eletrônicos.

A partir da metade do século XIX, calculadoras mecânicas como a Arithmometer e seus clones se tornaram acessíveis ao grande público e uma nova revolução estava prestes a se iniciar, como veremos no capítulo seguinte.

---

<sup>1</sup> “Rechenmaschine wilhelm schickard” by *The original uploader was Wernerroth at German Wikipedia - Transferred from de.wikipedia to Commons. Licensed under Public Domain via Wikimedia Commons* – Disponível em: [https://commons.wikimedia.org/wiki/File:Rechenmaschine\\_wilhelm\\_schickard.png#/media/File:Rechenmaschine\\_wilhelm\\_schickard.png](https://commons.wikimedia.org/wiki/File:Rechenmaschine_wilhelm_schickard.png#/media/File:Rechenmaschine_wilhelm_schickard.png)

<sup>2</sup> “Schickard” por teclasorg. Disponível em: <http://www.flickr.com/photos/teclasorg/2834616624/>. Licenciado sob CC BY 2.0 via Wikimedia Commons

<https://commons.wikimedia.org/wiki/File:Schickard.jpg#/media/File:Schickard.jpg>

<sup>3</sup> Tradução do autor a partir do inglês (original em latim).

<sup>4</sup> “Arts et Metiers Pascaline dsc03869” by © 2005 David Monniaux /. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Arts\\_et\\_Metiers\\_Pascaline\\_dsc03869.jpg#/media/File:Arts\\_et\\_Metiers\\_Pascaline\\_dsc03869.jpg](https://commons.wikimedia.org/wiki/File:Arts_et_Metiers_Pascaline_dsc03869.jpg#/media/File:Arts_et_Metiers_Pascaline_dsc03869.jpg)

<sup>5</sup> Tradução: Raciocinar significa computar.

<sup>6</sup> “Leviathan by Thomas Hobbes” by Unknown. Disponível em: <http://www.loc.gov/exhibits/world/world-object.html>; <http://www.securityfocus.com/images/columnists/leviathan-large.jpg>. Licensed under Public Domain via Wikimedia Commons. [https://commons.wikimedia.org/wiki/File:Leviathan\\_by\\_Thomas\\_Hobbes.jpg#/media/File:Leviathan\\_by\\_Thomas\\_Hobbes.jpg](https://commons.wikimedia.org/wiki/File:Leviathan_by_Thomas_Hobbes.jpg#/media/File:Leviathan_by_Thomas_Hobbes.jpg)

<sup>7</sup> Domínio Público. Cortesia: Dalakov, G. *History of Computers, Hardware, Software, Internet...* Disponível em: <http://history-computer.com/Dreamers/Kircher.html>

<sup>8</sup> Domínio Público. Cortesia: Dalakov, G. *History of Computers, Hardware, Software, Internet...* Disponível em: <http://history-computer.com/MechanicalCalculators/Pioneers/Morland.html>.

<sup>9</sup> By Samuel Morland (inventor). *Engraver unknown. - This image is available from the United States Library of Congress's Prints and Photographs division under the digital ID cph.3c10470. Public Domain.* Disponível em: <https://commons.wikimedia.org/w/index.php?curid=6246399>

<sup>10</sup> “Leibniz Stepped Reckoner drawing” by Hermann Julius Meyer - Downloaded 2008-1-16 from Leibniz’s Calculator, *Rechnerlexikon* which states it is from Hermann Julius Meyer, ed. (1893-1897) Meyers Konversationslexicon, 5th Ed., Bibliographischen Institut, Leipzig, Germany. Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Leibniz\\_Stepped\\_Reckoner\\_drawing.png#/media/File:Leibniz\\_Stepped\\_Reckoner\\_drawing.png](https://commons.wikimedia.org/wiki/File:Leibniz_Stepped_Reckoner_drawing.png#/media/File:Leibniz_Stepped_Reckoner_drawing.png)

<sup>11</sup> By Barbara - Own work, GFDL. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=11203355>

<sup>12</sup> Domínio Público. Cortesia: Dalakov, G. *History of Computers, Hardware, Software, Internet...* Disponível em: <http://history-computer.com/ModernComputer/thinkers/Stanhope.html>.

<sup>13</sup> “Detail der Rechenmaschine von Johann Helfrich Müller” by Nick Stahlkocher - *Stahlkochers collection. Licenced under CC BY-SA 3.0 via Wikimedia Commons.* Disponível em:

[https://commons.wikimedia.org/wiki/File:Detail\\_der\\_Rechenmaschine\\_von\\_Johann\\_Helfrich\\_M%C3%BCller.jpg#/media/File:Detail\\_der\\_Rechenmaschine\\_von\\_Johann\\_Helfrich\\_M%C3%BCller.jpg](https://commons.wikimedia.org/wiki/File:Detail_der_Rechenmaschine_von_Johann_Helfrich_M%C3%BCller.jpg#/media/File:Detail_der_Rechenmaschine_von_Johann_Helfrich_M%C3%BCller.jpg)

<sup>14</sup> “Jacquard loom, 6 of 6” by GeorgeOnline - Own work. *Licenced under CC BY-SA 3.0 via Wikimedia Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Jacquard\\_loom,\\_6\\_of\\_6.jpg#/media/File:Jacquard\\_loom,\\_6\\_of\\_6.jpg](https://commons.wikimedia.org/wiki/File:Jacquard_loom,_6_of_6.jpg#/media/File:Jacquard_loom,_6_of_6.jpg)

<sup>15</sup> “A la mémoire de J.M. Jacquard” by Michel Marie Carquillat (*tisseur*) *d’après Claude Bonnefond - Bonhams. Licenced under Public Domain via Wikimedia Commons.* Disponível em:

[https://commons.wikimedia.org/wiki/File:A\\_la\\_mC3%A9moire\\_de\\_J.M.\\_Jacquard.jpg](https://commons.wikimedia.org/wiki/File:A_la_mC3%A9moire_de_J.M._Jacquard.jpg#/media/File:A_la_mC3%A9moire_de_J.M._Jacquard.jpg)

<sup>16</sup> “Thomas Arithmometer 1875” by Ezrdr - Own work. *Licenced under Public Domain via Wikimedia Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Thomas\\_Airthmometer\\_1975.png#/media/File:Thomas\\_Airthmometer\\_1975.png](https://commons.wikimedia.org/wiki/File:Thomas_Airthmometer_1975.png#/media/File:Thomas_Airthmometer_1975.png)

<sup>17</sup> “DesktopMechanicalCalculators inProduction intheXIXCentury” by Ezrdr - Own work. *Licensed under Public Domain via Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:DesktopMechanicalCalculators\\_inProduction\\_intheXIXCentury.svg#/media/File:DesktopMechanicalCalculators\\_inProduction\\_intheXIXCentury.svg](https://commons.wikimedia.org/wiki/File:DesktopMechanicalCalculators_inProduction_intheXIXCentury.svg#/media/File:DesktopMechanicalCalculators_inProduction_intheXIXCentury.svg)

<sup>18</sup> “Mechanism Arithmometer 1822” by La compagnie d’assurance “Le Soleil” - *French book celebrating the 100th anniversary of the insurance company “Le Soleil” 1829-1929, and its founder Thomas de Colmar. Licenced under Public Domain via Wikimedia Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Mechanism\\_Airthmometer\\_1822.png#/media/File:Mechanism\\_Airthmometer\\_1822.png](https://commons.wikimedia.org/wiki/File:Mechanism_Airthmometer_1822.png#/media/File:Mechanism_Airthmometer_1822.png)

# **PARTE III**

## **De Babbage a Hollerith**

O século XIX viu surgir o primeiro projeto de computador de propósito universal: a Máquina Analítica de Charles Babbage. Neste século também se consolidaram as calculadoras mecânicas baseadas em engrenagens, que passaram a ser bastante utilizadas em empresas e organizações. Pesquisas iniciais com dispositivos elétricos como os relês, permitiram que ao final desse século as primeiras máquinas somadoras ou contadoras com base eletromecânica fossem construídas por Hermann Hollerith. Não menos importante, durante o século XIX ocorreram avanços nas ciências da Lógica e Matemática, devido principalmente a Boole, Frege, dentre outros que lançaram os fundamentos teóricos para a Ciência da Computação a nascer no início do século XX com trabalhos como o de Turing.

### **3.1 Máquina Diferencial de Babbage – 1821**

Lembra das tabelas astronômicas de Kepler que motivaram a construção das calculadoras de Schickard e Pascal? Pois em 1821 o matemático e inventor Charles Babbage (Reino Unido, 1791-1871) estava revisando várias tabelas astronômicas calculadas à mão. Após encontrar vários erros, diz-se que ele exclamou: “Por Deus! Eu queria que esses cálculos tivessem sido executados a vapor!” Esse brado marcou o início de uma nova evolução no projeto das máquinas de calcular.

As máquinas de cálculo baseadas em engrenagens que efetuavam somas facilmente e lançavam o “vai um” para a esquerda sempre que necessário já tinham nessa época quase duzentos anos de existência. Mas a máquina que Babbage idealizou inicialmente seria muito mais complexa.

A Máquina Diferencial (*Difference Engine*) seria capaz de computar mecanicamente séries de polinômios utilizando o método das diferenças, que permite calcular o valor de uma sucessão de polinômios usando apenas somas – coisa que as máquinas da época já eram capazes de fazer. Por exemplo, o polinômio  $p(x) = x^2+1$  define uma série de valores para  $x$  valendo 1, 2, 3 etc. No caso:

- $p(1) = 1^2+1 = 2$
- $p(2) = 2^2+1 = 5$
- $p(3) = 3^2+1 = 10$
- etc.

O método das diferenças exige que se calcule manualmente ou que se saiba de antemão os  $n+1$  primeiros valores da série, onde  $n$  é o grau do polinômio. Assim, como  $x^2+1$  tem grau 2, precisamos calcular os três primeiros valores, como mostrado na [Tabela 3.1](#).

**Tabela 3.1: Três primeiros valores para o polinômio  $x^2+1$**

$x$	$p(x)$
1	2
2	5
3	10

A primeira diferença  $1d(x)$  é calculada como  $p(x+1) - p(x)$ . Assim, temos que calcular:

- $p(2) - p(1) = 5-2 = 3$
- $p(3) - p(2) = 10-5 = 5$ .

Esses valores são mostrados na coluna “ $1d$ ” da [Tabela 3.2](#).

**Tabela 3.2: Inclusão da primeira diferença**

$x$	$p(x)$	$1d(x)$
1	2	$3 = 5-2$
2	5	$5 = 10-5$
3	10	

Na sequência, o método exige que se calcule similarmente a segunda diferença, que é obtida subtraindo os valores das primeiras diferenças, ou seja,  $1d(2) - 1d(1) = 5-3 = 2$ . Esse valor é mostrado na coluna “ $2d$ ” da [Tabela 3.3](#).

**Tabela 3.3: Inclusão da segunda diferença**

$x$	$p(x)$	$1d(x)$	$2d(x)$
1	2	3	$2 = 5-3$
2	5	5	
3	10		

Sabe-se que se o polinômio é de grau  $n$ , então a  $n$ -ésima diferença será constante, ou seja, terá sempre o mesmo valor. Assim, a coluna  $2d(x)$  na tabela terá apenas o valor 2 em todas as suas linhas. Neste ponto, considera-se que a tabela está inicializada e pode-se proceder ao cálculo de todos os valores do polinômio para  $n > 3$ .

Fazendo primeiramente o cálculo para  $p(4)$ , o primeiro passo é colocar 2 (a constante) na posição  $2d(2)$ . Já a coluna das primeiras diferenças pode ter seus valores calculados como  $1d(x) = 1d(x-1) + 2d(x-1)$ . Assim,  $1d(3) = 1d(2) + 2d(2) = 5+2 = 7$ . O resultado é mostrado na [Tabela 3.4](#).

**Tabela 3.4: Calculando a  $1d(3)$  a partir de  $1d(2) + 2d(2)$**

$x$	$p(x)$	$1d(x)$	$2d(x)$
1	2	3	2
2	5	5	2
3	10	$7 = 2+5$	

De forma similar,  $p(4)$  é calculado como  $p(3) + 1d(3) = 10+7 = 17$ , o que corresponde corretamente ao valor de  $4^2+1$ . O resultado é

mostrado na [Tabela 3.5](#).

**Tabela 3.5: Calculando  $p(4)$  a partir de  $p(3) + 1d(3)$**

$x$	$p(x)$	$1d(x)$	$2d(x)$
1	2	3	2
2	5	5	2
3	10	7	
4	$17 = 10+7$		

Repetindo-se estes passos, pode-se calcular  $p(5)$ ,  $p(6)$  etc. até onde se desejar. O importante é que o método não exige multiplicações nem subtrações, coisa complicada para máquinas movidas a engrenagens, como vimos anteriormente. Assim, a tabulação de polinômios poderia ser feita caso fosse construída uma máquina com a capacidade de efetuar estas somas e armazenar os resultados intermediários das diferenças.

Para polinômios de ordem mais alta seriam necessárias mais diferenças, por exemplo, para um polinômio de grau 7, seria necessário ir até a sétima diferença. Mas o importante é que ela seria uma constante e a partir daí a série polinomial inteira poderia ser calculada.

Esta foi a máquina que Babbage idealizou. Ela teria oito acumuladores, sendo, portanto, capaz de calcular polinômios de ordem 7. Os acumuladores seriam semelhantes a uma calculadora de Leibniz em funcionalidade, permitindo que a cada giro da manivela um número fosse somado neles. No caso do exemplo acima, a máquina precisaria ter os acumuladores 6, 7 e 8 inicializados com os valores 10, 5 e 2, respectivamente. No primeiro instante, a máquina faria o acumulador 8 ser somado no acumulador 7, o qual passaria a valer  $2+5 = 7$ . Em seguida, o acumulador 7 seria somado ao acumulador 6, que passaria a valer  $7+10 = 17$ . Assim, estaria calculado o valor do polinômio para  $x = 4$ . Em seguida repete-se a sequência: soma-se o valor do acumulador 7 no

$6: 2+7 = 9$ , e o valor do acumulador 6 no 5:  $17+9 = 26$ , ou seja, o valor do polinômio para  $x = 5$ .

A máquina de Babbage, porém, não faria essas somas sequencialmente do primeiro para o último acumulador. Ele idealizou um método que faria as adições em paralelo em dois ciclos: inicialmente somaria os valores dos acumuladores pares nos ímpares à sua esquerda e depois dos acumuladores ímpares nos pares à sua esquerda. Desta forma, a velocidade da máquina não seria afetada pela ordem do polinômio. Com a técnica sequencial, um polinômio de ordem 7 gastaria sete ciclos da máquina, com a técnica paralela, apenas dois ciclos. Mesmo que a máquina fosse construída com 100 acumuladores, ainda assim, ela usaria apenas dois ciclos para gerar o próximo valor da sequência.

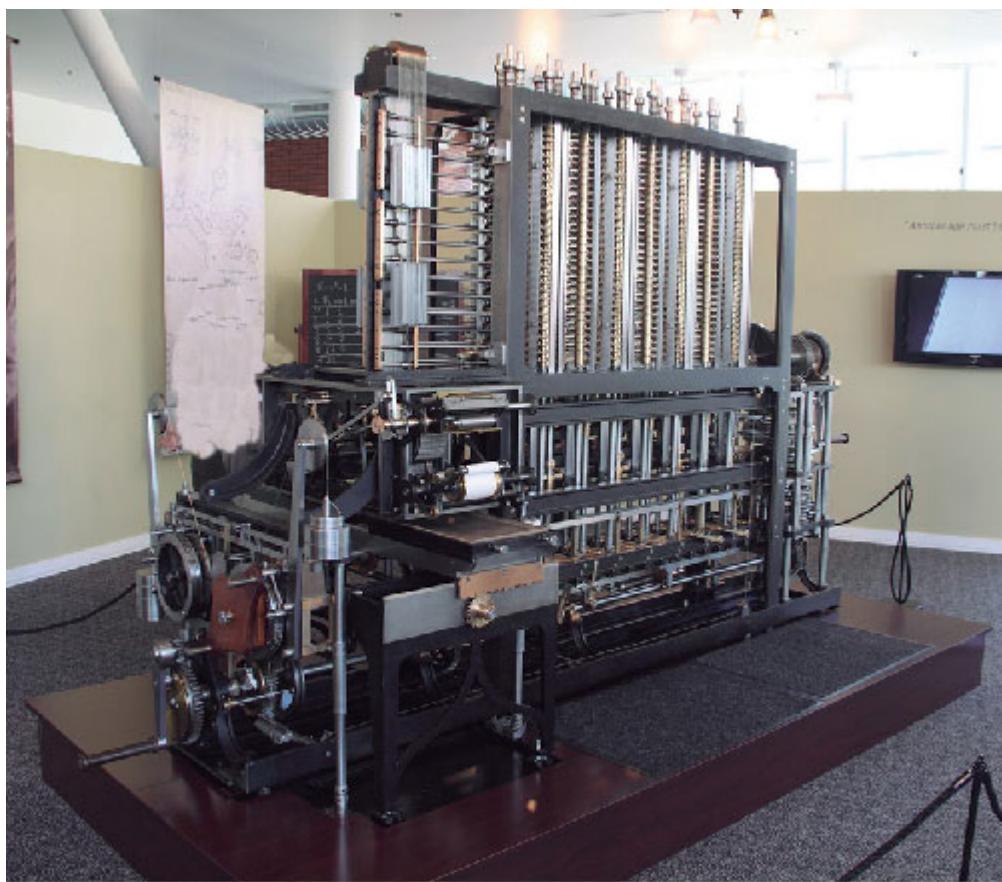
A rigor, a ideia de produzir tal instrumento era um pouco mais antiga. O primeiro a idealizá-lo foi Johann H. Müller, como já vimos. A ideia foi publicada em um livro em 1786, mas ele não obteve recursos para fazê-la se materializar.

Babbage, por sua vez, conseguiu recursos financeiros do governo inglês em 1823 para iniciar a construção de sua máquina. Porém, apenas partes dela (provas de conceito) foram construídas durante a vida do engenheiro. Entre outros fatores, consta que ele se desmotivou a construir a máquina diferencial porque, durante o projeto, teria vislumbrado uma máquina totalmente inovadora: a máquina analítica, que tornaria a máquina diferencial supérflua. Ele, no entanto, acabou não terminando nem uma nem outra. Em 1842, quando o governo inglês cancelou o projeto, o valor gasto de 17 mil libras equivalia ao preço de 22 locomotivas novas – uma verdadeira fortuna.

A partir da evolução das ideias de Babbage para a máquina analítica, ele pôde conceber uma nova versão da máquina diferencial, conhecida hoje como “Segunda Máquina Diferencial”. Ela usaria apenas 8 mil peças, um terço do necessário para o projeto original. Infelizmente, ele não conseguiu convencer mais

ninguém a financiar seu projeto e morreu frustrado, dizendo que a História o julgaria. De fato, cerca de 100 anos depois, o projeto foi retomado com a construção dos primeiros computadores da Era Contemporânea.

Em 2002, o Museu de Ciências de Londres produziu a primeira versão totalmente operacional da segunda máquina diferencial, após um projeto de 17 anos. Ela funciona perfeitamente como previsto pelo seu criador. Inclusive, também foi construída a impressora que Babbage idealizou para imprimir as sequencias de números. A máquina é mostrada na [Figura 3.1](#).



**Figura 3.1: Segunda máquina diferencial de Babbage construída entre 1985 e 2002 pelo Museu de Ciências de Londres.<sup>1</sup>**

Babbage inventou o “travamento”. Quando a máquina parava com alguma engrenagem em uma posição inválida (entre dois números), ela travava propositalmente para que o operador corrigisse o problema e evitasse a propagação do erro. Embora os engenheiros procurem minimizar esse tipo de situação, computadores modernos fazem isso até hoje, ainda que nem sempre de forma proposital.

### **3.2 Semyon Korsakov e o Registro de Informação em Cartões Perfurados – 1832**

Semyon Korsakov (Ucrânia, 1787-1853) foi um funcionário público que trabalhava no Departamento de Estatística do Ministério da Polícia do então Império Russo. Em 1832 ele publicou uma série de métodos e mecanismos para armazenamento de informação em cartões perfurados. Tais cartões já eram usados desde os tempos de Jacquard, mas apenas na indústria têxtil. Korsakov teve o mérito de propor, pela primeira vez, seu uso para armazenar e facilitar a busca de informação. Suas máquinas seriam capazes de localizar informações em cartões realizando aquilo que hoje fazemos com a pesquisa por palavras-chave na Internet.

Infelizmente, ele não foi capaz de convencer a Imperial Academia de Ciências de São Petersburgo de que seu trabalho era importante, e suas invenções ficaram esquecidas até serem redescobertas na década de 1960, graças a um trabalho de revisão histórica.

Korsakov se referia às suas invenções como “máquinas para comparação de ideias” ou “máquinas inteligentes”. De fato, ele é considerado um dos precursores da área de sistemas especialistas. Mais de 100 anos antes dos primeiros sistemas especialistas médicos, ele, que tinha formação de homeopata, propôs que suas máquinas poderiam ser usadas de modo que um médico com pouca experiência perfurasse os sintomas de um paciente em um cartão de papel e, a partir da busca, o aparelho localizaria e prescreveria automaticamente um tratamento, agindo como um sistema especialista.

Korsakov apresentou os projetos para cinco diferentes máquinas:

- Homeoscópio linear com partes móveis;
- Homeoscópio linear sem partes móveis;
- Homeoscópio plano;
- Ideoscópio; e
- Comparador simples.

A [Figura 3.2](#) mostra um desenho do homeoscópio linear sem partes móveis, feito pelo próprio Korsakov.

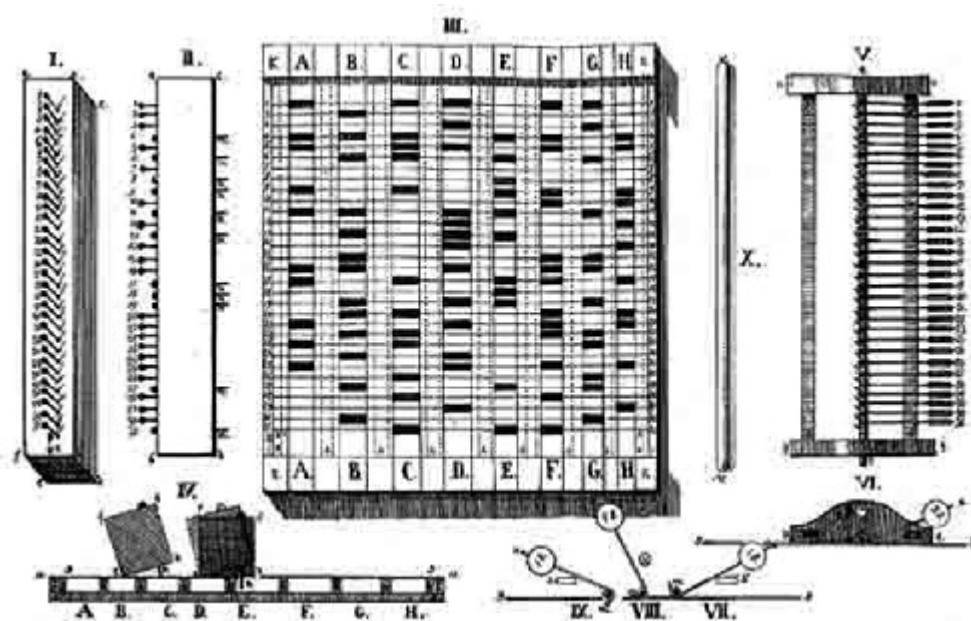


Figura 3.2: Homeoscópio linear sem partes móveis.<sup>2</sup>

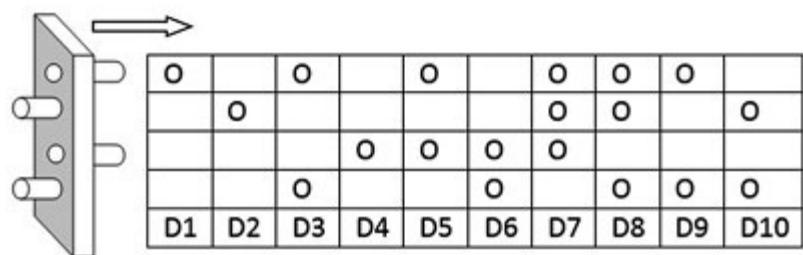
O funcionamento do homeoscópio linear sem partes móveis pode ser entendido da seguinte forma: imagine uma tabela na qual as linhas superiores são perfuradas cada qual com um padrão diferente e que a linha inferior contém a informação associada ao padrão. Essa ideia, demonstrada na [Figura 3.3](#) é muito semelhante à estrutura de dados baseada em chave-valor (ou dicionário) usada nas modernas linguagens de programação. Na figura, as quatro primeiras linhas representam a chave e a quinta linha, os dados

associados à cada chave. Cada uma das quatro primeiras linhas poderia representar, por exemplo, um dos sintomas observados no paciente, e a linha inferior poderia conter o tratamento. A coluna correspondente a um tratamento teria perfurados apenas os sintomas que eventualmente levariam à conclusão de que aquele seria o tratamento adequado.

O		O		O		O	O	O	
	O					O	O		O
			O	O	O	O			
		O			O		O	O	O
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10

**Figura 3.3:** Esquema representativo de uma tabela onde informação é acessada a partir de padrões perfurados.

O padrão de pesquisa então seria representado através de uma barra com pinos ([Figura 3.4](#)) que poderiam ser baixados, ou não, de forma que quando a barra corresse sobre a tabela, ela seria encaixada apenas na coluna na qual os furos correspondessem aos pinos abaixados. Assim, por exemplo, se os sintomas relativos à primeira e à terceira linhas fossem registrados como pinos abaixados na primeira e terceira posição da barra, ao correr sobre a tabela da esquerda para a direita, a barra iria parar apenas na coluna 5 e o tratamento aplicado seria D5.



**Figura 3.4: Esquema da barra de leitura de padrão.**

Para que esse mecanismo efetivamente funcionasse era necessário que as colunas fossem ordenadas de maneira que as com mais furos ficassem à direita daquelas com menos furos. Por exemplo, com os mesmos pinos 1 e 3 abaixados, caso a coluna 7 fosse encontrada antes da coluna 5, o tratamento indicado seria D7 e não D5, visto que a coluna 7 tem três furos, dois dos quais correspondem ao padrão de pesquisa.

O próprio Korsakov percebeu que seu mecanismo poderia ser aplicado em uma infinidade de áreas e não apenas no diagnóstico médico.

### **3.3 Calculadora com Teclado de Torchi – 1834**

Até a invenção do teclado, trabalhar com calculadoras ainda era muito difícil. Especialmente porque às vezes a precisão com que os discos ou barras eram movidos poderia afetar o funcionamento das máquinas, causando emperramentos ou resultados errados.

A invenção da primeira calculadora com teclado (*Macchina pei Conteggi*) é atribuída a Luigi Torchi (Itália, 1812-?). Assim, como no caso de Schickard, trata-se de uma invenção que foi redescoberta muitos anos depois e que provavelmente, apesar de ter sido pioneira, não chegou a influenciar decisivamente os desenvolvimentos posteriores, visto que suas ideias acabaram se concretizando de forma independente por outros cientistas algumas décadas mais tarde.

De fato, Torchi sequer era cientista; ele era um construtor de moinhos de apenas 22 anos quando apresentou ao Instituto Real-Imperial Lombardo de Ciências, Letras e Artes o protótipo de uma máquina calculadora com teclado, pelo qual ele recebeu a medalha de ouro em 1834.

Existem apenas duas descrições conhecidas da máquina de Torchi: a descrição do prêmio recebido em 1834 escrita à mão pela banca avaliadora e um artigo anônimo publicado na revista *La Fama* em

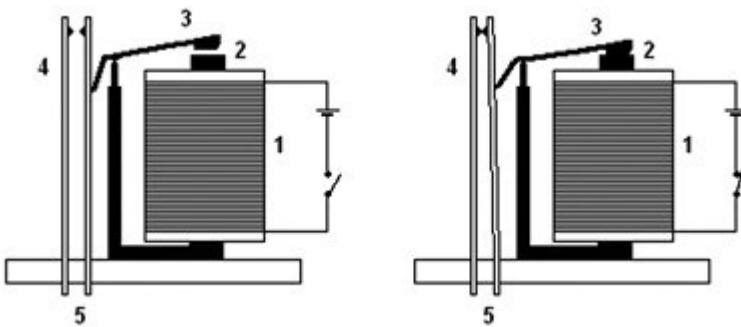
1840, o qual apresentava uma concepção da máquina desenhada por um artista.

Por recomendação da banca que o premiou, Torchi teria recebido mil liras para construir uma versão metálica mais robusta de sua máquina, mas não existe registro de que ela tenha se materializado algum dia. O protótipo original, feito com madeira e arame, ficou em exposição de 1834 a 1837 no Palácio de Ciência e Artes de Brera. Infelizmente, a informação disponível sobre a máquina é escassa demais para que se possa saber com precisão como ela funcionava. Além disso, o protótipo original, reexaminado em 1872, já havia perdido muitas peças, tornando impossível sua reconstrução.

### **3.4 Relê Eletromecânico de Joseph Henry – 1835**

Joseph Henry (Estados Unidos, 1797-1878) foi um cientista americano que estudou o eletromagnetismo, vindo a descobrir um fenômeno muito interessante, chamado indução eletromagnética. Uma bobina – ou seja, um fio condutor enrolado várias vezes ao longo de um cilindro –, quando ativada com energia elétrica, forma um campo magnético capaz de atrair metais. Com este fenômeno pode-se, por exemplo, usar bobinas elétricas para atrair placas metálicas que funcionam como interruptores de corrente em circuitos independentes.

Como mostra o lado esquerdo da [Figura 3.5](#), se não houver energia na bobina (1), não haverá indução eletromagnética no núcleo (2) e, assim, a placa metálica (3) não é atraída e não passa corrente no circuito (4) que está aberto, ou seja, sem contato. No lado direito da figura, aplica-se energia à bobina (1). Isso faz com que o núcleo (2) atraia a placa metálica (3) e esta, devido ao movimento, faz fechar o contato no circuito (4), que passa a conduzir energia.



**Figura 3.5: Relê eletromecânico.<sup>3</sup>**

Diz-se que Henry não percebeu as possíveis implicações de sua invenção e que a apresentava apenas como uma curiosidade para seus alunos. Porém, o relê eletromecânico foi a base tecnológica para a invenção do telégrafo por Samuel Morse (Estados Unidos, 1791-1872), ainda no século XIX.

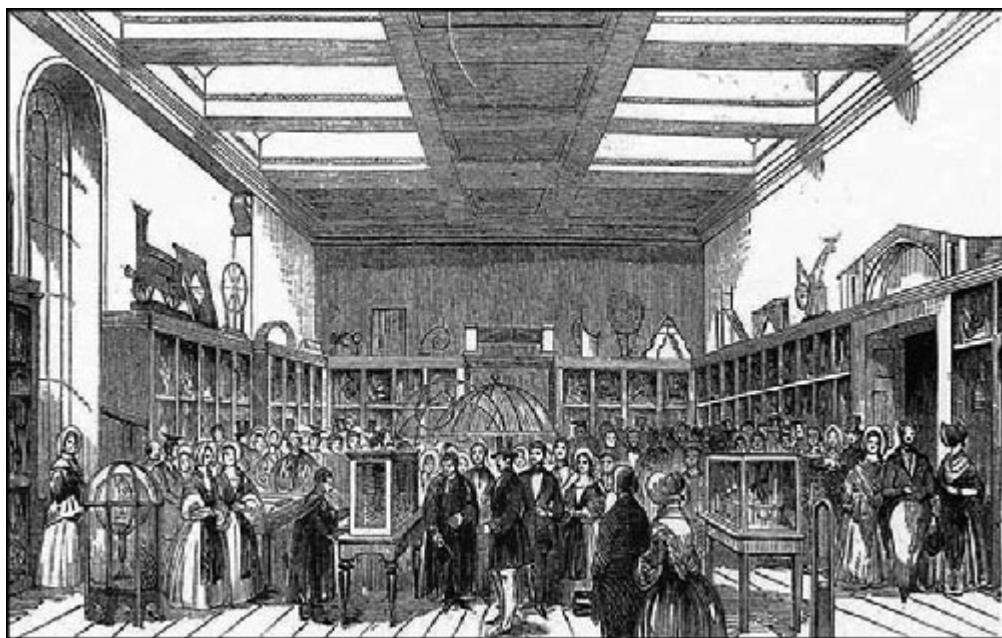
Além disso, o relê eletromecânico foi usado para a construção dos primeiros computadores do século XX, como o Z2 e Z3, os Harvard Mark I e II, dentre outros. Acontece que, embora um conjunto de relês fosse mais caro de implementar do que um sistema de engrenagens, como no caso das calculadoras mecânicas no estilo da Arithmometer, era muito mais fácil reconfigurar um conjunto de relês usando conectores e cabos elétricos. Dessa forma, novos circuitos poderiam ser facilmente construídos ou “programados” nos computadores a relê, coisa que com calculadoras mecânicas seria impraticável.

Os computadores a relê, porém, tiveram vida curta, pois a invenção e uso da válvula eletrônica rapidamente tornou-se o padrão para a construção de computadores no início dos anos 1950.

### 3.5 Máquina Analítica de Babbage – 1837

Charles Babbage ficou famoso por duas grandes criações: a máquina diferencial, que já vimos, e a máquina analítica, parte da qual é mostrada na [Figura 3.6](#). Enquanto a máquina diferencial era

simplesmente um calculador de polinômios, a máquina analítica que ele imaginou seria um computador completo no sentido moderno, ou seja, uma máquina capaz de computar qualquer função computável por uma máquina de Turing – conceito que, aliás, só seria definido quase 100 anos depois.



**Figura 3.6: Parte da máquina analítica em demonstração.<sup>4</sup>**

A máquina analítica era programável. Ela nunca foi construída até hoje; apenas pequenas partes foram feitas ao longo dos anos, mas nunca uma máquina completa. Ela seria programável com o uso de cartões perfurados. Babbage inspirou-se efetivamente nos teares de Jacquard para criar o mecanismo que permitiria que seu aparato fosse programável e, portanto, um computador de propósito geral. Diz-se até que ele teria em sua sala uma cópia do tecido no qual aparecia a imagem de Jacquard, aquele que fora produzido automaticamente a partir de 10 mil cartões, que apresentamos na Figura 2.20.

Descreve-se que a máquina possuiria um conjunto de registradores compostos por colunas de discos numerados, cada coluna

representando um número em base decimal. Como Babbage provavelmente não tinha conhecimento da invenção do relê, ocorrida quase que concomitantemente à sua criação, ele baseou seu design no conceito das calculadoras mecânicas, que na época eram tecnologia já dominada e consistente. A máquina teria ainda uma unidade de processamento denominada “moinho”, a qual seria capaz de realizar pelo menos as quatro operações básicas da aritmética.

Para multiplicar dois números, por exemplo, a máquina deveria ser alimentada com quatro cartões: um para indicar que a operação era de multiplicação, dois que indicariam em quais registradores estariam os operandos e um último indicando em qual registrador o resultado deveria ser armazenado.

Assim, embora Babbage nunca tenha definido uma linguagem de programação específica para sua máquina, especula-se que ela teria sido muito parecida com a linguagem de máquina (Assembly) utilizada pelos computadores modernos. Por exemplo, para calcular o resultado da expressão  $(ax+y)^2$ , a máquina poderia ser programada com o seguinte conjunto de instruções:

- Armazenar  $a$  no registrador V0.
- Armazenar  $x$  no registrador V1.
- Armazenar  $y$  no registrador V2.
- Multiplicar V0 por V1, armazenando o resultado em V3 ( $ax$ ).
- Somar V2 e V3, armazenando o resultado em V4 e também em V5 ( $ax+y$ ).
- Multiplicar V4 por V5 armazenando o resultado em V6 ( $(ax+y)^2$ ).

Essa forma de resolver um problema, trabalhando uma parte de cada vez, é denominada “análise”; daí o aparato ser chamado de “máquina analítica”, pois foi concebido para realizar não apenas um cálculo isolado ou pré-programado, mas qualquer tipo de cálculo possível utilizando-se do processo de análise. Para fazer diferentes

processos de cálculo bastaria alimentar a máquina com um conjunto diferente de cartões.

Os registradores foram nomeados com a letra V, mesmo no tempo de Babbage, visto que eles estariam representando “variáveis”, nome ainda hoje usado. Uma distinção importante foi feita em relação a essas variáveis. No exemplo anterior, as variáveis V0 a V2 são denominadas “variáveis de entrada”, pois contêm os dados iniciais com os quais a máquina é carregada. As variáveis de V3 a V5 são denominadas de “variáveis intermediárias”, porque contêm valores intermediários do cálculo e finalmente a variável V5, neste exemplo, seria a “variável de saída”, pois ela contém o resultado final do cálculo.

A máquina analítica também cunhou no século XIX o termo “memória” (*storage*), fundamental para os computadores modernos. A memória da máquina era considerada o conjunto das colunas que continham os números, ou seja, o conjunto dos registradores.

No que diz respeito ao tratamento de números negativos, deveria se considerar que o disco mais alto da coluna, ou seja, o de mais alta ordem, indicaria a condição do número ali inscrito; se ele fosse par, o número da coluna seria positivo, se ele fosse ímpar, o número seria negativo. Essa codificação ainda é usada nos computadores atuais, apenas com a ressalva de que como eles trabalham com base binária, o dígito de mais alta ordem sendo 0 indica número positivo e sendo 1 indica número negativo.

A máquina seria capaz também de realizar operações mais complexas, como a potenciação. Para calcular  $a^n$ , por exemplo, bastaria carregar o valor de  $a$  em V0 e  $n$  em V1. O cartão de operação indicaria a potenciação aplicada a essas duas variáveis, e indicaria que o valor final deveria ser armazenado em V2. Assim, a máquina faria o seguinte: ela carregaria o valor  $n-1$  em um aparato especial de contagem ao mesmo tempo em que armazenaria o valor de  $a$  em V2. Na sequência, a máquina verificaría o valor contido nesse contador; se fosse igual a 0 a operação estaria concluída.

Caso contrário, o mecanismo iria subtrair 1 do contador e multiplicar V2 pelo valor de  $a$  armazenado em V0. Procedendo dessa forma até que o contador fosse igual a 0, a máquina acabaria registrando em V2 o valor de  $a^n$ .

Caso o número  $n$  fosse negativo, o aparato procederia a uma sequência de divisões ao invés de multiplicações.

Outro conceito que surgiu com essa máquina foi o de constantes em programação. Valores como  $\pi$  ( $\pi$ ), por exemplo, poderiam ser introduzidos diretamente no moinho a partir de cartões especiais, sem precisar ser introduzidos em um dos registradores ou variáveis. Assim, para multiplicar, por exemplo, V0 por  $\pi$  bastaria introduzir um cartão de operação de multiplicação: o cartão que indica que um dos operandos está em V0, o cartão que indica que o segundo operando é a constante  $\pi$  e o cartão que indica em qual variável o resultado deve ser armazenado.

### **3.6 A Primeira Programadora: Ada Lovelace – 1842**

Ada Augusta Byron King (Inglaterra, 1815-1852), mais conhecida como Ada Lovelace – pois foi condessa de Lovelace –, desde cedo interessou-se pelo trabalho de Babbage. Ela traduziu para o inglês um artigo escrito em francês pelo italiano L. F. Menabrea em 1842, ao qual ela adicionou tantas notas que chegaram a exceder o tamanho do próprio artigo. Matemática amadora, Ada é considerada a primeira programadora de computadores da História por sua capacidade de imaginar e descrever estruturas como o desvio condicional, o laço condicional e as sub-rotinas, conceitos que foram incorporados aos computadores modernos e que são essenciais para seu funcionamento.

Dentre suas várias contribuições, Ada percebeu que aqueles mecanismos poderiam ser mais do que meras calculadoras, como a máquina diferencial, capazes apenas de computar polinômios: eram engenhos aptos a simular qualquer outra máquina de computação

desde que programados com os cartões corretos. Lovelace percebeu que a sequência de cartões (programa) era independente dos valores efetivamente operados (aqueles armazenados nas variáveis de entrada), dando origem ao conceito de “procedimento”, fundamental nas linguagens de programação: “*The operating mechanism can even be thrown into action independently of any object to operate upon (although of course no result could then be developed).*”<sup>5</sup>

Ada destacou também que tal máquina seria facilmente capaz de trabalhar com símbolos quaisquer e não apenas com números. Se uma linguagem simbólica fosse criada, por exemplo, para representar as harmonias musicais, ela acreditava que a máquina seria capaz de compor quaisquer músicas, desde que devidamente instruída para isso. “*Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.*”<sup>6</sup>

Outro exemplo desta capacidade de tratamento simbólico da máquina foi a alusão ao sinal positivo ou negativo dos números nas colunas de variáveis; os números pares e ímpares poderiam ser facilmente substituídos pelos sinais “+” e “-” ([Figura 3.7](#)).

$V_1$	$V_2$	$V_3$	$V_4$	&c.
+	+	+	+	
0	0	0	0	&c.
0	0	0	0	
0	0	9	0	
5	7	8	0	&c.
$a$	$n$	$x$	$ax^n$	
$\overbrace{\phantom{aaa}}^{ax^n}$				

**Figura 3.7: Representação de símbolos (os sinais “+” no alto) e números nas colunas de variáveis.<sup>7</sup>**

Em uma de suas notas, Ada define claramente o comando de atribuição, inclusive fazendo a distinção necessária com relação à expressão algébrica similar. Ela observa que as colunas de discos que contêm as variáveis são efetivamente depósitos de números (*storehouse*) e que alguns destes depósitos, especialmente os utilizados para armazenar valores intermediários nos cálculos, podem ser utilizados mais de uma vez. Por exemplo, se V3 já armazenou um valor intermediário que não vai mais ser necessário nos cálculos, então uma próxima computação pode armazenar na mesma variável V3 o valor de um novo cálculo. Assim, as variáveis realmente podem assumir vários significados diferentes ao longo do tempo e não necessariamente o valor de uma combinação específica de outras variáveis.

Para indicar os vários instantes de tempo ou estados dessas variáveis ela utiliza índices sobreescritos. Assim, o estado inicial de uma variável, quando ela contém apenas zeros, seria indicado por  $^0V$ . Depois de armazenar um valor qualquer nessa variável, ela passaria a ser indicada como  $^1V$ . Se posteriormente outro valor

qualquer fosse armazenado na mesma variável, ela passaria a ser denominada  ${}^2V$ , e assim por diante.

Assim, a definição de uma variável passando de um estado no qual ela tem um valor específico para um novo estado, é indicada, por exemplo, da seguinte forma:  ${}^{m+1}V_n = {}^qV_p + {}^mV_n$ . Esse é exatamente o significado da instrução de atribuição, conforme utilizado nas modernas linguagens de programação. Ada observa que a notação algébrica usual  $V_n = V_p + V_n$  seria inadequada, pois se a atribuição fosse entendida como igualdade essa equação seria verdadeira apenas se  $V_p$  fosse zero, o que não é a intenção.

Em função da possibilidade de realizar diferentes escolhas na ordem em que as operações são realizadas, ela tece comentários referentes à otimização de tempo da operação da máquina: “*In almost every computation a great variety of arrangements for the succession of the processes is possible, and various considerations must influence the selection amongst them for the purposes of a Calculating Engine. One essential object is to choose that arrangement which shall tend to reduce to a minimum the time necessary for completing the calculation.*”<sup>8</sup>

Ao explicar como calcular uma série finita de termos matemáticos, ela cunha o conceito de ciclo ou laço repetitivo: “*A cycle of operations, then, must be understood to signify any set of operations which is repeated more than once.*”<sup>9</sup> Ela não apenas define o laço repetitivo, mas apresenta um exemplo de laço dentro de outro laço e especula que poderiam haver problemas que exigissem “ciclos de ciclos de ciclos” em uma regressão indeterminada (como de fato acontece com programas de computador). O mecanismo de laço poderia ser implementado pela máquina caso ela pudesse retroceder o conjunto de cartões  $n$  posições e reiniciar dali sua operação. Tecnicamente, isso era possível.

Na nota G ao manuscrito de Menabrea, Ada apresenta um algoritmo completo para computar os números de Bernoulli, o qual é mostrado

na [Figura 3.8](#). Esse algoritmo especificamente é considerado o primeiro programa de computador já escrito no mundo.

**Figura 3.8:** O primeiro programa de computador.<sup>10</sup>

**3.7 A Máquina Relacional de Alfred Smee – 1851**

Contemporâneo de Babbage e Lovelace, o Dr. Alfred Smee (Inglaterra, 1818-1877) foi também um visionário que, seguindo a linha de Hobbes, Llull, Kircher e outros, sonhou com a possibilidade de automatizar não apenas o cálculo matemático, mas toda a forma de raciocínio humano. Em 1851, ele publicou um importante livro (*The Process of Thought Adapted to Words and Language. Together with a Description of the Relational and Differential Machines*) que, além de ser um tratado sobre lógica e raciocínio, especulou sobre a possibilidade de que o pensamento humano fosse reproduzível mecanicamente ou eletricamente. Diz ele: “Pelas leis que já foram detalhadas, parece que o pensamento é regido por princípios fixos. Em função do conhecimento destes princípios, ocorreu-me que

dispositivos mecânicos poderiam ser construídos para obedecer às mesmas leis, e produzir os resultados que se considerariam obtêveis apenas pela operação da própria mente.”<sup>11</sup>

Inicialmente, ele propunha que ideias eram formadas a partir de propriedades cuja presença ou ausência poderiam ser verificadas como, por exemplo, aspereza, cor, transparência etc. Cada palavra teria que ser denotada por um símbolo e esses símbolos seriam arranjados de forma que suas relações com outros símbolos pudessem ser representadas. Assim, o arranjo geométrico dos símbolos seria fundamental para o correto funcionamento da máquina.

Ele obteve essa observação a partir da suposição de que a presença ou ausência de uma propriedade seria registrada no cérebro a partir da presença ou ausência de um sinal elétrico em um nervo. Assim, vários nervos ativados a partir de sensores de visão, tato, audição etc. poderiam, por sua combinação, produzir as ideias mais complexas que o cérebro é capaz de idealizar. Tal princípio não se diferencia muito daquele por trás das modernas redes neurais artificiais.

Smee imaginava ainda que a memória (por exemplo, lembrar de um objeto redondo e quente) poderia ser o resultado da reativação das fibras nervosas correspondentes no próprio cérebro, sem que o estímulo sensorial estivesse necessariamente presente.

Faz em seu livro ainda uma interessante análise das classes de palavras, como substantivos, adjetivos, verbos, pronomes etc. e sua possível representação no arcabouço teórico da representação de ideias.

Em relação às máquinas, a primeira a ser proposta por Smee foi chamada de Máquina Relacional ([Figura 3.9](#)). Ela estaria representando fisicamente uma hierarquia com todas as propriedades possíveis e suas combinações em ideias de nível cada vez mais elevado; na prática, seria a representação física de uma espécie de taxonomia sistemática. A presença ou ausência de uma

propriedade em qualquer nível seria representada pelo movimento de dobradiças. Estando a dobradiça na posição “aberta”, a propriedade estaria presente; estando fechada, a propriedade estaria ausente.

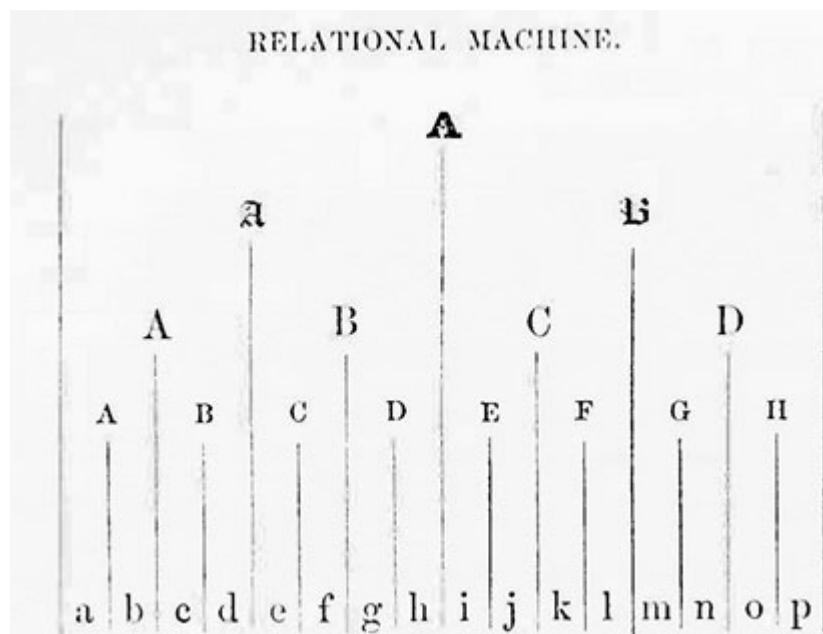


Figura 3.9: Esboço da máquina relacional de Smee.<sup>12</sup>

Smee também propôs uma “Máquina diferencial”, mas que nada tinha de relação com a de Müller e Babbage. A Máquina Diferencial de Smee seria capaz de calcular a diferença entre dois conceitos. Com essa máquina, ele poderia representar em duas partes, através de barras móveis, duas asserções a serem comparadas. Assim, o movimento das barras indicaria se cada propriedade possível é absolutamente conhecida ou desconhecida.

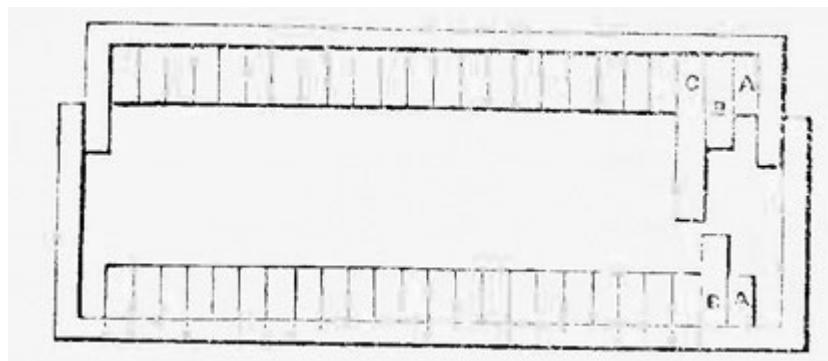


Figura 3.10: Máquina diferencial de Smee.<sup>13</sup>

Na parte de cima da figura as barras poderiam ser colocadas em três posições, indicando similaridade, dissimilaridade ou desconhecido (posições A, B e C) em relação a cada uma das propriedades. Colocando as duas séries juntas, em função dos encaixes, poder-se-ia ler a resposta que indicaria se a similaridade é total, provável, possível ou não existente. Assim, se as duas partes da máquina se encaixassem perfeitamente, as ideias seriam perfeitamente compatíveis (por exemplo, “dor de cabeça e febre” com “gripe”). Se em um dos lados uma das propriedades fosse desconhecida de forma que quando juntadas as metades não encaixassem perfeitamente, ficando um espaço entre elas, a resposta seria “provável”. Se nos dois lados na máquina a propriedade fosse desconhecida, então o espaço seria ainda maior e a resposta “possível”. Finalmente, se nos dois lados da máquina as propriedades discordassem, a resposta seria “não”, indicando a incompatibilidade das ideias. Assim como no caso das máquinas de Korsakov, ela estaria comparando padrões mecanicamente.

Embora não haja evidências de que seus engenhos tenham sido efetivamente construídos, exceto talvez por pequenos protótipos – visto que o próprio autor duvidava de sua viabilidade técnica –, o livro foi uma leitura bastante popular no final do século XIX e possivelmente influenciou outras mentes na direção da criação de máquinas efetivamente capazes de raciocinar.

### **3.8 Teletrofone de Antonio Meucci – 1856**

Interessante observar que a maioria das linhas de tempo da história da computação não inclui a invenção do telefone apesar de hoje a telecomunicação ser um dos seus componentes mais importantes. Afinal, computadores sem conectividade não eram tão populares, não é mesmo?

Quem inventou o telefone? Cuidado! Antes de responder “Alexander Graham Bell”, saiba que o verdadeiro inventor do telefone foi Antonio Meucci (Italia, 1808-1889). Meucci migrou para os Estados Unidos em 1850 e consta que, em 1856, cansado de subir e descer escadas para supervisionar sua esposa permanentemente acamada devido a uma paralisia, ele inventou um dispositivo chamado “teletrofone” (telégrafo falante). Desta maneira, poderia se comunicar de seu escritório com a esposa no quarto, que ficava na parte superior da casa.

Ele conseguiu conceber este dispositivo ao perceber que sinais elétricos podiam conduzir e reproduzir sons enviados por um fio de telégrafo.

Meucci era pobre e registrou apenas uma patente provisória ([Figura 3.11](#)), mas não tinha meios de pagar a patente definitiva. Assim, acabou vendendo seu protótipo para Alexander Graham Bell (Reino Unido, 1847-1922), que o patenteou em 1876. Ao ver o sucesso de seu invento, Meucci processou Graham Bell pelos direitos da invenção, mas faleceu durante o processo e, em função disso, o caso foi encerrado.

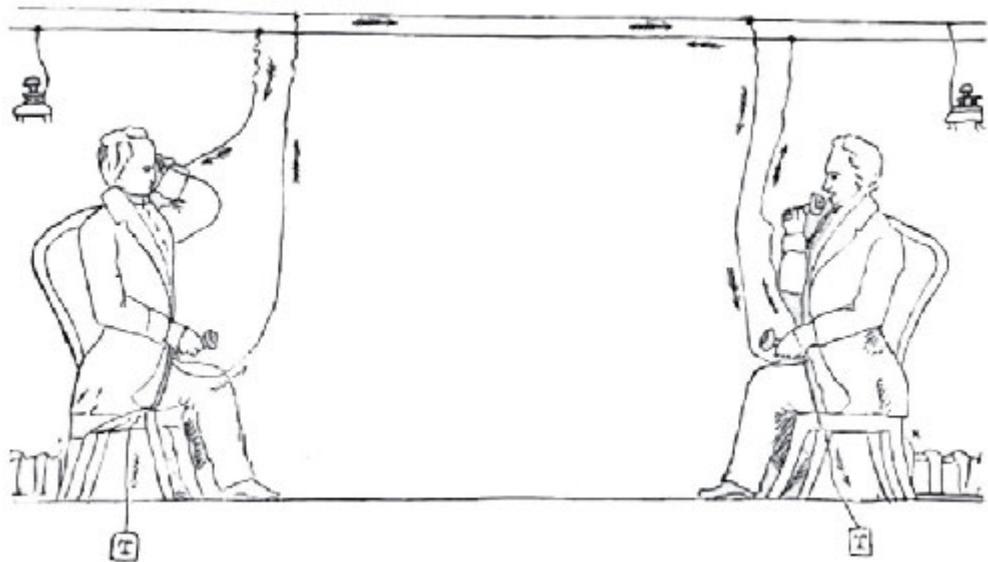


Figura 3.11: Desenho da patente provisória de Meucci.<sup>14</sup>

Porém, em 2002 o Congresso americano reconheceu formalmente que o inventor do telefone foi mesmo Meucci e que a patente de Graham Bell se baseava em afirmações falsas. No final, se Meucci possuísse os 10 dólares necessários para pagar sua patente teria provavelmente ficado rico e sido reconhecido em seu tempo.

### 3.9 Álgebra Booleana – 1854

George Boole (Inglaterra, 1815-1864) deu uma contribuição fundamental à história da lógica e, posteriormente, da computação ao perceber – acredita-se que pela primeira vez na história –, que as leis da lógica se comportam como uma álgebra, ou seja, de forma semelhante à aritmética. Em seu livro de 1854<sup>15</sup> ele observa, entre outras coisas, que se  $x$  e  $y$  são ideias como “homem” e “mulher”, e se o sinal “+” representar uma operação lógica como “e” ou “ou”, então pode-se afirmar que  $x+y = y+x$ , ou seja “homens e mulheres” é igual a “mulheres e homens” ou ainda “homens ou mulheres” é igual a “mulheres ou homens”. Dessa forma, os operadores lógicos “e” e “ou” seriam comutativos.

Rapidamente, ele concluiu também que esses operadores podem ser distribuídos algebricamente. Por exemplo, se  $z$  significa “europeu”, então, “homens europeus e mulheres europeias” é também equivalente a “homens e mulheres europeus”, ou seja,  $zx+zy = z(x+y)$ .

Ele ainda observa que afirmar a mesma coisa duas vezes equivale a simplesmente afirmar a coisa, assim, em sua álgebra  $x+x = x$  ou ainda  $x^2 = x$ . Como a álgebra de Boole lida apenas com valores verdade (verdadeiro ou falso), ele associa tais valores a 0 e 1, e observa que, coincidentemente ou não,  $0^2 = 0$  e  $1^2 = 1$ . Assim, ele criou uma álgebra com apenas dois valores: 0 e 1.

A álgebra de Boole foi aperfeiçoada ao longo do século XIX por outros matemáticos, como William Stanley Jevons (Inglaterra, 1835-1882), Friedrich W. K. Ernst Schröder (Alemanha, 1841-1902), Edward V. Huntington (Estados Unidos, 1874-1952) e Marshall H. Stone (Estados Unidos, 1903-1989), até chegar à forma moderna como a conhecemos hoje. Em especial, Stone demonstrou que a álgebra de Boole é isomórfica (equivalente) à álgebra dos conjuntos.

A álgebra de Boole teve forte influência na construção dos computadores e linguagens de programação pelo menos desde a década de 1930, quando Claude Shannon (Estados Unidos, 1916-2001) defendeu em sua dissertação de mestrado no MIT (Massachusetts Institute of Technology) que circuitos elétricos baseados na álgebra de Boole poderiam representar quaisquer relações lógicas ou aritméticas, criando assim a base teórica para a construção de computadores eletrônicos.

Tão importante e influente foi o trabalho de Boole que quase todas as modernas linguagens de programação têm um tipo de dados especial cujos valores são 0 e 1, ou verdadeiro e falso. Esse tipo é denominado em quase todas elas como “booleano”.

### **3.10 Calculadora de Coluna Única de Caroline Winter – 1859**

Durante o século XIX foram desenvolvidos vários modelos de calculadoras mecânicas que eram, em termos de projeto, bem mais simples do que os modelos de Schickard, Pascal e Leibniz: eram as chamadas “calculadoras de coluna única”, porque em vez de somarem números de vários dígitos um de cada vez, permitiam apenas somar uma coluna de dígitos de cada vez. Por exemplo, para somar:

678

134

+389

Seria necessário primeiramente somar 8+4+9 (coluna das unidades) e anotar o resultado 21 em uma folha de papel. Depois, soma-se 7+3+8+2 (coluna das dezenas, sendo 2 o “carry”) e anota-se o resultado 20 no papel. Finalmente soma-se 6+1+3+2 (sendo 2 o “carry”) e anota-se o resultado 12 no papel. O resultado final é o resultado da última soma (coluna das centenas, no exemplo) e o último dígito de cada uma das outras colunas até a das unidades, chegando-se assim a 1.201, neste exemplo.

Considerando-se que séries muito longas de números poderiam ter que ser somadas (por exemplo, uma lista de compras em um mercado), haveria possibilidade de o *carry* ser muito grande. Entretanto, essas calculadoras extremamente simples, caso funcionassem, ainda poderiam ser muito úteis para quem precisasse somar longas séries de números que estivessem convenientemente anotados em uma lista.

Uma dessas calculadoras de coluna única pode ter sido inventada por uma mulher, segundo especulações do historiador Georgi Dalakov. Trata-se de uma patente emitida nos Estados Unidos em 1859 para uma pessoa misteriosamente chamada de “C. Winter”, da cidade de Piqua, Ohio. Não se pode ter certeza sobre quem ela foi, mas a única C. Winter mencionada nos registros de Piqua da época era Caroline Winter, uma comerciante que possivelmente veria

grande utilidade para uma calculadora simples capaz de somar colunas de números de 1 a 9. Como a máquina permitia chegar até 699, pode-se especular que ela poderia dar conta de qualquer lista de compras realizada no século XIX.

A máquina baseava-se em um mecanismo muito simples, no qual nove teclas, cada qual representando um dígito de 1 a 9, moviam de uma a nove posições uma engrenagem com 100 posições (100 dentes). Isso era obtido fazendo-se com que cada tecla fosse movida até o ponto de parada, que era incrementalmente maior de 1 até 9 para permitir o movimento exato dos dentes conforme o dígito somado, como pode ser visto na [Figura 3.12](#).



Figura 3.12: Calculadora original de C. Winter.<sup>16</sup>

Assim, a tecla  $n$  moveria  $n$  dentes da engrenagem de 100 dentes que, por sua vez, estava ligada ao ponteiro que se vê no alto da figura. O mostrador grande no qual esse ponteiro se move tem 100 divisões que estão numeradas de 5 em 5: 0, 5, 10, 15 etc.

Um mecanismo simples de transmissão de movimento fazia um ponteiro em um mostrador menor mover-se também. Esse mostrador menor, visível na figura dentro do mostrador maior, tinha marcações de 0 a 6 e movia-se uma posição cada vez que o ponteiro maior dava uma volta completa, ou seja, quando o valor ultrapassasse 100.

Completa o mecanismo da máquina um conjunto de molas, colocado na parte traseira das alavancas, que fazia com que cada tecla retornasse à sua posição original após ter sido pressionada, como pode ser visto na [Figura 3.13](#).



**Figura 3.13: Parte traseira da calculadora de Winter,  
mostrando as molas que levavam as teclas de volta à sua posição original.<sup>17</sup>**

Este projeto foi o quarto dentre as somadoras mecânicas patenteadas nos Estados Unidos, sucedendo os de Du Bois D. Parmelee (1850), Orlando Lane Castle (1857) e Leonard Nutz

(1858). No mundo, teria sido a sexta patente do tipo, sendo precedida também por Luigi Torchi (1834) e Jean-Baptiste Schwilgué (1844). Porém, não há registro de outra patente desse tipo obtida por uma mulher na história da computação.

### **3.11 A Máquina de Escrever – 1867**

A máquina de escrever, seja mecânica ou elétrica, certamente pode ser considerada uma antecessora dos atuais teclados e impressoras de computador. Em uma charge moderna, um pai apresenta uma máquina de escrever ao seu filho pequeno, que fica impressionado com a capacidade que ela tem de imprimir ao mesmo tempo em que se digita.

A máquina de escrever pode ser considerada um marco tão importante para a história da escrita quanto a prensa de Gutenberg. Porém, seu início não é muito preciso. O registro mais antigo de uma máquina deste tipo vem de uma patente britânica, concedida ao inventor Henry Mill (Inglaterra, 1683-1771) em 1714. O registro da patente informava que ele “(...) inventou e aperfeiçoou uma máquina ou método artificial para impressão ou transcrição de letras, uma após a outra, como na escrita (...)”<sup>18</sup>. Porém, nenhum registro de que essa máquina tenha sido algum dia construída existe, nem mesmo um esquema mais detalhado de seu funcionamento.

A primeira máquina de escrever que, apesar de não ter sobrevivido até nossos dias, deixou rastros de sua existência, foi inventada pelo italiano Pellegrino Turri em 1808. Segundo a história, essa invenção teria ocorrido em função de uma paixão. Conta-se que Turri era apaixonado pela condessa Carolina Fantoni da Fivizzono, e que frequentemente se correspondiam por carta. Ocorre que Carolina foi ficando cega, e chegou a um ponto em que ela não sabia mais se estava escrevendo em linha reta, nem se estava borrando tinta sobre o papel. Pessoas cegas ou iletradas, naquela época, usualmente ditavam suas cartas a quem pudesse escrever, mas

isso possivelmente seria uma indiscrição neste caso. Assim, Turri inventou e construiu uma máquina capaz de imprimir letras no papel utilizando um sistema de impressão baseado em papel carbono, cuja invenção é também atribuída a ele. De fato, a grande maioria das máquinas de escrever até o final do século XX baseou-se em martelos nos quais a forma da letra moldada batia em uma fita embebida em tinta, semelhante ao papel carbono, deixando sua marca sobre a folha.

Nem a máquina de Turri nem seu projeto sobreviveram ao tempo, mas as evidências de que ela realmente existiu são fragmentos de cartas escritas por Carolina. Pode-se considerar assim que essa foi uma das primeiras interfaces especificamente criadas para pessoas com necessidades especiais.

Suspeita-se também que na verdade a máquina tenha sido inventada em 1802 por Agostino Fantoni para ajudar a sua irmã também cega, e que Turri teria apenas aperfeiçoado o projeto ao criar e introduzir o papel carbono.

A primeira máquina comercialmente explorada e que sobreviveu até os nossos dias foi a “bola de escrever” ([Figura 3.14](#)) de Rasmus Malling-Hansen (Dinamarca, 1835-1890), que começou a ser produzida e vendida em 1865. Consta que o filósofo Friedrich Nietzsche teria ganhado uma dessas de sua mãe e de sua irmã, e que a teria odiado.



Figura 3.14: Bola de escrever de Malling-Hansen.<sup>19</sup>

No Brasil, considera-se que a máquina de escrever tenha sido inventada pelo padre Francisco João de Azevedo (Brasil, 1814-1880) em 1865, que a teria construído em madeira apenas com canivetes e lixas. Consta que, apesar de o padre ter até recebido uma medalha do imperador Dom Pedro II pela invenção, ela jamais foi patenteada e, assim, acabou não recebendo o reconhecimento internacional que lhe era devido.

Apenas três anos depois, os americanos Christopher Latham Sholes (Estados Unidos, 1819-1890), Carlos Glidden (Estados Unidos, 1834-1877) e Samuel W. Soulé (Estados Unidos, 1830-1875) projetaram uma máquina bastante semelhante à do Padre Francisco. Ela começou a ser produzida pelo departamento de máquinas de costura da Remington, fabricante de armas fundada em 1816.

Talvez por isso as primeiras máquinas tivessem seu *carriage return* (o comando que fazia a máquina avançar para a linha seguinte e retornar ao início da linha) implementado como um pedal, como nas máquinas de costura ([Figura 3.15](#)). Pode-se dizer então que no início a tecla “*Return*”, ancestral do “*Enter*” era acionada com o pé.



**Figura 3.15:** Máquina de escrever Sholes & Glidden, fabricada pela Remington.<sup>20</sup>

Esta máquina foi um sucesso comercial, e ela já utilizava o padrão QWERTY de disposição das teclas, até hoje usado na maioria dos teclados de computador. Originalmente o objetivo dessa disposição era evitar que duas teclas vizinhas fossem acionadas em sequência, pois se isso ocorresse haveria uma grande chance de que seus respectivos martelos se chocassem ou se emaranhassem. Assim, a disposição das teclas visava minimizar a possibilidade de tais

sequências de letras, além de distribuir as letras mais frequentes nos lados direito e esquerdo do teclado, de forma a equilibrar a quantidade de ações realizadas por cada uma das mãos e por cada um dos dedos.

A Sholes & Glidden imprimia apenas letras maiúsculas. Contudo, sua sucessora, a Caligraph 2 de 1882, já imprimia também minúsculas. Em vez da tecla *shift*, porém, ela tinha uma tecla específica para cada letra maiúscula e minúscula, o que a tornava uma máquina bastante grande.

Um dos maiores problemas com as primeiras máquinas de escrever é que, como os martelos batiam na fita carbono que estava sobre o papel, o datilógrafo não conseguia ver o que estava escrevendo até que rolasse o papel para cima. Esse problema foi resolvido a partir de 1891 com a Daugherty Visible, a qual, com um mecanismo diferenciado de ação, permitia que o datilógrafo enxergasse a letra que acabava de ser batida.

Embora a produção de máquinas de escrever em larga escala tenha praticamente acabado no mundo no início do século XXI, elas ainda são bastante usadas nas zonas mais pobres do planeta, especialmente porque são puramente mecânicas e não necessitam de eletricidade para funcionar.

### **3.12 O Piano Lógico de Jevons – 1869**

William Stanley Jevons (Inglaterra, 1835–1882) foi um economista que além de ter apresentado contribuições significativas ao trabalho de Boole, propôs e construiu uma máquina capaz de realizar inferências lógicas. Interessante observar o início do artigo de Jevons (1869), no qual ele apresenta o projeto desta máquina, pois faz referência a vários outros inventos relatados nesta história: ábaco, bastões de Napier, calculadoras de Pascal, Morland, Leibniz e Colmar, máquina diferencial e analítica de Babbage, silogismos, demonstrador de Stanhope e até a Ars Magna de LLull.

A máquina de Jevons era capaz de eliminar, em uma tabela-verdade, todas as opções contraditórias a uma expressão lógica que fosse inserida em seu teclado, restando assim apenas as opções compatíveis com a expressão lógica.

De forma simplificada, a máquina que chegou a ser construída, conforme mostrado na [Figura 3.16](#), permitia representar em seu painel todas as combinações de até quatro premissas: A, B, C e D; e suas negações a, b, c e d. Suponha, por exemplo, que A=animal, B=marinho, C=mamífero e D=extinto, então ABCd representaria os animais que são marinhos, mamíferos e que não estão extintos. Por outro lado, a expressão C→A (C implica A, ou seja, todos os mamíferos são animais) faz com que quaisquer combinações de símbolos que contenham aC (mamífero que não é animal) sejam inválidas, ou seja, aBCD, abCD, aBCd e abCd.



Figura 3.16: Piano Lógico de Jevons.<sup>21</sup>

Assim, a máquina de Jevons tinha em seu painel frontal todas as 16 combinações possíveis de valores-verdade para as quatro premissas:

A	A	A	A	A	A	A	a	a	a	a	a	a	a	a	a
B	B	B	B	b	b	b	B	B	B	b	b	b	b	b	b

C	C	c	c	C	C	c	c	C	C	c	c	C	C	c	c
D	d	D	d	D	d	D	d	D	d	D	d	D	d	D	d

Se fosse indicado no seu teclado que  $C \rightarrow A$ , por exemplo, um sistema de alavancas e polias faria desaparecer todas as combinações inconsistentes com essa afirmativa, resultando em um quadro como o seguinte:

A	A	A	A	A	A	A	A	a	a	a	a	a	a	a	a
B	B	B	B	b	b	b	b	B	B	B	b	b	b	b	b
C	C	c	c	C	C	c	c	C	C	c	c	C	C	c	c
D	d	D	d	D	d	D	d	D	d	D	d	D	d	D	d

A entrada subsequente de mais afirmações faria a máquina eliminar outras opções, restando apenas as opções consistentes com todas as afirmações. Por exemplo, se na sequência tivesse entrado “a ou b”, seriam eliminadas quaisquer combinações com A e B simultaneamente, resultado em:

A	A	A	A	A	A	A	A	a	a	a	a	a	a	a	a
B	B	B	B	b	b	b	b	B	B	B	b	b	b	b	b
C	C	c	c	C	C	c	c	C	C	c	c	C	C	c	c
D	d	D	d	D	d	D	d	D	d	D	d	D	d	D	d

Assim, apenas as oito combinações restantes no painel são consistentes com  $(C \rightarrow A) \wedge (a \vee b)$ .

Esse mecanismo, embora de utilidade limitada, foi segundo o próprio Jevons muito útil em suas aulas de lógica. Além disso, ele é

digno de nota pois foi a primeira máquina efetivamente capaz de realizar inferências lógicas mais rápido do que um humano seria.

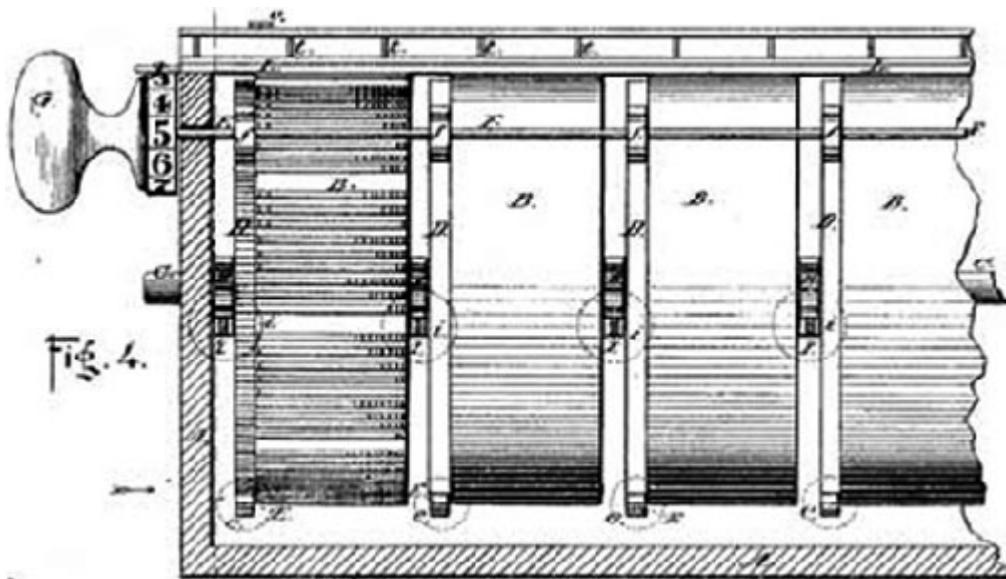
### 3.13 Multiplicação Direta de Edmund Barbour – 1872

Edmund Dana Barbour (Estados Unidos, 1841-1925) recebeu uma patente em 1872 pela invenção de uma calculadora capaz de realizar multiplicações e divisões com um único giro de manivela. Basicamente, ele percebeu que para multiplicar  $7 \times 6$ , por exemplo, não era necessário somar 7 na coluna das unidades 6 vezes, mas bastaria somar 4 na coluna das dezenas e 2 na coluna das unidades, já que  $7 \times 6 = 42$ .

A dificuldade maior na construção desta máquina estava no fato de que era necessário ter todas as combinações da tabuada fisicamente representadas. Entretanto isso era possível com um engenhoso arranjo de cilindros escalonados, como os de Leibniz, com registros para dezenas e unidades. O arranjo físico destes números era bem semelhante ao arranjo dos bastões de Napier. A [Figura 3.17](#) mostra um detalhe do projeto de Barbour com os vários cilindros escalonados.

Assim, para multiplicar  $567 \times 3$ , por exemplo, a máquina seria posicionada de forma que o cilindro das unidades ficasse na posição 7, o das dezenas na posição 6 e o das centenas na posição 5. O multiplicador seria, então, alocado para a posição 3 e com um único giro de manivela os cilindros do acumulador seriam movimentados da seguinte forma:

- Para o cilindro das unidades, 2 nas dezenas e 1 nas unidades:  $3 \times 7 = 21$ .
- Para o cilindro das dezenas, 1 nas centenas e mais 8 nas dezenas:  $3 \times 6 = 18$ , totalizando  $180 + 21 = 201$ .
- Para o cilindro das centenas, 1 nos milhares e 5 nas centenas:  $3 \times 5 = 15$ , totalizando  $1500 + 201 = 1701$ .



**Figura 3.17: Detalhe do projeto de calculadora de Barbour.<sup>22</sup>**

Para multiplicadores com mais de um dígito, a máquina permitia que se fosse passando um de cada vez e acumulando o resultado. Por exemplo, para multiplicar  $123 \times 456$ , inicialmente colocava-se o multiplicando 123 e o último dígito do multiplicador (6) na casa das unidades, assim, com um giro se obteria o resultado  $123 \times 6 = 738$ . Em seguida, colocar-se-ia o segundo dígito do multiplicador (5) na casa das dezenas, obtendo-se  $123 \times 50 = 6150$ , e acumulando  $738 + 6150 = 6.888$ . Finalmente, colocar-se-ia o terceiro dígito do multiplicador (4) na casa das centenas, obtendo-se  $123 \times 400 = 49200$  e acumulando  $6888 + 49200 = 56088$ , ou seja, o resultado de  $123 \times 456$ .

Após Babbage e Müller, Barbour teria sido um dos primeiros a propor um mecanismo de impressão para sua máquina. Porém, não há evidências fortes de que alguma das máquinas concebidas por Barbour tenha sido efetivamente construída por ele.

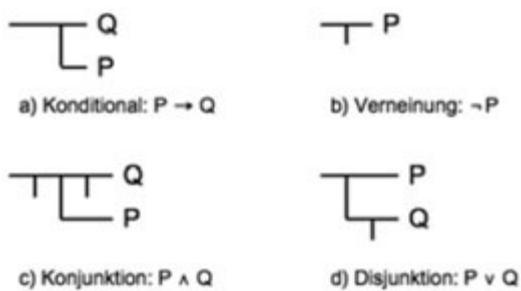
O mecanismo de Barbour, embora não implementado pelo próprio, foi aperfeiçoado por outros e passou a ser a forma padrão de implementação de multiplicação. Otto Steiger (Suíça, 1858-1923), baseado no mecanismo em questão, criou uma calculadora de

multiplicação direta, chamada “Millionaire”, que foi um sucesso de vendas, apesar de custar o equivalente ao preço de um carro na época. Ela foi produzida até 1935.

### 3.14 Lógica de Predicados de Frege – 1879

No ano de 1879, o matemático Gottlob Frege (Alemanha, 1848-1925) publicou um livro chamado *Begriffsschrift*, considerado a contribuição mais importante na história da lógica formal desde os silogismos de Aristóteles. A palavra, em alemão, parece não ter uma tradução exata em português, mas significa algo como “escrita de conceitos” e sua importância vem do fato de ter sido o primeiro trabalho a sistematizar com sucesso a lógica de predicados.

Frege foi o primeiro a introduzir o conceito de variáveis quantificadas universal e existencialmente. Sua notação original era bidimensional e estranha ([Figura 3.18](#)), mas atualmente esses quantificadores são representados pelos símbolos  $\forall$  e  $\exists$ . Pode-se escrever, por exemplo,  $\forall(x) x \text{ é humano} \rightarrow x \text{ é mortal}$ , ou seja, “para todo  $x$ , se  $x$  é humano,  $x$  é mortal”. Na moderna lógica de predicados, a atribuição de uma propriedade  $p$  a uma variável  $x$  usualmente é escrita como  $p(x)$ , assim, a expressão acima também poderia ser representada como  $\forall(x) \text{ humano}(x) \rightarrow \text{mortal}(x)$ .



**Figura 3.18: Exemplos na notação de Frege.**<sup>23</sup>

Com a notação formal de Frege, tornou-se possível eliminar certas ambiguidades com as quais trabalhos anteriores de lógica não

lidavam bem. Por exemplo, poder-se-ia dizer que “todos os rapazes amam uma moça”. Mas não fica claro se cada um ama uma moça diferente ou se todos amam a mesma moça. Com a lógica de predicados de Frege, a sentença teria que ser escrita como  $\forall(x)$  rapaz( $x$ )  $\rightarrow \exists(y)$  moça( $y$ )  $\wedge$  ama( $x,y$ ), significando “para todo rapaz existe uma moça a qual ele ama”, ou  $\exists(y)$  moça( $y$ )  $\wedge \forall(x)$  rapaz( $x$ )  $\rightarrow$  ama( $x,y$ ), significando “existe uma moça a qual todos os rapazes amam”. A lógica formal de predicados elimina, assim, certas formas de ambiguidade e imprecisão que muitas vezes a linguagem natural não consegue evitar.

Essa nova forma de escrever proposições lógicas, além de permitir o desenvolvimento da matemática e da lógica no início do século XX, também foi a base para muitas formas de representação de conhecimento em computadores mais tarde, inclusive em linguagens de programação como PROLOG. Além disso, os conceitos de função e variável usados por Frege tornaram-se parte da maioria das linguagens de programação existentes.

Embora haja autores que discordem disso, parece que o trabalho de Frege não foi muito bem recebido por seus colegas no início. Coube a Bertrand Russell (Reino Unido, 1872-1970) colocar a obra no seu devido destaque ao publicar os *Principia Mathematica* entre 1910 e 1913. Em função disso, Frege absteve-se de tentar publicar vários de seus textos, mas consciente de seu valor, pediu a seu filho adotivo Alfred que os guardasse muito bem, pois um dia seria muito mais apreciado do que em vida. Alfred levou os escritos para a Universidade de Münster para que fossem bem guardados, mas infelizmente um bombardeio Aliado destruiu a universidade e os escritos em 1945. Apesar de cópias de alguns documentos terem sido feitas, boa parte do que Frege escreveu perdeu-se.

### 3.15 NE e NOU – 1880

Por volta de 1880, Charles Peirce (Estados Unidos, 1839-1914) fez uma descoberta no mínimo interessante. Ele observou as tabelas

verdade para operadores lógicos booleanos de duas variáveis. Já se sabia que era possível conceber 16 diferentes operadores lógicos para duas variáveis. Alguns desses operadores são facilmente descritos em língua natural, como por exemplo:

- Conjunção: “A e B” é verdadeiro se ambos os valores de A e B forem verdadeiros.
- Disjunção: “A ou B” é verdadeiro se pelo menos um dos dois valores for verdadeiro.
- Implicação: “ $A \rightarrow B$ ” é verdadeiro se ambos forem verdadeiros ou se A for falso. Essa expressão, portanto, só é falsa se A for verdadeiro e B falso.
- Negação: “não A” é verdadeiro se A for falso, não importando o valor de B.

Mais 12 outros operadores podem ser construídos para diferentes definições sobre as condições dos valores de A e B. Contudo, Peirce percebeu que dois desses operadores são muito especiais: NE e NOU (em inglês, NAND e NOR). Por definição, “A ne B” é verdadeiro se A ou B ou ambos forem falsos. Assim, “A ne B” será falso se A e B forem verdadeiros. Portanto essa expressão é equivalente a “não (A e B)”.

Até aí nada demais. Entretanto, essa operação tem uma propriedade singular já que combinações de NEs permitem simular todos os demais 16 operadores lógicos como “e”, “ou”, “não” etc. A mesma propriedade vale para o operador “NOU”. Por exemplo, “A e B” é equivalente a “( $A \text{ nou } A$ ) nou ( $B \text{ nou } B$ )”, “A ou B” é equivalente a “( $A \text{ nou } B$ ) nou ( $A \text{ nou } B$ )” etc.

Em notação formal, o NOU é representado por uma seta: “ $A \downarrow B$ ”, chamada de “seta de Peirce” em homenagem ao seu descobridor.

A característica especial desses operadores foi utilizada para a concepção das portas lógicas dos computadores, décadas mais tarde, o que possibilitou a construção de memórias e processadores.

### 3.16 A Calculadora Brasileira de Azevedo Coutinho – 1884

Há um relato curioso sobre um projeto de calculadora mecânica que teria sido patenteado no Brasil por Azevedo Coutinho (Portugal, 1860-1918) em 1884. O projeto era simples e parecia confiável, mas aparentemente nunca chegou a ser construído.

A máquina era capaz de executar várias somas seguidas e tinha um botão para zerar os mostradores. A [Figura 3.19](#) constava no registro de patente. Ela permite ver que a máquina tinha botões (a) para entrar com os algarismos, um de cada vez. Cada um destes botões acionava uma engrenagem incompleta com um número de dentes igual ao algarismo que estava sendo pressionado. Esse movimento era transmitido por engrenagens (d) ao mostrador adequado (i), que era definido pelo movimento horizontal do eixo (g), que engrenava apenas um mostrador de cada vez.

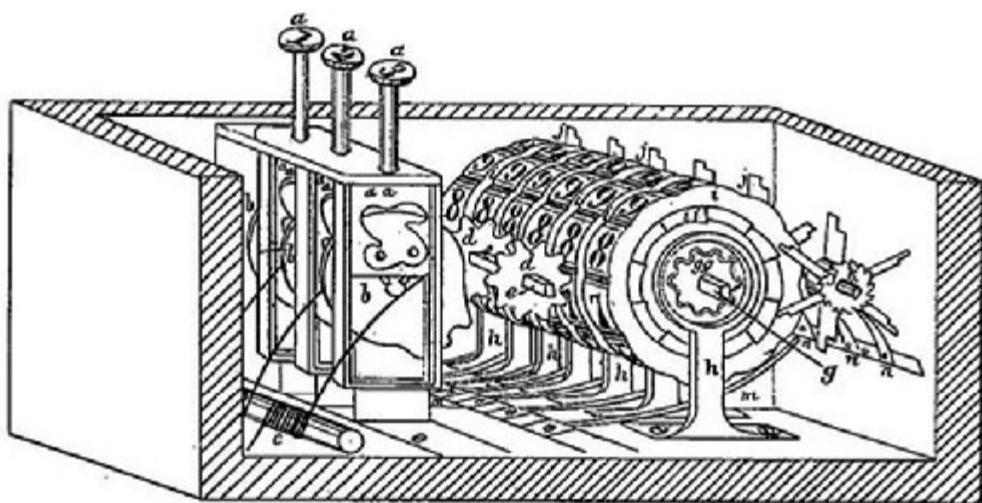


Figura 3.19: Máquina de Somar de Azevedo Coutinho.<sup>24</sup>

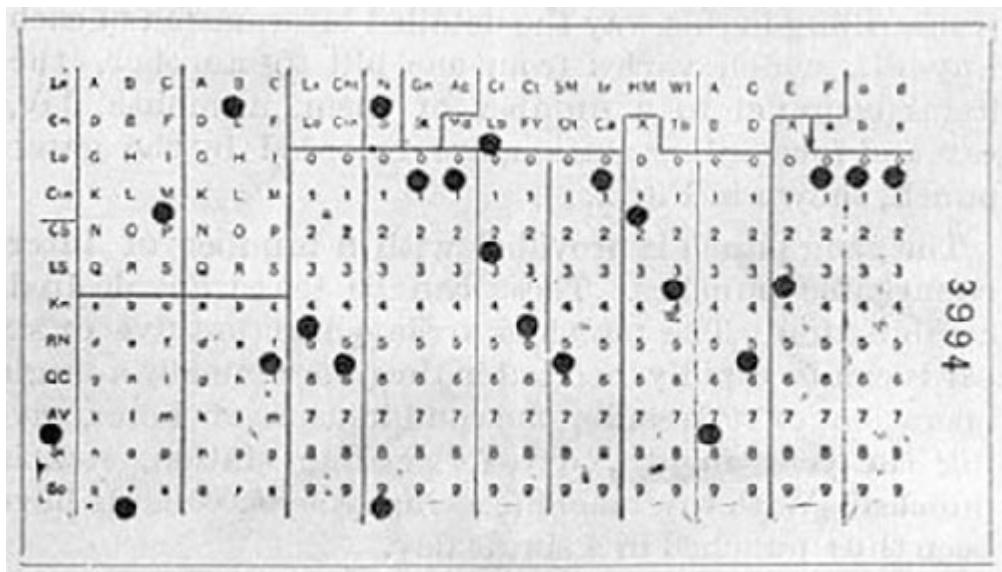
Um sistema de molas (c) fazia os botões retornarem à posição inicial e o *carry* era efetuado pelos dentes de engrenagens (j) acopladas aos mostradores.

Assim, para entrar com um número como 456, coloca-se o eixo (g) na posição das centenas e aperta-se o botão 4, depois na posição das dezenas e aperta-se o botão 5, finalmente coloca-se na posição das unidades e aperta-se o botão 6. Se a máquina não for zerada, o próximo número inserido será somado ao anterior. Assim, vários números em sequência podem ser inseridos e a máquina vai acumulando o resultado.

### 3.17 Máquina Tabuladora de Hollerith – 1884

Em 1879, Herman Hollerith (Estados Unidos, 1860-1929) após se graduar na Escola de Minas da Universidade de Columbia arrumou emprego no escritório do censo americano. O censo de 1880 estava prestes a ser iniciado e embora os dados tenham sido coletados em poucos meses, sua análise demandaria vários anos e, de fato, levou oito para ser concluída. Como a população americana crescia nessa época aos milhões devido principalmente às várias ondas de imigração, o fato era que os dados, quando finalmente compilados, já estariam bastante desatualizados. De fato, esperava-se que os resultados do censo pudessem pelo menos ser divulgados antes do censo seguinte, que aconteceria em 1890. Além do mais, como a população iria crescer bastante nesta década, já se imaginava que o censo de 1890 levaria ainda mais tempo para ser compilado.

Hollerith criou um projeto para realizar a compilação do censo de 1890 que foi um grande marco para a história da computação. Ele propôs que as informações coletadas fossem registradas em cartões de papel perfurado nos quais determinadas posições representariam diferentes respostas para as várias questões. A [Figura 3.20](#) mostra um destes cartões. Certamente Hollerith teria se inspirado no sistema de cartões dos teares de Jacquard, já em uso havia quase um século. Ele se decidiu pelo sistema de cartões após algumas tentativas frustradas com fitas de papel. O cartão de Hollerith tinha 24 colunas e 12 linhas.



**Figura 3.20:** Um dos cartões perfurados de Hollerith usados no censo de 1890.<sup>25</sup>

Assim, com todas as respostas ao censo representadas por meio de cartões perfurados, Hollerith propôs uma máquina que poderia ler e totalizar todas as respostas de uma única vez. Sua máquina tinha uma matriz de 12x24 agulhas metálicas. Elas eram baixadas todas de uma vez sobre cada cartão e apenas aquelas que atingissem os furos fariam contato com um recipiente cheio de mercúrio (um metal líquido) colocado abaixo do cartão. Cada agulha que fizesse esse contato ativava um circuito elétrico baseado em relês eletromecânicos que fazia um ponteiro avançar uma posição em um mostrador.

Os mostradores podiam ser associados a um furo específico ou até a uma combinação de furos. Cada ponteiro marcava 100 posições e como havia um ponteiro maior e um menor, como em um relógio, cada mostrador era capaz de contar até 10 mil. A [Figura 3.21](#) mostra um detalhe dessa máquina. Em primeiro plano aparece o local onde os cartões são colocados, um a um, e a matriz de agulhas. Em segundo plano aparecem os mostradores acionados por cada agulha, os quais efetivam a contagem de cada um dos furos.



**Figura 3.21: Detalhe da máquina tabuladora de Hollerith.**<sup>26</sup>

O uso dessa máquina fez com que o censo de 1890 fosse calculado de forma bem mais rápida do que teria sido caso fosse feito manualmente. A literatura diverge quanto ao tempo exato, mas o processo parece ter levado de seis meses a cerca de dois anos, no máximo.

Hollerith acabou até recebendo um doutorado pela Universidade de Columbia em 1890 em função da invenção desta máquina. Esse foi o primeiro sistema de processamento de informação a substituir com sucesso o papel e caneta.

Outras máquinas semelhantes foram construídas e usadas no processamento do censo em vários outros países, e também no censo americano de 1900.

Em 1896, Hollerith fundou a TMC (Tabulating Machine Company), considerada antecessora direta da IBM. No mesmo ano, ele criou uma máquina capaz de somar números codificados nos cartões, não apenas contar os furos.

Além disso, para o censo de 1900, como a quantidade de informação crescia muito, ele produziu um mecanismo para fazer automaticamente a alimentação de cartões na tabuladora, tornando o processo ainda mais rápido. Em 1901, ele também criou uma máquina perfuradora de cartões, a qual possuía um teclado numérico e traduzia as cifras digitadas pelos operadores em furos nos locais apropriados.

Em 1911, a TMC fundiu-se a outras empresas e assim surgiu a CTR (Computing-Tabulating-Recording Company). Essa empresa mudou de nome em 1924 para IBM (International Business Machines Corporation).

### **3.18 Até Aqui...**

Vimos que durante o século XIX vários avanços e tecnologias contribuíram para o futuro desenvolvimento da computação. Charles Babbage e Ada Lovelace conceberam a máquina analítica, capaz de efetuar qualquer computação 100 anos antes de que sua construção fosse possível. Suas publicações e ideias continuaram sendo citadas por pesquisadores ao longo deste século cada vez que uma nova contribuição para a automatização do cálculo ou do raciocínio era proposta.

A lógica também teve uma importante evolução nesse período, não só pela criação da lógica de predicados de Frege quanto também pela contribuição de Peirce com a descoberta de que um único operador lógico (um NE ou um NOU) pode ser usado para simular quaisquer dos demais operadores.

A criação do relê eletromagnético, e outras evoluções na área de eletricidade e eletrônica, permitiram que Hollerith produzisse uma máquina capaz de agilizar o processamento do censo. Essa máquina, o tabulador, não era mais do que uma calculadora eletromecânica; na verdade, sua primeira versão não passava de um simples contador. Contudo, os conceitos da tabuladora e da máquina analítica de Babbage puderam ser combinados no início do século XX, dando origem aos modernos computadores eletrônicos, como veremos a seguir.

---

<sup>1</sup> “Difference engine” by Canticle at English Wikipedia. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Difference\\_engine.JPG#/media/File:Difference\\_engine.JPG](https://commons.wikimedia.org/wiki/File:Difference_engine.JPG#/media/File:Difference_engine.JPG)

<sup>2</sup> “Linear Homeoscope” by Semen Korsakov - Korsakov, Semen N. *Apercu d'un procede nouveau d'investigation au moyen de machines a comparer les idees* - St. Petersbourg, 1832 (downloaded from Russian WP and converted to PNG). Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Linear\\_Homeoscope.png#/media/File:Linear\\_Homeoscope.png](https://commons.wikimedia.org/wiki/File:Linear_Homeoscope.png#/media/File:Linear_Homeoscope.png)

<sup>3</sup> By cs:Wikipedista:Jamottl, corrected by Jx - cs:Image:Schéma relé.PNG, Public Domain. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=4823359>

<sup>4</sup> By Unknown engraver. Disponível em: [http://www.academia.edu/1853185/Premises\\_for\\_exhibition\\_and\\_use\\_Kings\\_College\\_London\\_Museum\\_mid\\_to\\_late\\_19th\\_century](http://www.academia.edu/1853185/Premises_for_exhibition_and_use_Kings_College_London_Museum_mid_to_late_19th_century), Licenced under Public Domain

*via Wikimedia Commons.* <https://commons.wikimedia.org/w/index.php?curid=25667658>

<sup>5</sup> Nota A ao manuscrito de Menabrea. Tradução: “O mecanismo operacional pode mesmo ser colocado em ação independentemente de qualquer objeto para operar (embora, certamente, nenhum resultado seja então obtido).”

<sup>6</sup> Nota A ao manuscrito de Menabrea. Tradução: “Suponha, por exemplo, que as relações fundamentais entre as tonalidades na ciência da harmonia e a composição musical fossem suscetíveis a tais expressões e adaptações; a máquina poderia compor peças musicais elaboradas e científicas em qualquer grau de complexidade ou duração.”

<sup>7</sup> Imagem de domínio público. Fonte: Menabrea, L. F. (1842). *Sketch of the Analytical Engine Invented by Charles Babbage*. Bibliothèque Universelle de Genève, October, No. 82. Disponível em: <http://www.fourmilab.ch/babbage/sketch.html>

<sup>8</sup> Nota D ao manuscrito de Menabrea. Tradução: “Em quase toda computação, uma grande variedade de arranjos para a sucessão de processos é possível, e várias considerações devem influenciar a seleção entre eles para os propósitos da Máquina Calculadora. Um elemento essencial é escolher o arranjo que tenderá a reduzir a um mínimo o tempo necessário para completar o cálculo.”

<sup>9</sup> Nota E ao manuscrito de Menabrea. Tradução: “Um ciclo de operações então deve ser entendido como significando qualquer conjunto de operações que é repetido mais do que uma vez.”

<sup>10</sup> “Diagram for the computation of Bernoulli numbers” by Ada Lovelace. Disponível em: <http://www.sophiararebooks.com/pictures/3544a.jpg>. *Licenced under Public Domain via Wikimedia Commons.* [https://commons.wikimedia.org/wiki/File:Diagram\\_for\\_the\\_computation\\_of\\_Bernoulli\\_numbers.jpg#](https://commons.wikimedia.org/wiki/File:Diagram_for_the_computation_of_Bernoulli_numbers.jpg#/) [media/File:Diagram\\_for\\_the\\_computation\\_of\\_Bernoulli\\_numbers.jpg](media/File:Diagram_for_the_computation_of_Bernoulli_numbers.jpg)

<sup>11</sup> Tradução nossa da Sessão 73 do livro de Smee.

<sup>12</sup> Domínio público. Cortesia de Georgi Dalakov. Disponível em: [http://history-computer.com/Library/Smee\\_thought\\_adapted.pdf](http://history-computer.com/Library/Smee_thought_adapted.pdf)

<sup>13</sup> Domínio público. Cortesia de Georgi Dalakov. Disponível em: [http://history-computer.com/Library/Smee\\_thought\\_adapted.pdf](http://history-computer.com/Library/Smee_thought_adapted.pdf)

<sup>14</sup> Domínio público. Disponível em: <http://motherboard.vice.com/blog/alexander-graham-bell-did-not-invent-the-telephone>

<sup>15</sup> Uma versão mais aprimorada de um trabalho publicado em 1847, denominado “Mathematical Analysis of Logic”.

<sup>16</sup> © 2009 by Auction Team Breker, Koeln, Germany. Cortesia. Disponível em: [www.breker.com](http://www.breker.com).

<sup>17</sup> © 2009 by Auction Team Breker, Koeln, Germany. Cortesia. Disponível em: [www.breker.com](http://www.breker.com).

<sup>18</sup> Tradução nossa.

<sup>19</sup> “Writing ball keyboard 3”. Licensed under Public Domain via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Writing\\_ball\\_keyboard\\_3.jpg#/media/File:Writing\\_ball\\_keyboard\\_3.jpg](https://commons.wikimedia.org/wiki/File:Writing_ball_keyboard_3.jpg#/media/File:Writing_ball_keyboard_3.jpg)

<sup>20</sup> “Sholesglidden2” by Unknown - Smith, Clarence Charles (1922). *The Expert Typist*. New York: The Macmillan Company, p. 4 Melville, Arthur (1923). “The Machine Gun of Commerce” *The Rotarian (Rotary International)* 23, p. 18. Licenced under Public Domain via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:Sholesglidden2.png#/media/File:Sholesglidden2.png>

<sup>21</sup> “William Stanley Jevons Logic Piano” by Original uploader was Nelson at en.wikipedia - Transferred from en.wikipedia; transferred to Commons by User:Shizhao using CommonsHelper.. Licensed under CC BY-SA 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:William\\_Stanley\\_Jevons\\_Logic\\_Piano.jpg#/media/File:William\\_Stanley\\_Jevons\\_Logic\\_Piano.jpg](https://commons.wikimedia.org/wiki/File:William_Stanley_Jevons_Logic_Piano.jpg#/media/File:William_Stanley_Jevons_Logic_Piano.jpg)

<sup>22</sup> Domínio público. Fonte: Dalakov, G. *History of Computers, Hardware, Software, Internet...* Disponível em: <http://history-computer.com/MechanicalCalculators/19thCentury/Barbour.html> (Consultado em 01/09/2015).

<sup>23</sup> “Begriffsschriftnotation”. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:Begriffsschriftnotation.png#/media/File:Begriffsschriftnotation.png>

<sup>24</sup> Domínio Público. Fonte: Dalakov, G. *History of Computers, Hardware, Software, Internet...* Disponível em: <http://history-computer.com/CalculatingTools/Gadgets/Azevedo.html>

<sup>25</sup> “Hollerith punched card” by Herman Hollerith - Railroad Gazette. Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Hollerith\\_punched\\_card.jpg#/media/File:Hollerith\\_punched\\_card.jpg](https://commons.wikimedia.org/wiki/File:Hollerith_punched_card.jpg#/media/File:Hollerith_punched_card.jpg)

<sup>26</sup> “Hollerith card reader closeup” by Marcin Wichary - Flickr: Hollerith census machine. Licenced under CC BY 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Hollerith\\_card\\_reader\\_closeup.jpg#/media/File:Hollerith\\_card\\_reader\\_closeup.jpg](https://commons.wikimedia.org/wiki/File:Hollerith_card_reader_closeup.jpg#/media/File:Hollerith_card_reader_closeup.jpg)

## **PARTE IV**

### **Surgimento da Eletrônica**

O período entre o final do século XIX e início do século XX foi uma era em que os negócios cresceram. Bancos, indústria, comércio e governos tornaram-se cada vez maiores e mais complexos e por isso passaram a necessitar de mais armazenamento e processamento de informação. Não havia computadores propriamente ditos, mas os ecos dos sonhos de Babbage por uma máquina analítica universal ainda podiam ser ouvidos.

Com o surgimento da eletrônica, a existência efetiva dessas máquinas tornava-se progressivamente mais provável. O final do século XIX testemunhou os primeiros avanços na área de registro de informação por meio magnético, bem como a transmissão de voz e dados a distância através de fio e também de sinais eletromagnéticos pela atmosfera. Cartões perfurados começaram a ser usados com cada vez mais frequência para armazenar dados de todos os tipos por várias empresas, a maioria das quais clientes da CTR-IBM.

#### **4.1 Burroughs – 1886**

William Seward Burroughs I (Estados Unidos, 1857-1898) foi um inventor americano que deu origem a uma das principais empresas de computação do século XX. Conta-se que desde a infância ele se interessava por projetos de calculadoras. Acabou trabalhando em um banco e percebeu que as máquinas da época eram muito pouco

confiáveis, especialmente se operadas por pessoas sem experiência com elas.

No início dos anos 1880, ele se mudou para St. Louis e começou a trabalhar na Boyer Machine Shop. Ali, ele passou a dedicar-se a um projeto de criar uma máquina calculadora mais resistente e confiável do que todas as que existiam.

Em 1885 ele submeteu sua primeira patente e logo no início de 1886 fundou com mais três sócios a American Arithmometer Company, que em 1904 tornou-se a Companhia de Máquinas de Adição Burroughs. Em 1953, ela se torna a Burroughs Corporation, até que finalmente, em 1986, funde-se com a Sperry Corporation para formar a Unisys.

Em 1888 ele recebeu a patente pela invenção de uma máquina ([Figura 4.1](#)) muito semelhante às máquinas registradoras mecânicas que durante quase todo o século XX foram utilizadas em mercados pelo mundo inteiro.

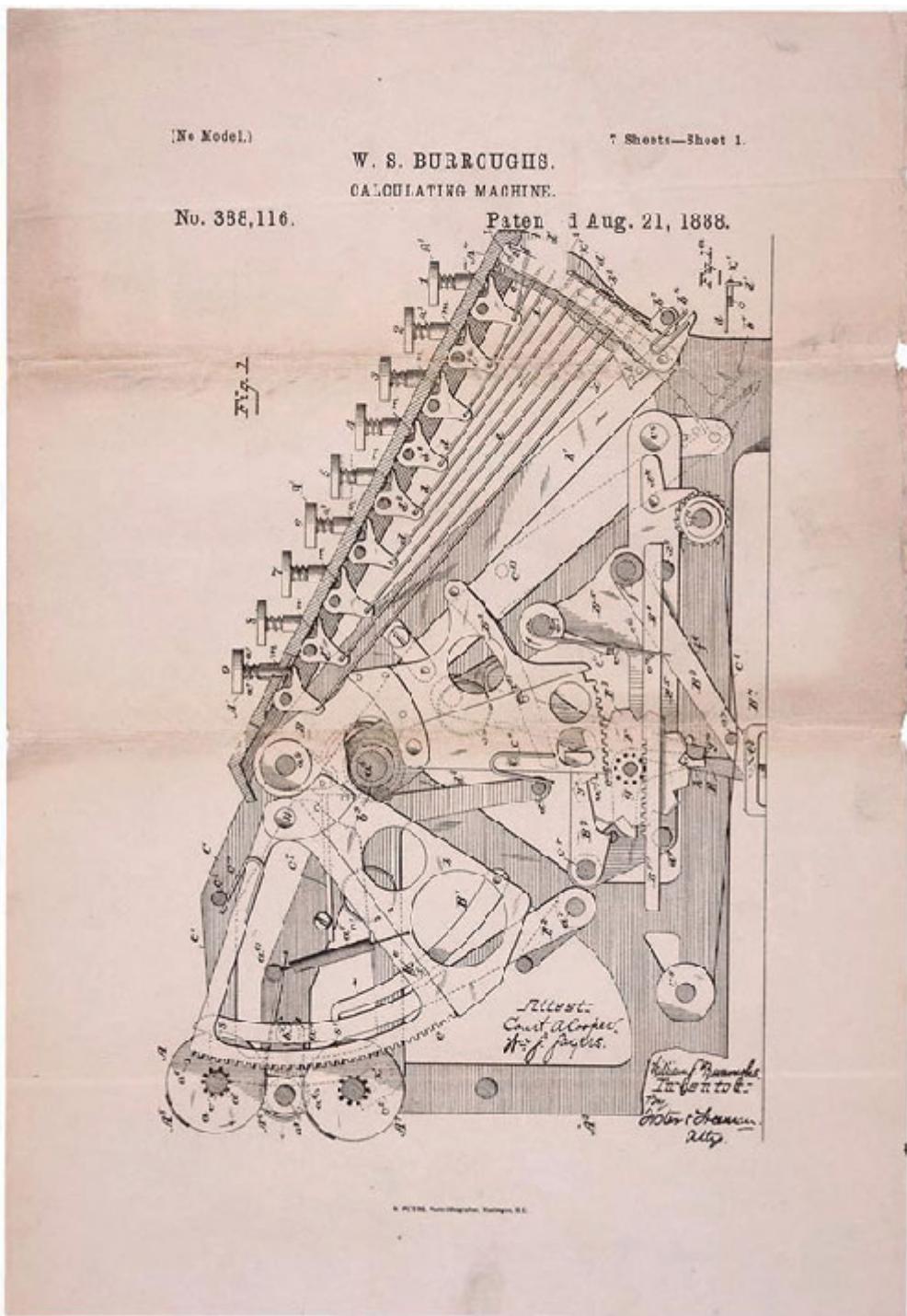


Figura 4.1: Máquina de calcular de Burroughs (1891).<sup>1</sup>

Conta-se que ele teria produzido cerca de 50 cópias de um primeiro modelo, mas que ao perceber que não eram tão confiáveis quanto

desejava, jogou-as, uma por uma, do segundo andar do prédio onde estavam.

## **4.2 Calculadora Eletromagnética de Charles Weiss – 1886**

A primeira calculadora eletromagnética de que se tem notícia é uma máquina patenteada por Charles Weiss em 1886. A única informação que se tem sobre ela é um registro de patente disponível em <http://www.google.com/patents/US348437>. Fora isso, quase nada se sabe sobre a eventual repercussão desta máquina ou mesmo sobre a vida de seu inventor. Consta no site mencionado acima que essa patente foi citada em pelo menos três outras, bem mais tarde, em 1940, 1947 e 1951.

Era, de fato, uma máquina de somar de coluna única, semelhante à de Caroline Winter, ou seja, ela é capaz de somar um dígito de 1 a 9 de cada vez e acumular o resultado em um registrador. Com o movimento de uma alavanca, visível no alto da [Figura 4.2](#), era possível determinar se estava somando unidades, dezenas ou centenas.

(No Model.)

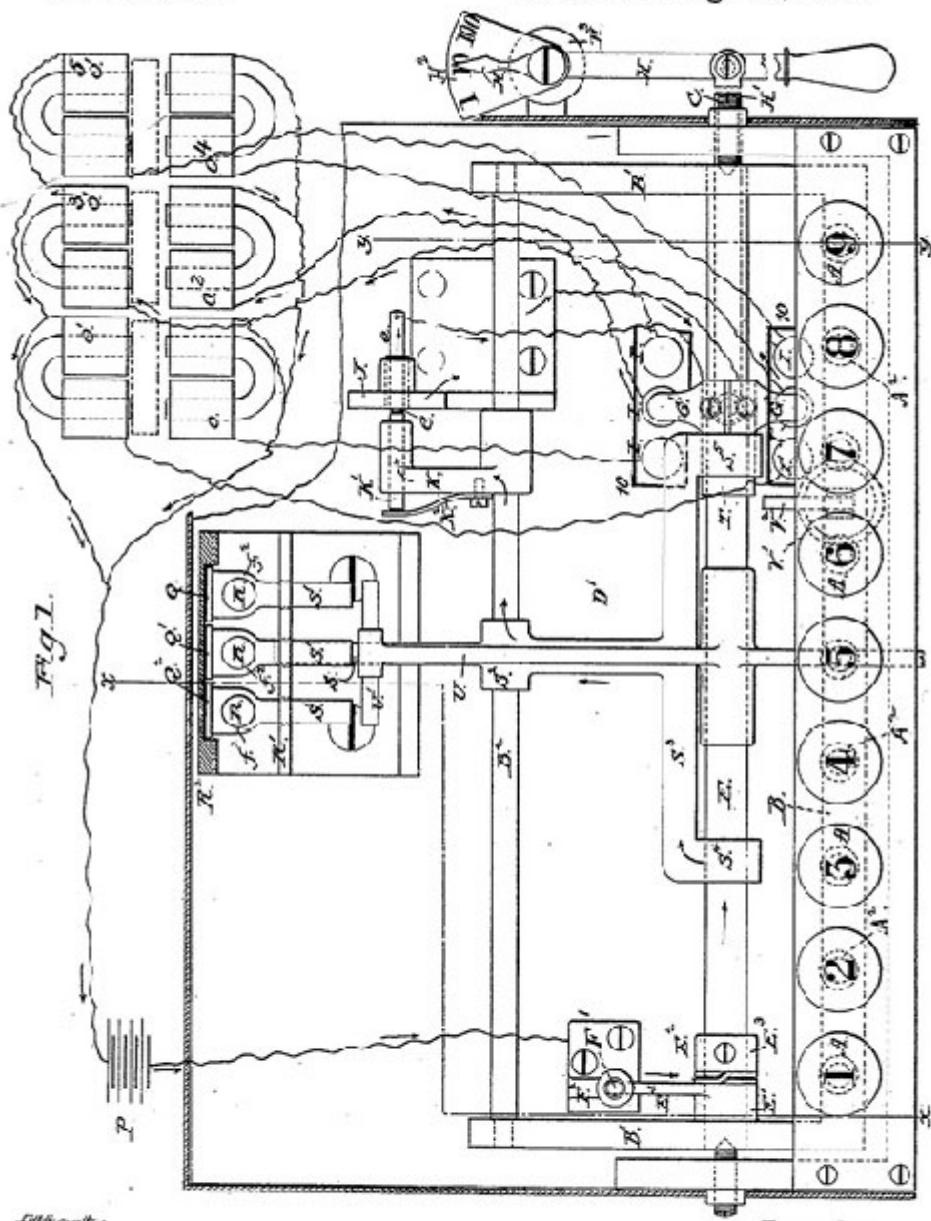
6 Sheets—Sheet 1.

C. W. WEISS,

ELECTRO MAGNETIC ADDING MACHINE.

No. 348,437.

Patented Aug. 31, 1886.



Attest:

John C. Ellis  
ABM: m.

Charles W. Weiss

By David A. Brown

Atty.

N. PITEL, Photo-Lithographer, Washington, D. C.

**Figura 4.2: Calculadora eletromecânica de Weiss.<sup>2</sup>**

Convém mencionar que Hollerith já havia desenvolvido uma máquina tabuladora eletromagnética pelo menos dois anos antes da patente de Weiss. Entretanto, a invenção de Hollerith somava de um e um, ou seja, era uma máquina de incremento unitário que correspondia a uma contadora de furos em cartões. A máquina de Hollerith não seria capaz de calcular  $5+6+9+7+4\dots$ . Já a máquina de Weiss tinha um teclado com nove teclas, cada qual marcada com um algarismo. Quando uma delas era pressionada, um dispositivo eletromagnético fazia girar uma roda no ângulo que correspondia à quantidade de movimento associada ao algarismo, acumulando assim os valores à medida que iam sendo digitados.

### 4.3 Máquina Lógica de Allan Marquand – 1886

Allan Marquand (Estados Unidos, 1853-1924) era professor da Universidade de Princeton. Ele obteve um PhD pela Universidade Johns Hopkins, sob orientação de Charles S. Peirce. Logo após seu doutoramento, por sugestão de Peirce, concebeu uma máquina lógica muito semelhante ao piano lógico de Jevons. Porém, ela possuía execução muito mais simples, pois operava com circuitos elétricos.

A máquina básica construída por Marquand também operava com quatro variáveis lógicas: A, B, C, D, e suas negações a, b, c e d. Basicamente ela permitia visualizar uma matriz com as 16 combinações possíveis, mostrada na [Tabela 4.1](#).

**Tabela 4.1: Combinações de quatro variáveis lógicas**

		A		a	
		B	b	B	b
C	D	ABCD	AbCD	aBCD	abCD
	d	ABCd	AbCd	aBCd	abCd
c	D	ABcD	AbcD	aBcD	abcD
	d	ABcd	Abcd	aBcd	abcd

Essa disposição matricial da [Tabela 4.1](#) pode ser visualizada no diagrama da máquina exibido na [Figura 4.3](#). Na mesma figura, observam-se as teclas que acionam os circuitos elétricos que respectivamente acionam as bobinas magnéticas, cada qual representando uma das posições da tabela de combinações. Essas bobinas eram responsáveis por alterar a exibição dos mostradores da máquina de forma que o usuário pudesse saber quais das opções eram consistentes ou não com as expressões lógicas informadas.

Da mesma forma que o piano de Jevons, essa máquina permitia a entrada de premissas como  $A \rightarrow B$  e  $B \rightarrow C$ , e à medida que iam sendo informadas, ela eliminaria as opções na matriz que fossem incompatíveis com tais expressões lógicas.

Novamente, o equipamento era incapaz de apresentar inferências diretas, como “se  $A \rightarrow B$  e  $B \rightarrow C$  então  $A \rightarrow C$ ”, já que a expressão resultante deveria ser inferida a partir das opções consistentes na matriz apresentadas pela máquina. Entretanto, foi um passo importante na direção de mostrar que máquinas eletrônicas poderiam ser usadas para fazer aquilo que antes era realizado apenas por máquinas mecânicas, especialmente no que se refere ao tratamento matemático e lógico. Este foi, portanto, o primeiro projeto de uma máquina lógica eletrônica.

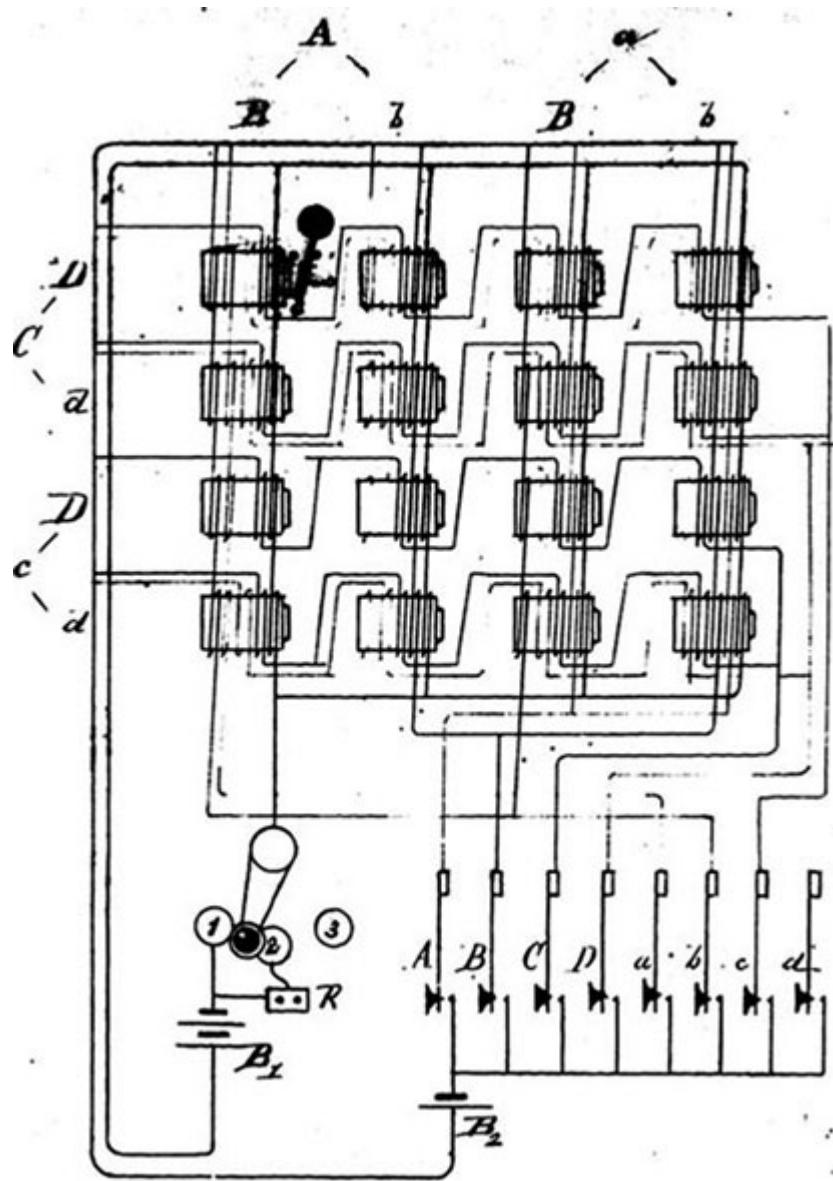


Figura 4.3: Diagrama da máquina lógica de Marquand.<sup>3</sup>

#### 4.4 Telautograph – 1888

Em 1888, um inventor chamado Elisha Gray (Estados Unidos, 1835-1901) inventou um mecanismo que é considerado precursor dos aparelhos de fax. O aparelho, denominado Teleautograph, consistia basicamente de uma caneta que ficava presa a dois fios, cada um dos quais ligados a um potenciômetro ([Figura 4.4](#)). Quando a caneta

é movida, os fios fazem girar os potenciômetros. Sinais elétricos que passam pelos dois potenciômetros são enviados então a uma estação de recepção, onde servomecanismos (motores) reproduzem o movimento da caneta.

Desta maneira, o aparelho conseguia de fato transmitir escrita e desenhos feitos à mão a distância, desde que duas linhas de transmissão estivessem disponíveis.

No final do século XIX ele foi modificado por Foster Ritchie para que funcionasse simultaneamente como telefone. Esse novo aparelho foi rebatizado como Telewriter.

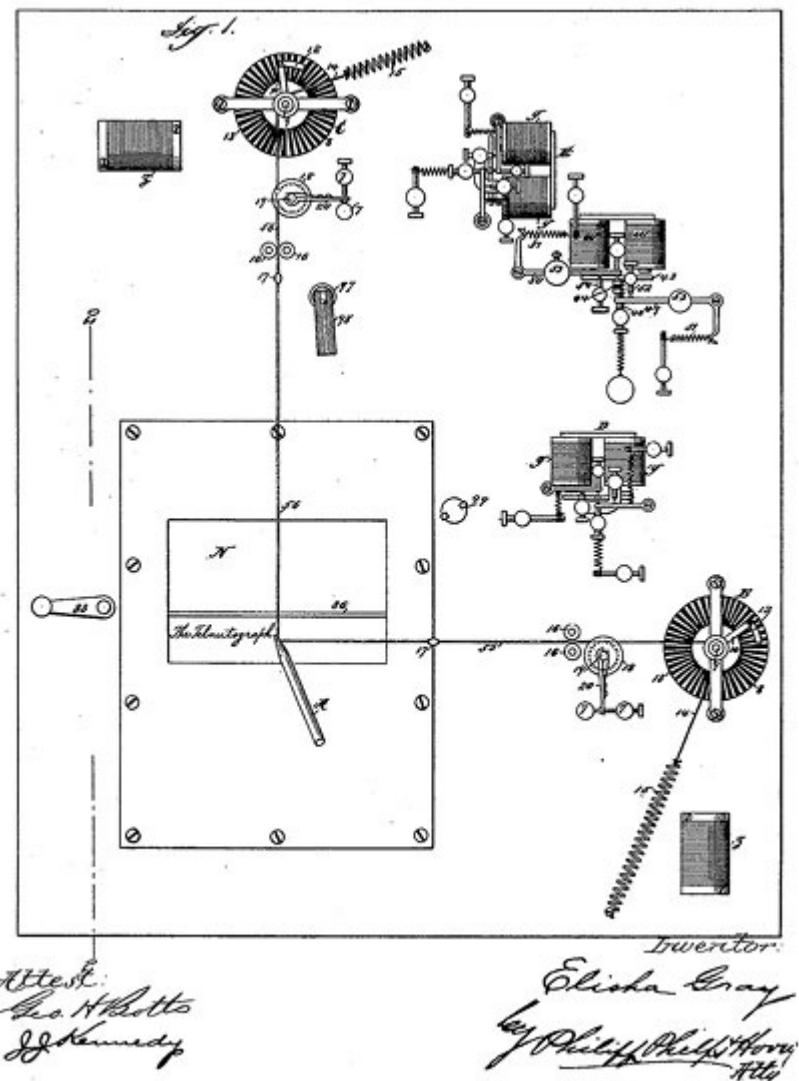


Figura 4.4: Teleautograph.<sup>4</sup>

O Teleautograph foi popular na época, sendo instalado em bancos e hospitais, especialmente para obtenção de assinaturas de clientes e médicos a distância. Estações ferroviárias também utilizavam o aparelho para passar informações sobre os horários dos trens e possíveis atrasos – inclusive, o sistema ficou operacional na estação de Chicago até a década de 1970. Gray chegou a receber mais de 5 milhões de dólares pelas suas patentes ao longo da vida.

A empresa Teleautograph Corp., que fabricava o equipamento, mudou de nome várias vezes durante sua história. Ela finalmente foi comprada pela Xerox em 1999 e transformou-se na divisão Omnifax.

## 4.5 Gravador Magnético de Oberlin Smith – 1888

Oberlin Smith (Estados Unidos, 1840-1926) era um engenheiro mecânico bastante famoso no final do século XIX. Por volta de 1880, ele visitou os laboratórios de Thomas Edison (Estados Unidos, 1847-1931), que tinha acabado de inventar o fonógrafo e o microfone. O fonógrafo de Edison permitia gravar sons através de sulcos em materiais, como folhas metálicas, de forma semelhante aos antigos LPs (discos plásticos também conhecidos como “*long play*”, ainda hoje usados).

Sendo um apaixonado por música, Oberlin imediatamente comprou um fonógrafo, mas não ficou satisfeito com a qualidade do som, pois chiava muito. Desta maneira, Oberlin colocou sua criatividade para funcionar e conseguiu imaginar uma forma totalmente diferente de gravar sons. Como as músicas captadas pelo microfone eram transformadas em corrente elétrica, ele imaginou que um fio ou fita metálica magnetizada também seriam capazes de armazenar sequências de sons.

Ele corretamente presumiu que um fio metálico não seria o meio adequado, pois o magnetismo induzido se espalharia por toda a sua extensão, não ficando registrada a sequência de gravação. Para evitar esse problema, propôs que um fio de tecido fosse “embebido” em pó de ferro de forma que as partículas não se tocassesem, ficando isoladas umas das outras, para evitar a propagação da indução magnética. Assim, ao passar o fio sobre um indutor magnético ligado, por exemplo, a um microfone, as correntes induzidas fariam cada uma das partículas ficar magnetizada de um jeito diferente – ou seja, o som estaria registrado no fio e poderia ser reproduzido

quando ele fosse passado novamente sobre uma bobina capaz de captar o magnetismo e transformá-lo em corrente elétrica outra vez.

A [Figura 4.5](#) mostra o desenho de uma pré-patente que Oberlin registrou em 1878. O fio embebido em pó de ferro (C) seria desenrolado e enrolado através dos cilindros D e E. Um microfone (A), captando sons, produziria corrente elétrica no fio B que, enrolado em forma de bobina sobre o fio C, induziria magneticamente as partículas de ferro neste outro fio. Assim, todos os sons captados pelo microfone ficariam ali representados. Ao passar o mesmo fio sobre uma bobina ligada a um autofalante, os sons poderiam ser ouvidos.

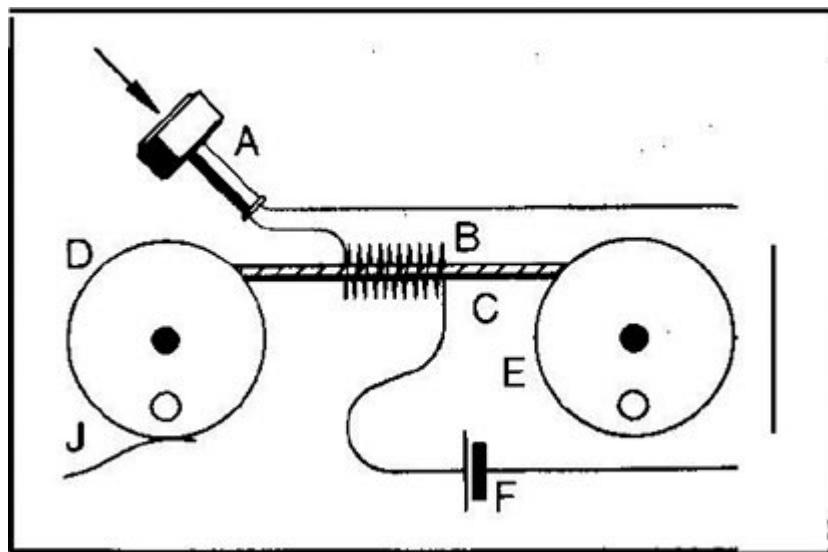


Figura 4.5: O primeiro projeto de um gravador magnético.<sup>5</sup>

Nasceu assim o primeiro dispositivo de armazenamento magnético de que se tem notícia, e isso em pleno século XIX. Até hoje os dispositivos de armazenamento magnético seguem o mesmo princípio, com a ressalva de que temos materiais muito mais eficientes do que tecido e limalha de ferro atualmente. Porém, Smith, ocupado com outras invenções e devido à grande dificuldade tecnológica de transformar seu invento em algo útil, acabou

desistindo de patenteá-lo e preferiu compartilhar a ideia ao publicá-la na revista *Electrical World* em 1888.

## 4.6 Calculadoras Analógicas de Leonardo Torres y Quevedo – 1893

Leonardo Torres y Quevedo (Espanha, 1852-1936) foi provavelmente um dos maiores cientistas da virada do século XIX para o XX. Ainda falaremos de outras invenções dele, mas no momento o assunto será sua contribuição na área de computação analógica.

Um computador analógico representa dados e produz soluções para problemas utilizando grandezas físicas em vez de representação simbólica. Pode-se dizer também que computadores analógicos operam com variáveis contínuas, enquanto os digitais operam com variáveis discretas.

Essas grandezas físicas podem ser de várias naturezas; por exemplo, uma distância percorrida (como no caso das régulas de cálculo), o girar de um eixo, a frequência de uma vibração, uma tensão elétrica, um volume ou um peso. Por exemplo, quer saber quanto é  $3,75 \times 3$ ? Consiga três objetos que pesem exatamente 3,75 kg e ponha-os juntos na balança. O resultado será  $3,75 \times 3$ , ou seja, 11,25 kg. Você acabou de fazer uma computação analógica.

Um computador analógico é, assim, uma máquina que resolve um problema matemático através de uma representação física que, processada, apresenta um resultado físico que é depois reinterpretado como valor matemático.

Torres y Quevedo trabalhou neste primeiro período basicamente com máquinas mecânicas, que ele chamou de “calculadoras algébricas” em uma primeira publicação em 1893, totalmente inspirada no trabalho de Babbage. Não só ele, como outros na época, mostram que Babbage não foi esquecido mesmo décadas depois de sua frustrada tentativa de construir um computador mecânico universal.

Ele criou uma concepção de máquina puramente geométrica, o que era novidade até então, pois máquinas eram definidas sempre em função de tempos e movimentos. Torres eliminou a necessidade de usar o tempo para descrever uma máquina e propôs que uma descrição de relações geométricas seria suficiente para quaisquer delas.

Afirmava, então, que uma máquina é definida por um conjunto de relações entre suas partes móveis e que tais relações se expressam por certas fórmulas. Ele faz então uma pergunta chave: “¿Podrá construirse una fórmula cualquiera?”<sup>6</sup> Assim, cria todo um novo conceito científico: fórmulas matemáticas podem ser representadas fisicamente por máquinas? Ele passa então a conceber as mais variadas formas de máquinas que mecanicamente realizavam diversos tipos de cálculos.

Por um curto, mas intenso, período ele se desdobra em descrever vários tipos de máquinas, calculadoras de polinômios, integradores, revolvedores de equações diferenciais, e passa a construir alguns desses equipamentos, inclusive, provando que suas teorias eram praticáveis. A [Figura 4.6](#) mostra uma das máquinas algébricas, atualmente em exposição no Museu da Faculdade de Engenharia Civil, em Madrid.



Figura 4.6: Uma das máquinas algébricas de Torres y Quevedo.<sup>7</sup>

## 4.7 TMC: Tabulating Machine Company – 1896

Lembra de Herman Hollerith, do censo norte-americano de 1890? Conforme vimos, a partir do sucesso de sua máquina tabuladora, e de várias patentes concedidas, ele fundou em 1896 uma empresa chamada TMC (Tabulating Machine Company), a qual se especializou na construção e venda de serviços de tabulação e ordenação de informações representadas em cartões perfurados. Essa empresa, já sabemos, foi uma das que formou a CTR e posteriormente a IBM.

Por essa época, Hollerith estava fornecendo serviços para empresas de trens e preparava-se para o grande desafio do censo de 1900, o qual suas máquinas conseguiriam processar em tempo recorde. Infelizmente, só havia censos nos EUA a cada 10 anos e, nesse meio tempo, ele ocupava-se em buscar novos clientes e abrir novos mercados para suas máquinas tabuladoras.

Como Hollerith era o único detentor da tecnologia de máquinas tabuladoras, ele elevou sobremaneira o preço de seus serviços, cobrando do departamento do censo americano valores absurdos para a época. Assim, o censo acabou investindo no desenvolvimento de tecnologia própria, e já para a pesquisa de 1910 tinham máquinas melhores que as de Hollerith, deixando-o de fora do processo.

Seu negócio foi, por fim, comprado por um investidor em 1911 pela fantástica quantia de 2,3 milhões de dólares (uma fortuna ainda hoje, quanto mais naquela época), dos quais 1,2 milhões ficaram com Hollerith. Ele permaneceu como membro do conselho da nova empresa, até sua aposentadoria cerca de 10 anos depois.

## 4.8 O Monitor CRT – 1897

A história dos monitores CRT e dos televisores do século XX remonta ao final do século XIX. CRT é uma sigla que significa “Cathode Ray Tube” (Tubo de Raios Catódicos). Já em 1869, Johann Hittorf (Alemanha, 1824-1914) observou que raios desconhecidos eram emitidos a partir de um cátodo, um dispositivo que atua, neste caso, como um emissor de elétrons.

Em 1897, Karl Ferdinand Braun (Alemanha, 1850-1918) construiu um tubo de vidro, dentro do qual produziu vácuo e associou esse tubo em uma das pontas a um emissor de elétrons (um cátodo) e na outra ponta a uma tela fluorescente. Como resultado ele podia visualizar na tela um pequeno ponto brilhante que correspondia ao fino raio de elétrons. Na [Figura 4.7](#), o gerador de raios está à direita,

a tela à esquerda (dentro do tubo) e os indutores magnéticos na parte central da figura.

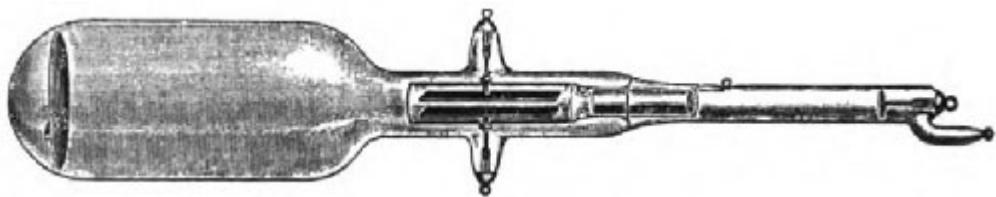


Figura 4.7: Tubo de raios catódicos de Braun.<sup>8</sup>

Ainda no mesmo ano, o cientista Joseph John Thompson (Reino Unido, 1856-1940) fez experiências que demonstraram a deflexão eletromagnética do feixe de elétrons, ou seja, um campo magnético poderia mudar a direção do raio. Alterando-se a potência do campo magnético podia-se fazer o raio desviar em maior ou menor grau para uma direção ou outra. O resultado na tela fluorescente era um ponto luminoso que ficava dançando de um lado para outro.

Agora imagine duas bobinas induzindo campos magnéticos, uma na horizontal e outra na vertical perpendicularmente ao movimento do raio de elétrons em direção à tela. Imagine que a bobina vertical mantém o feixe parado numa posição, enquanto a horizontal leva o feixe de elétrons de um lado ao outro da tela. Daí a bobina vertical faz o feixe descer uma posição discreta (uma linha) e a bobina horizontal repete o processo. Neste caso, você vai ver o feixe de elétrons passando por todas as posições da tela linha por linha. Se a velocidade com que as bobinas fazem suas mudanças for suficientemente rápida, você não verá mais o pontinho dançando na tela, mas enxergará a tela toda iluminada. Claro que isso é só uma ilusão, pois de fato apenas um ponto está iluminado a cada instante, mas seu olho é incapaz de ver isso porque o ponto muda sua posição muito rápido.

Finalmente, acrescente um circuito que aumenta ou diminui a intensidade do raio de elétrons, de acordo com uma programação

predefinida. Quando a intensidade do raio diminui, o ponto da tela que ele toca fica mais escuro, mas se a intensidade do raio aumenta, o ponto fica mais luminoso. Se você criar uma codificação adequada para esta sequência de altos e baixos de intensidade, provavelmente será capaz de deixar algumas regiões da sua tela mais escuras e outras mais claras. Fez tudo isso? Então você acaba de inventar a televisão; pelo menos a TV CRT, que usa o tubo de imagem.

Esse experimento para criação artificial de imagens em uma tela fluorescente a partir de um tubo de raios catódicos foi feito com sucesso pela primeira vez por Boris Rosing (Rússia, 1869-1933), que conseguiu mostrar figuras geométricas simples em uma tela de CRT. O primeiro televisor comercializado foi fabricado pela alemã Telefunken em 1934.

## 4.9 Telephone – 1898

Oberlin Smith concebeu, mas não chegou a construir seu gravador magnético. Essa honra coube a Valdemar Poulsen (Dinamarca, 1869-1942), que em 1898 construiu e patenteou a primeira máquina que realmente realizou gravações a serem posteriormente tocadas. Essa máquina chamava-se Telephone ([Figura 4.8](#)).

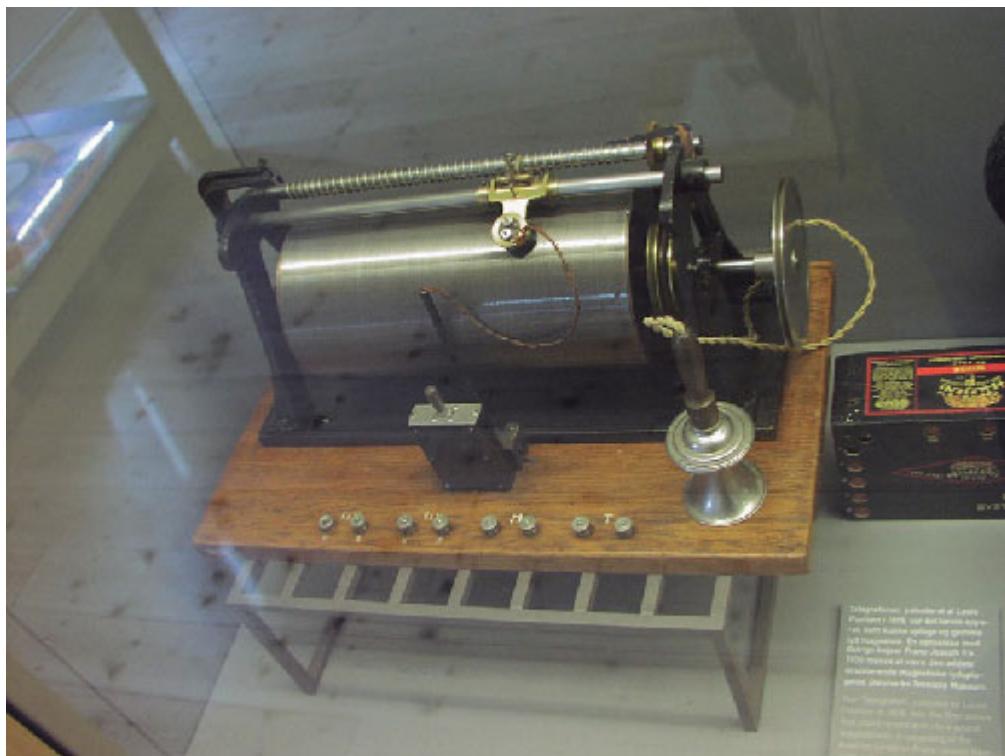


Figura 4.8: Telephone.<sup>9</sup>

Ela era bem parecida com a máquina de Smith. Basicamente possuía um cilindro sobre o qual estava enrolado um fio metálico que era magnetizado por uma espécie de “cabeçote” móvel, o qual a partir de uma corrente elétrica gerada por um microfone, induzia diferentes graus de magnetismo no fio. O fio enrolado no cilindro podia depois acionar um cabeçote de leitura (bobina) na qual seus diferentes níveis de magnetismo iriam induzir uma corrente variável que, se conectada a um autofalante resultaria em sons audíveis.

Na época não havia ainda amplificadores, assim o som ouvido era de baixíssima intensidade. Entretanto, com o fone bem encostado ao ouvido, como no caso de um telefone, ele poderia ser ouvido.

A máquina fez muito sucesso na Feira Mundial de Paris em 1900. Consta que a primeira gravação feita foi o som da voz do imperador Franz Josef da Áustria, a qual é considerada o mais antigo registro de som em meio magnético do mundo.

Mesmo assim, ele não conseguiu investimentos para produzir sua máquina em escala comercial. Em 1903, porém, com alguns sócios americanos fundou a American Telephone Company e conseguiu produzir e vender uma versão comercial do invento.

## 4.10 Autarith de Alexander Rechnitzer – 1900

A evolução das calculadoras mecânicas não parou, e em 1900 Alexander Rechnitzer (Eslováquia, c. 1880-1922) patenteou e construiu um novo modelo de calculadora mecânica capaz de fazer as quatro operações de forma automática, a Autarith.

A novidade que ele introduziu foi o uso de um motor elétrico para mover as engrenagens, semelhantes aos cilindros escalonados de Leibniz. Desta forma, evitava-se que o usuário tivesse que controlar por conta própria a quantidade de giros de alavanca necessários para efetuar multiplicações e divisões longas.

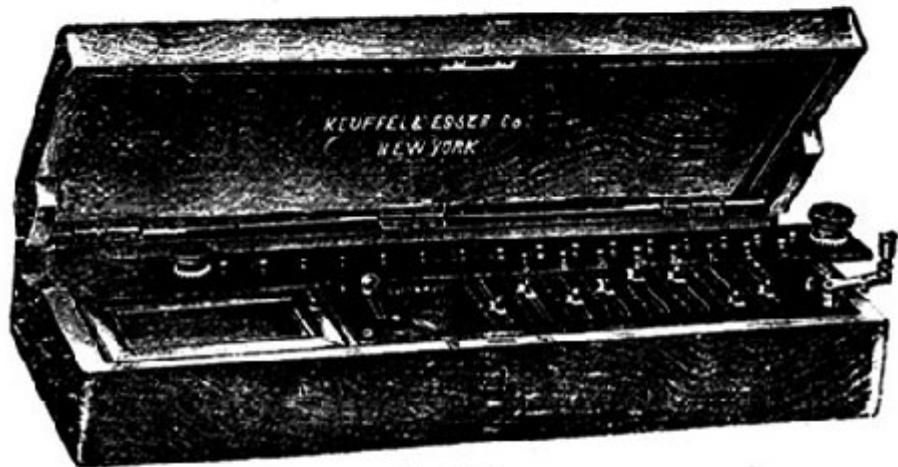
Desta maneira, a simples existência do motor permitiu simplificar bastante a operação da calculadora e foram projetados e introduzidos mecanismos de controle para sua parada automática.

Rechnitzer chegou a conseguir produzir suas máquinas em escala comercial, mas infelizmente não era bom vendedor e não obteve muito sucesso nas vendas. Após sua morte (seu corpo foi encontrado no East River em Nova York), suas ideias acabaram sendo exploradas por outras empresas.

A [Figura 4.9](#) mostra um anúncio do catálogo da Keuffel & Esser, importante fabricante de réguas de cálculo nos Estados Unidos desde 1867, oferecendo três diferentes modelos da Autarith.

KEUFFEL & ESSER CO. NEW YORK.

## RECKONING MACHINES.



No. 4006.

Calculations, such as Addition, Subtraction, Multiplication, Division, Squaring, Cubing, Extracting of Square Roots, etc., etc., can be performed with rapidity and unfailing accuracy and without mental effort by means of this machine. The tedious work of reckoning is reduced to setting the figure discs or shifters and turning the crank handle, a simple mechanical operation. The machine will multiply two factors, each of which may have as many figures as there are grooves in the machine.

The machine is mechanically perfect and built in the most substantial manner so that it will retain its efficiency and accuracy for a very long time. There are a great many of these machines in use in public and private offices and scientific laboratories here and in Europe, and they are in all cases giving the greatest satisfaction.

4005.	Reckoning Machine,	6	grooves,	12	holes	in upper row . .	each	\$ 193 25
4006.	do.	do.	8	"	16	" "	" . . "	241 50
4007.	do.	do.	10	"	20	" "	" . . "	389 25



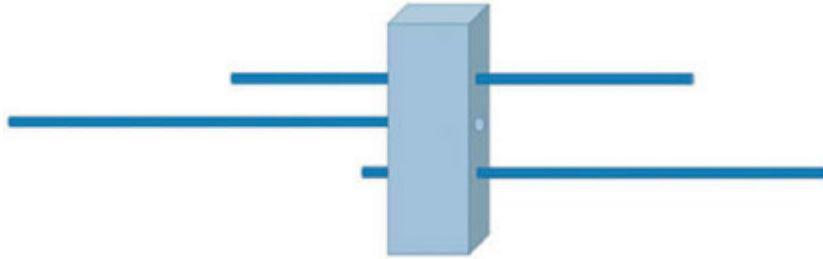
Figura 4.9: Anúncio da Autarith no catálogo da Keuffel & Esser.<sup>10</sup>

## 4.11 Máquina Analítica de Percy Ludgate – 1903

Percy Edwin Ludgate (Irlanda, 1883-1922) começou a trabalhar em 1903 em um hobby: projetar calculadoras. Mesmo sem nunca ter ouvido falar na máquina analítica de Babbage (pelo menos inicialmente), ele projetou uma máquina analítica de igual poder computacional, porém com um mecanismo muito mais simples.

Enquanto a máquina de Babbage representava seus registradores em torres de discos de alta complexidade, a máquina de Ludgate representaria os registradores (ou variáveis, como ele também chamava) em bastões metálicos que podiam assumir diferentes posições em relação a um suporte. Cada uma das posições correspondia a um algarismo de 0 a 9, e a quantidade de bastões colocados no suporte definiria o número de dígitos que cada variável poderia representar. Um dos bastões representaria o sinal do número (positivo ou negativo). Ludgate não chegou a nos deixar desenhos, mas a partir de suas descrições uma concepção do provável desenho da máquina foi feita em 1973. Os desenhos são apresentados em um artigo de Brian Randell disponível em <http://www.cs.ncl.ac.uk/research/pubs/articles/papers/398.pdf>.

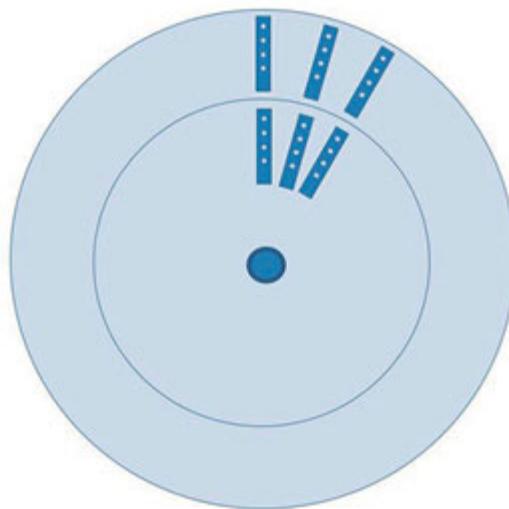
A [Figura 4.10](#) mostra um detalhe do suporte de memória com três barras. A do alto está na posição média, representando assim o número 5. A barra central está na posição mais recolhida, representando assim o 0 e a barra mais abaixada, praticamente toda estendida representa um 8. Este suporte, portanto, está representando o número 508.



**Figura 4.10:** Detalhe de uma barra de memória da máquina analítica de Ludgate.

Cada suporte no projeto original continha 20 barras, cada qual representando diferentes dígitos de um número entre 0 e 99.999.999.999.999.999.999 armazenado no respectivo registrador ou variável.

A memória da máquina seria formada por dois discos concêntricos, cada qual contendo 96 suportes. A [Figura 4.11](#) mostra de forma esquemática a colocação de três suportes em cada um dos dois discos (o interno e o externo). Cada suporte no desenho conta com quatro barras, neste caso podendo representar seis variáveis capazes de assumir qualquer valor entre 0 e 9.999.



**Figura 4.11:** Esquema dos discos de memória da máquina analítica de Ludgate.

Pode-se observar que os diferentes registradores são acondicionados em dois discos concêntricos com giro independente. No total, a máquina seria capaz de armazenar 192 variáveis. Para operar sobre duas variáveis, então, era necessário girar os dois discos até que suas posições coincidissem com a posição de operação, quando então seriam lidos pelo “moinho”. Ludgate usou o mesmo nome que Babbage utilizou para se referir ao que hoje chamamos de ULA, ou Unidade de Lógica e Aritmética.

Pode-se ver, assim, que o mecanismo de Ludgate era muito mais simples que o de Babbage e usava menos peças. Além disso, os discos poderiam ser removidos e substituídos facilmente, o que não era possível com as enormes torres de discos de Babbage.

Outra inovação de Ludgate era a maneira como sua máquina faria multiplicações. Essa técnica é hoje conhecida como “logaritmo irlandês”. Observe a [Tabela 4.2](#); ela contém um índice equivalente para cada dígito possível de 0 a 9. Para multiplicar dois dígitos, a ideia seria tomar os índices equivalentes a cada um dos dígitos, somá-los e encontrar em uma segunda tabela, na posição correspondente à soma destes índices, o valor do produto.

**Tabela 4.2: Índice correspondente para cada dígito decimal nos logaritmos irlandeses**

Dígito	0	1	2	3	4	5	6	7	8	9
Índice	50	0	1	7	2	23	8	33	3	14

Assim, por exemplo, para multiplicar 4 por 5, deveríamos somar os índices correspondentes a 4 e 5:  $2+23 = 25$ . Assim, na posição 25 da tabela de antilogaritmos encontrariamo o valor 20, que corresponde ao produto de 4 por 5. A [Tabela 4.3](#) apresenta o início da tabela de antilogaritmos irlandeses.

**Tabela 4.3: Parte da tabela para encontrar o antilogaritmo irlandês dado o índice**

Índice composto	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Valor do produto	1	2	4	8	16	32	64	3	6	12	24	48	-	-	49	18	36

							...						
Índice composto	17	18	19	20	21	22	23	24	25	26	27	28	29
Valor do produto	72				27	54	5	10	20	40		81	15
	-	-	-	-							-	-	

Observe então como os índices compostos são obtidos:

- 0 é a soma dos índices 0 e 0, ou seja,  $1 \times 1 = 1$ .
- 1 é a soma dos índices 0 e 1, ou seja,  $1 \times 2 = 2$ .
- 2 é a soma dos índices 1 e 1, ou seja,  $2 \times 2 = 4$ .
- 5 é a soma dos índices 2 e 3, ou seja,  $4 \times 8 = 64$ .
- 12 é indefinido porque não existem dois índices que possam somar 12.
- 17 é a soma dos índices 3 e 14, ou seja,  $8 \times 9 = 72$ .
- 25 é a soma dos índices 2 e 23, ou seja,  $4 \times 5 = 20$ .

E assim por diante.

A máquina de Ludgate seria controlada por uma fita perfurada, ou “papel fórmula”, como ele a chamava. Cada instrução na fita indicaria quais das duas variáveis seriam operadas, qual a operação a ser executada e em qual variável ou par de variáveis o resultado deveria ser armazenado.

Ele publicou um artigo sobre a máquina em 1909, no qual reconhecia o trabalho de Babbage e comparava o seu ao dele. Sua máquina também teria, embora não tivesse descrito, um comando para desvio condicional e repetição, possivelmente atuando na forma como a fita perfurada era movida. Mais impressionante de tudo era o fato de que a máquina de Babbage seria gigantesca e extremamente complexa, mas a de Ludgate poderia caber sobre uma mesa. Iniciava-se, assim, a miniaturização dos computadores.

## 4.12 Válvula-Diodo de John Fleming – 1904

A válvula foi um dispositivo muito importante na história da computação pois muitos dos primeiros computadores foram construídos com elas. É verdade que a válvula foi substituída pelo transistor e depois pelos circuitos integrados, mas estes são apenas uma execução física diferente do mesmo princípio: o dos semicondutores. Um semicondutor é um dispositivo eletrônico que conduz corrente elétrica apenas em uma direção e bloqueia-a na direção oposta.

O primeiro semicondutor de que se tem notícia foi construído pelo engenheiro John Ambrose Fleming (Reino Unido, 1849-1945) em 1904. Sua invenção, cujos primeiros protótipos de 1904 são mostrados na [Figura 4.12](#), era chamada de Válvula de Fleming ou Válvula-Diodo.

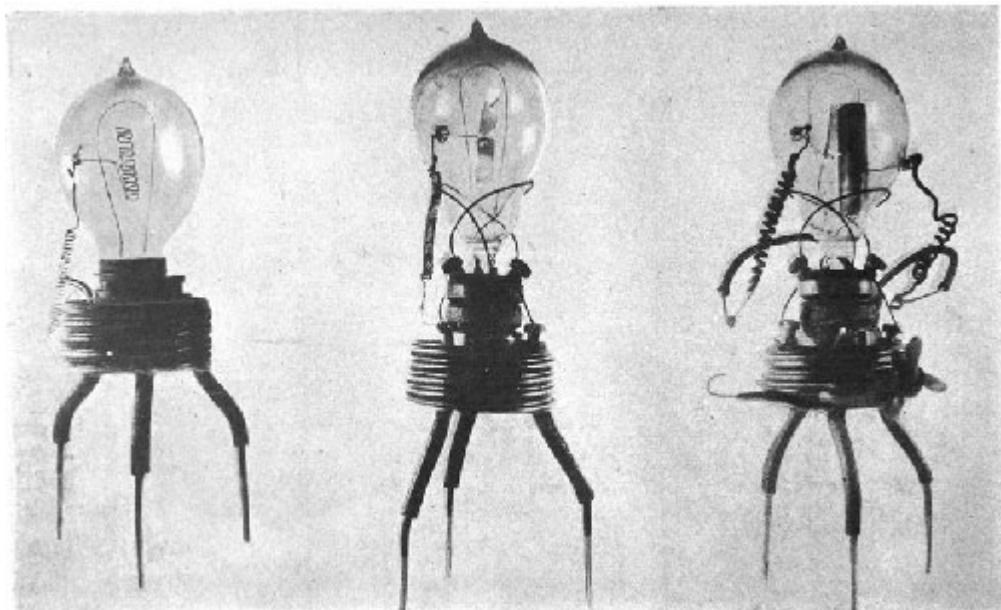


Figura 4.12: Válvulas-diodo de Fleming<sup>11</sup>.

Fleming trabalhou sobre um fenômeno observado inicialmente por Frederick Guthrie (Reino Unido, 1833-1886) em 1873 e posteriormente por Thomas Alva Edison (Estados Unidos, 1847-1931) em 1880, mas que acabou ficando conhecido como efeito

Edison. Ele consiste na criação de uma nuvem de elétrons em torno de um filamento metálico aquecido dentro de um tubo com alto vácuo. O aquecimento do fio, que chega a ficar incandescente, gera tanta energia que expulsa os elétrons do metal e estes ficam circulando no espaço ao redor do cabo.

Se você colocar próximo a este filamento aquecido dentro do vácuo um fio ou placa metálica ligada a uma carga positiva (ânodo), os elétrons vão ser atraídos por essa carga, migrando então do filamento aquecido para a placa. Porém, se esta placa estiver ligada a uma carga negativa, ela vai repelir os elétrons, e assim não haverá corrente nela. A [Figura 4.13](#) mostra esquematicamente os componentes deste tipo de válvula.

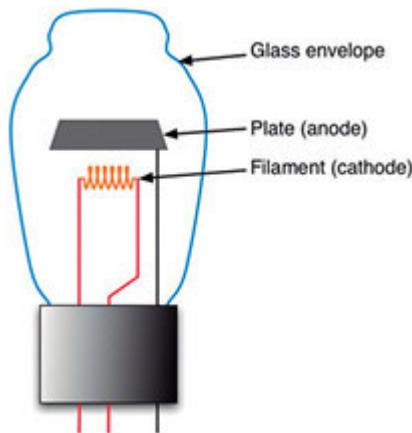


Figura 4.13: Esquema de uma válvula-diodo<sup>12</sup>.

Uma das principais aplicações deste tipo de válvula é a conversão da corrente alternada em corrente contínua, ou seja, a retificação da corrente. Ocorre que é mais fácil e barato gerar e transmitir energia elétrica na forma de corrente alternada, mas esse tipo, que temos em nossas tomadas, é bem diferente da corrente contínua gerada, por exemplo, por uma pilha. A corrente alternada não tem polo positivo e negativo fixos, ela fica justamente alternando o positivo e o negativo o tempo todo. A velocidade dessa alternância é chamada de frequência. Assim, uma corrente que alterna positivo e negativo

60 vezes por segundo é dita corrente de 60 Hertz (Hz), que é uma unidade de medida de frequência.

Desta forma, uma corrente alternada vai aquecer o filamento; mas a que passar da nuvem de elétrons para o ânodo será contínua, pois a válvula, como explicado, só permite que os elétrons entrem na placa, mas nunca saiam.

Outra aplicação muito importante deste efeito consiste na retificação de ondas de radiofrequência. Desta maneira, a válvula de Fleming poderia substituir os diodos de galena, bem mais difíceis de trabalhar, na recepção de sinais de rádio. Ocorre o seguinte: as oscilações eletromagnéticas de rádio que passeiam pela atmosfera são como a corrente alternada, indo do positivo para o negativo de forma muito rápida para ser detectada por um dispositivo como um galvanômetro, por exemplo. Aplicando-se a retificação, porém, aquela oscilação toda é transformada em uma corrente contínua, a qual pode então ser detectada. Em função disso, inclusive, a válvula foi batizada por ele de “válvula de oscilação”, porque ela só deixava passar metade da corrente alternada.

Na computação, as válvulas foram os componentes básicos para a construção das portas lógicas, que substituiriam as engrenagens mecânicas e relês eletromecânicos por circuitos puramente eletrônicos décadas mais tarde.

## 4.13 Cristal Líquido (LCD) – 1904

Tecnologia de cristal líquido ou LCD (Liquid Cristal Display) parece tão moderna que nem imaginaríamos que sua história remete ao século XIX. Em 1888, o botânico e químico Friedrich Richard Kornelius Reinitzer (Bohemia, 1857-1927) ao estudar moléculas de benzoato de colesterol observou um comportamento muito estranho. Essas moléculas tinham dois pontos de fusão (liquefação): o primeiro a 145,5°C e o segundo a 178,5°C. No primeiro estado, o sólido se transformava em um líquido esfumaçado e opaco, e no segundo estado em um líquido cristalino. Inicialmente, Reinitzer

pensou que o fenômeno fosse devido a impurezas na substância, mas não era; o efeito era reversível e purificações adicionais não mudaram absolutamente nada nas observações.

Ele identificou três importantes propriedades para essa substância:

- A existência de dois pontos de fusão.
- A reflexão de luz polarizada circularmente.
- A capacidade de rotacionar a polarização da direção da luz.

Reinitzer solicitou, então, ajuda ao físico Otto Lehmann (Alemanha, 1855-1922), o qual prosseguiu os estudos durante seu pós-doutorado e em 1904 cunhou o termo “cristal líquido”. Lehman imaginou que a fase opaca da substância era um novo estado da matéria, que ele batizou de “cristal líquido”, pois tinha propriedade cristalinas, mas era um líquido.

Porém, ninguém achou aplicação prática para a descoberta na época e durante quase 80 anos ela não passou de uma mera curiosidade científica. Hoje, muitos monitores de computador e TVs são construídos com essa tecnologia.

## 4.14 CTR: Computing-Tabulating-Recording Company – 1911

A história da CTR (Computing-Tabulating-Recording Company) merece ser contada porque ela é a origem da maior empresa de informática do século XX, a IBM. Ela foi criada oficialmente em 1911 por Charles Ranlett Flint (Estados Unidos, 1850-1934), como uma holding, ou seja, uma empresa que não necessariamente produz, mas que detém grandes parcelas das ações e outras empresas, obtendo assim controle sobre elas. As companhias que constituíram a CTR foram:

- *TMC: The Tabulating Machine Company.* Era a empresa de Hollerith, já mencionada, especializada na produção de máquinas de contagem baseadas em cartões perfurados, as

quais eram usadas principalmente nos censos de diferentes países, mas também, cada vez mais, em outros processos.

- *Computing Scale Company of America*. Ela, por sua vez, já era uma holding fundada em 1901, a qual controlava várias empresas que fabricavam balanças comerciais.
- *ITR: The International Time Recording Company*. Tratava-se de uma empresa que fabricava relógios. Teve suas origens em 1894 com o nome de Willard & Frick Manufacturing Company, a primeira fabricante de relógios-ponto do mundo. Em 1900, ela se fundiu com a Bundy Manufacturing Company, passando então a chamar-se ITR.
- *Bundy Manufacturing Company*. Essa companhia, fundada em 1889, é muitas vezes ignorada como fundadora da CTR porque já havia cedido para a ITR seu negócio de fabricação de relógios, tendo permanecido apenas no ramo de calculadoras mecânicas. Contudo, como a [Figura 4.14](#) mostra, ela claramente participou da formação inicial da CTR.



Figura 4.14: Anúncio da criação da CTR<sup>13</sup>.

A empresa consolidada já contava com 1.300 funcionários. Seu primeiro diretor, George Fairchild (Estados Unidos, 1854-1924), era membro do Congresso americano desde 1906 e por isso não se

esperava que ele tomasse um papel ativo na administração. Entretanto, quando o primeiro presidente do negócio saiu após apenas um mês na posição, Fairchild acabou tomando o comando da empresa e ali ficou até 1912, quando foi sucedido pelo ex-comandante da ITR.

A *alma mater* da CTR, entretanto, a pessoa que a levou a tornar-se a gigante IBM, certamente foi Thomas John Watson Sênior (Estados Unidos, 1874-1956). Ele foi efetivado como diretor-geral da CTR em 1914, logo depois de ser condenado criminalmente por formação de cartel – o que levou muitos membros do quadro da empresa a se perguntar como ele a dirigiria da cadeia. Porém, Watson apelou do julgamento e foi atendido. O novo julgamento acabou nunca acontecendo. Assim, ele jamais foi preso, podendo comandar a CTR e depois IBM, ainda por muitos anos.

Watson trouxe de sua antiga empresa o lema “*think*” (pense). Ele considerava que o sucesso de vendas não provinha do gasto da sola de sapato (andar de porta em porta), mas do trabalho com a cabeça (pensar). O lema continua a ser usado pela IBM até hoje, inclusive uma de suas linhas de notebooks era chamada de “Thinkpad”. Comenta-se que o lema “*think different*” (pense diferente) da Apple, décadas mais tarde, tenha sido uma resposta ou mesmo uma provocação à IBM.

Outra influência importante que Watson deixou nestes primeiros anos da CTR foi indicar que a empresa não vendia apenas as melhores máquinas do mercado, ela vendia serviços, ou seja, o cliente comprava não apenas a máquina, mas todo o suporte necessário para fazê-la funcionar.

A CTR operou basicamente dividida em três partes: as balanças, os relógios e as máquinas tabuladoras. O comércio das balanças sempre foi problemático e acabou sendo vendido na década de 1930. Os relógios foram o carro-chefe dos primeiros anos da CTR, mas depois perderam importância. O grande negócio acabou sendo mesmo as máquinas tabuladoras e seus serviços associados. Entre

outras coisas, a maioria dessas máquinas era alugada – não vendida –, o que combinava com a visão de negócios de Watson e permitia maior estabilidade nas rendas da empresa, visto que os aluguéis entravam regularmente no caixa, enquanto que as vendas podiam ter altos e baixos perigosos. Esse modelo de aluguel e serviços continuou sendo praticado pela IBM até os dias de hoje.

Em 1917, a CTR consolidou três empresas canadenses similares às que originalmente a formaram: a Canadian Tabulating Machine Co., Ltd.; a International Time Recording Co. of Canada, Ltd.; e a Computing Scale Co. of Canada, Ltd. A nova holding foi chamada de International Business Machines Co., Ltd. No mesmo ano, a convite do governo brasileiro, a empresa abre um escritório no Brasil e em 1922, na feira do centenário da Independência, faz as primeiras demonstrações de suas máquinas em solo brasileiro.

Após a morte de Fairchild em 1924, quando Watson tomou o controle total da CTR, ele a rebatizou de IBM, inspirado no nome dado às subsidiárias no exterior.

## 4.15 O Enxadrista de Torres y Quevedo – 1912

Voltamos então a falar de Torres y Quevedo. Esse engenheiro espanhol poderia ter perfeitamente servido de inspiração para Carl Barks da Disney criar o Professor Pardal, tamanha a sua capacidade criativa.

Um desses inventos digno de um Pardal foi o Enxadrista (*ajedrecista*), construído em 1912 e apresentado na feira de Paris de 1914. Trata-se simplesmente do primeiro jogo automático da história, uma máquina que executava fisicamente os movimentos de um rei e uma torre, sendo capaz de jogar uma final de partida de xadrez contra um adversário humano com apenas um rei. A máquina é mostrada na [Figura 4.15](#).

A situação é de vitória certa para o autômato, ou seja, para quaisquer jogadas do humano a máquina acabará vencendo. Um pouco injusto talvez, mas mesmo assim, um feito incrível

considerando que estamos falando de 1912 e que computadores ainda estão longe de serem construídos – quanto mais os jogos eletrônicos.

Houve duas máquinas: a primeira movimentava as peças por meio de um braço mecânico e a segunda por eletroímãs que ficavam abaixo do tabuleiro. A posição do rei oponente era sentida pela máquina por meio de sensores elétricos. Os movimentos eram calculados por um algoritmo simples que nem sempre executava a sequência de movimentos mais curta possível, mas ainda assim sempre ganhava. Considere que o computador joga com o rei e a torre brancos e que o humano joga com o rei preto. Considere também o tabuleiro dividido em três zonas: as três primeiras colunas e as três últimas colunas das oito que existem são chamadas “zona de torre” e as duas colunas centrais são chamadas de “zona central<sup>14</sup>”.



Figura 4.15: O enxadrista de Torres y Quevedo<sup>15</sup>.

Em resumo, o algoritmo funciona baseado em três regras simples:

1. Se o rei preto e a torre branca estão na mesma zona, a torre se afasta horizontalmente para outra zona.
2. Se a distância vertical entre o rei preto e a torre branca for de mais de uma fileira, a torre se aproxima verticalmente de forma a reduzir a mobilidade do rei preto.
3. Se a distância entre os dois reis é grande o suficiente para que uma aproximação do rei branco não provoque xeque-mate na jogada seguinte, o rei branco se aproxima do rei preto.

Se o humano fizer um movimento ilegal a máquina não joga até que ele seja corrigido. Se ele fizer isso três vezes a máquina para totalmente de jogar.

Tudo isso é feito com a construção dos contatos e circuitos adequados que compararam as posições relativas das peças e traduzem o movimento adequado de acordo com as regras acima.

Ambas as máquinas ainda funcionam no departamento de máquinas da Universidade Politécnica de Madrid, onde você pode desafiá-las para um jogo (e perder). Pelo menos uma delas anuncia o xeque-mate através de um alto-falante, usando um mecanismo não muito diferente do gramofone.

## **4.16 Máquina Analítica de Torres y Quevedo – 1913**

Possivelmente o sucesso com o enxadrista levou Torres Y Quevedo a sonhar mais alto. Explicitamente inspirado por Charles Babbage, ele idealizou uma máquina analítica capaz de computar quaisquer funções. Porém, em vez de usar discos e engrenagens para representar números, ele propunha que fossem utilizados circuitos eletromecânicos, ao estilo do enxadrista. Isso tornaria a máquina muito menor e mais barata do que a de Babbage.

Ele imaginava que a máquina poderia ter milhares ou talvez até milhões de interruptores, cada qual capaz de estar em diferentes

posições ou estados, e assim representar não apenas algarismos, mas também letras e sinais de pontuação. Assim, ao contrário de Babbage, que via a máquina analítica como um “mastigador de números”, Torres y Quevedo já imaginava que ela poderia armazenar na memória informação não numérica.

Infelizmente, ele não foi capaz de construir a máquina completa, mas apenas algumas provas de conceito. Em sua famosa publicação *Essays on Automatics* de 1913, ele apresenta um projeto bastante detalhado de uma máquina analítica eletromecânica capaz de calcular todos os valores para a fórmula  $\alpha = ax(y-z)^2$ . Entre outros componentes, ele descrevia o mecanismo de entrada de instruções de programação como um disco rotatório com diferentes regiões eletricamente condutoras. Assim, à medida que o disco rodasse, diversos contatos elétricos seriam ativados, causando a execução de certos comandos sobre os registradores da máquina. Em 1914, ele construiu um protótipo para tal engenho.

Consta ainda que neste projeto ele idealizou também um sistema aritmético de ponto flutuante, o qual só foi efetivamente colocado em prática por Konrad Zuse na década de 1940. Números de ponto flutuante são representações de números de maneira diferenciada. Em vez de concatenar os algarismos um ao lado do outro como 984, um número de ponto flutuante usaria uma representação como  $0,984 \times 10^3$ , ou seja,  $0,984 \times 1.000$ , que é exatamente 984. Isso pode parecer irrelevante para um número com apenas três dígitos, mas quando trabalhamos com valores muito grandes, essa notação pode ser bastante econômica. Por exemplo, para representar 5.200.000.000.000.000.000 bastaria escrever  $0,52 \times 10^{22}$ . Na prática, se assumirmos base 10 como padrão, poderíamos representar esse número gigantesco a partir de dois valores apenas: a mantissa 52 e o expoente 22.

A técnica também é útil para representar números muito pequenos. Assim,  $0,34 \times 10^{-16}$  seria a representação em ponto flutuante de 0,000.000.000.000.034. A representação desse número fica

bem mais simples se dissermos que ele tem mantissa 34 e expoente -16.

Outra façanha desse inventor ocorreu em 1920, quando o Arithmometer completava 100 anos. Para celebrar a ocasião, ele construiu um Arithmometer eletromecânico ([Figura 4.16](#)), ou seja, totalmente baseado em circuitos e chaves elétricas, sem giros de manivela. O equipamento era ligado a uma máquina de escrever e funcionava da seguinte maneira: para somar 67 com 156 o operador devia “bater” na máquina “6” e depois “7”, e na sequência a barra de espaço; depois teclava o símbolo da operação, no caso “+”, novamente espaço e em seguida entrava com o segundo número batendo “1”, “5”, “6” e espaço. Até esse momento, a máquina havia escrito “67 + 156” no papel. Mas depois de alguns instantes ela escrevia “=” e em seguida o resultado da operação, que no caso era “223”. Isso certamente assombrou muitas pessoas em pleno 1920. Infelizmente, ele nunca se interessou em produzir a máquina comercialmente e, desta forma, apenas o protótipo para fins de demonstração existiu.

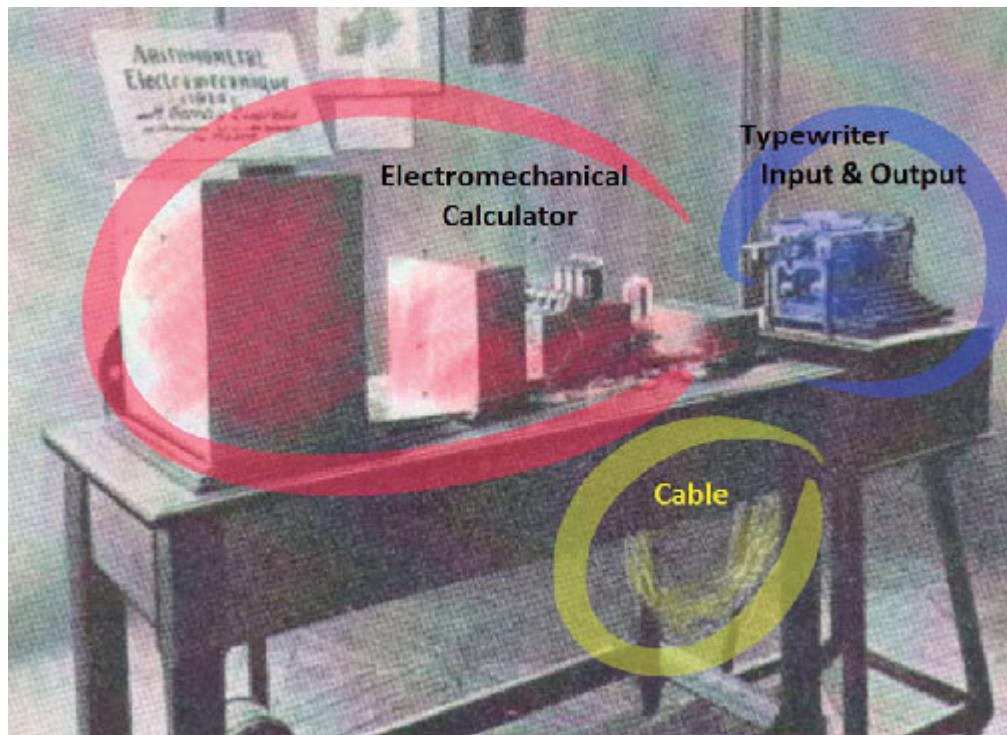


Figura 4.16: Arithmometer eletromecânico<sup>16</sup>.

## 4.17 Flip-Flop – 1918

Um dos circuitos mais importantes para a construção dos primeiros computadores a partir dos anos 1940 foi o flip-flop. Ele foi inventado em 1918 por Willian H. Eccles (Reino Unido, 1875-1966) e Frank W. Jordan (Reino Unido, 1882-?).

O circuito que criaram era baseado em apenas duas válvulas diodo. Ele tem dois estados possíveis, que podem ser trocados a partir de um sinal elétrico, tornando-o assim, perfeito para a representação de um bit (0 ou 1) em um computador.

O primeiro uso documentado de circuitos flip-flop em computadores ocorreu apenas 25 anos depois do registro da patente por parte de Eccles e Jordan, no computador Colossus Mark I.

Para entendermos um pouco melhor o comportamento deste componente, primeiro precisamos compreender que ele é um

círcuito que funciona como uma memória capaz de armazenar dois estados, que podem ser interpretados como 0 e 1.

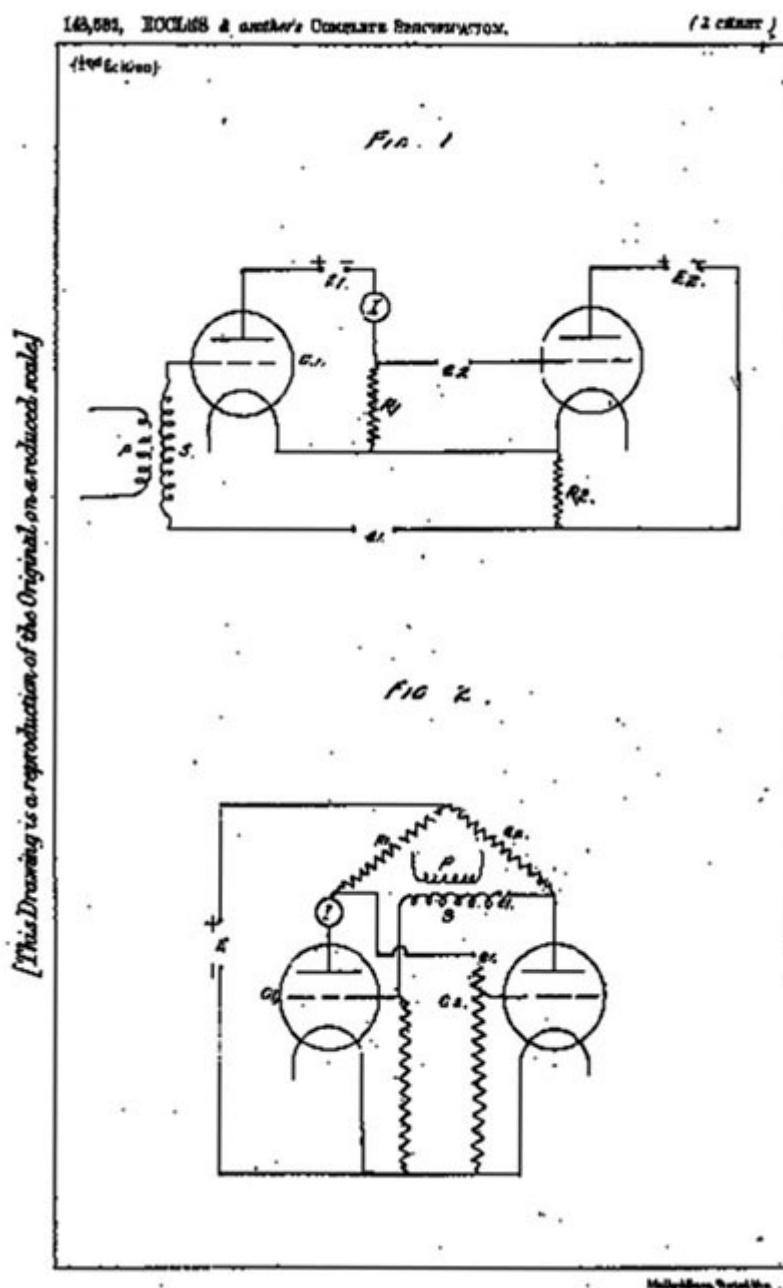


Figura 4.17: Desenho original da patente do flip-flop<sup>17</sup>.

O circuito tem duas entradas nas quais pulsos elétricos podem ser aplicados. Vamos chamar essas entradas de “set” e “reset”. A primeira muda o estado do flip-flop para 1 e a segunda muda-o para 0. Mesmo depois que o pulso de entrada cessa, o flip-flop permanece no estado em que foi colocado.

Ele possui também duas saídas, uma que tem corrente quando ele está no estado 0 e outra que tem corrente quando está no estado 1.

A Figura 4.18 mostra um circuito flip-flop no estado 0. Isso significa que o último contato a ser acionado foi “reset” (contato da esquerda). Neste momento, o coletor de Q1 não está recebendo energia, porque a energia que passa por R2 flui por Q2 e assim o resistor R4 impede que ela fluia para a base de Q1. Neste caso, temos energia fluindo por Q2, mas não por Q1 e o flip-flop está no estado 0.

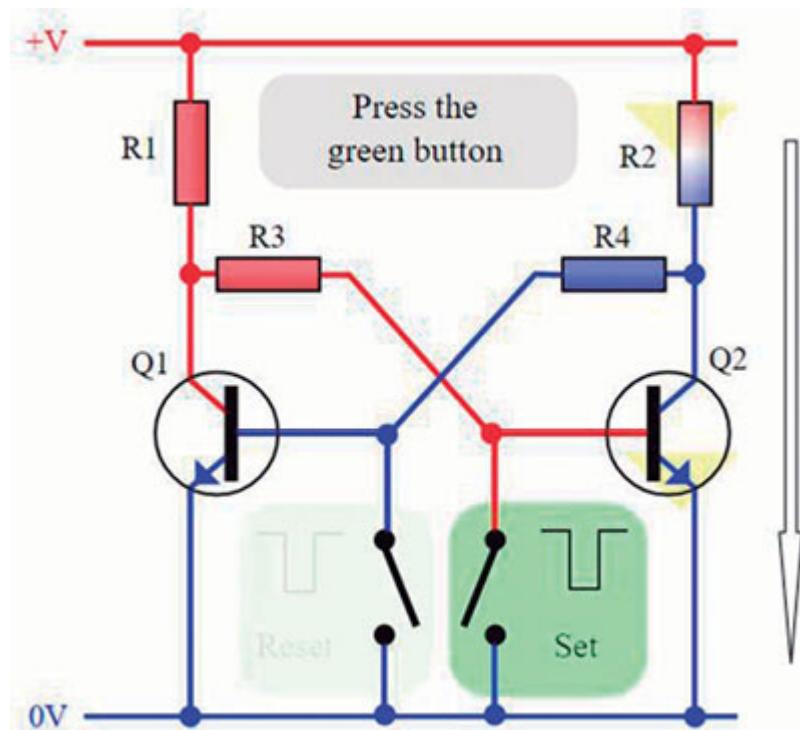


Figura 4.18: Flip-flop no estado 0<sup>18</sup>.

Ao fechar o contato “set” (contato da direita), mesmo que por um breve período, a corrente corre livremente pelo contato (não há resistor) e ela para de entrar pela base de Q2. Como a corrente em Q2 fica, assim, interrompida, a que vem de R2 passa agora a fluir por R4, energizando a base de Q1. Dessa forma a energia passa do coletor para o emissor de Q1 e o flip-flop assume um novo estado, 1, mostrado na Figura 4.19.

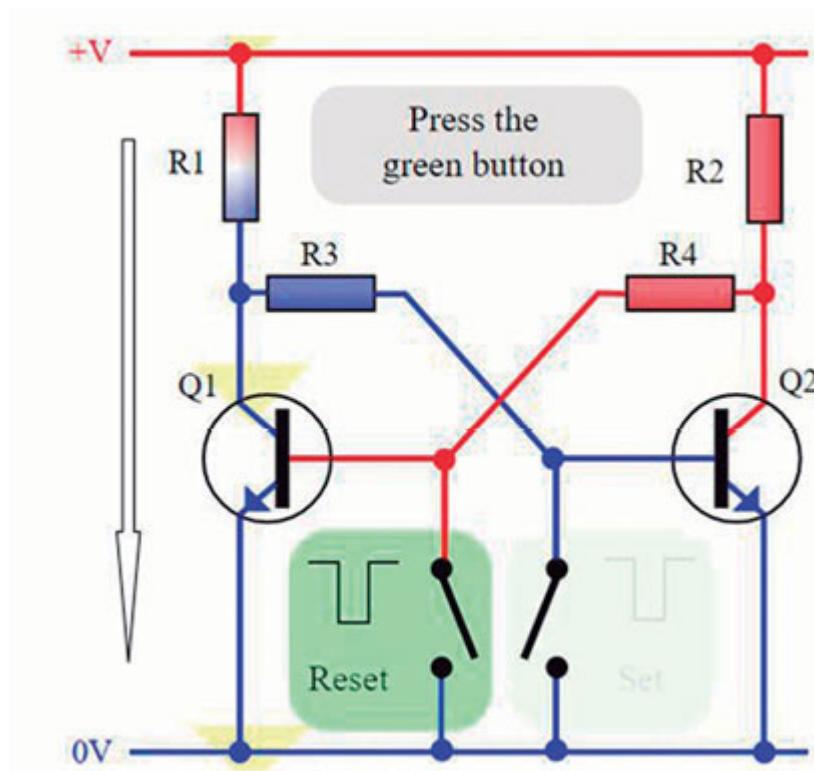


Figura 4.19: Flip-flop no estado 1.<sup>19</sup>

## 4.18 IBM – 1924

Nos anos que seguiram sua consolidação como IBM, a empresa cresceu muito e introduziu muitas práticas totalmente inovadoras para a época, como seguro de vida e férias pagas aos seus funcionários. Seu logotipo original é mostrado na [Figura 4.20](#). Porém, ainda nada de computadores. A empresa obtinha o principal

de suas rendas a partir do aluguel das máquinas tabuladoras no estilo de Hollerith. Além das aplicações já consolidadas no cálculo do censo de vários países, ela ainda diversificava as possíveis aplicações dessas máquinas, como por exemplo, um sistema completo para controle de horários escolares.



Figura 4.20: Logotipo original da IBM.<sup>20</sup>

Durante os anos da grande depressão, a IBM não parou de produzir, mesmo não tendo clientes. Isso permitiu que ela fosse a única empresa capacitada a vencer um grande contrato com a segurança social norte-americana em 1935, visto que contava com um grande estoque de máquinas que nenhuma outra empresa tinha. Esse contrato foi considerado a maior operação de registro de informação até então, visto que 26 milhões de registros de empregados foram mantidos pelos sistemas da empresa. A coisa foi tão positiva que logo contratos de outras áreas do governo começaram a surgir.

A empresa inovou também nessa época em muitos aspectos como, por exemplo, ao introduzir uma máquina tabuladora capaz de subtrair. Até então os números negativos precisavam ser introduzidos como complementos de 9, que deviam ser calculados à mão, o que tomava certo tempo das pessoas que operavam as máquinas. Entretanto, com a introdução dessa nova máquina, o processo passou a ser feito de forma bem mais rápida. Além disso, já em 1931 a empresa produziu a série IBM 600, as primeiras máquinas tabuladoras capazes de multiplicar e dividir.

Em 1933, a IBM adquiriu a empresa Electromatic Typewriters, Inc., o que permitiu não só que ela entrasse no mercado de máquinas de escrever, mas que também adquirisse tecnologia para aprimorar as impressoras que acompanhavam suas tabuladoras.

Em 1934, ela lança a série 801, uma máquina capaz de selecionar, endossar e totalizar cheques bancários, entrando assim em um mercado altamente lucrativo. Em 1937, ela lança o modelo 805, capaz de corrigir provas e dar notas a estudantes.

O importante aqui é mencionar que, embora a IBM não fabricasse computadores nessa época e que não tenha sido a primeira empresa a construí-los, ela criou todo um mercado que fez com que essas máquinas, depois de inventadas para uso científico, tivessem também utilização comercial muito intensa, como veremos mais adiante.

## 4.19 OCR de Gustav Tauschek – 1929

Bem antes de existirem computadores eletrônicos como os atuais, um engenheiro chamado Gustav Tauschek (Áustria, 1899-1945) inventou um aparelho que é considerado o precursor do OCR (Optical Character Recognition). O mecanismo, patenteado na Alemanha em 1929 e nos Estados Unidos em 1935, é mostrado na [Figura 4.21](#).

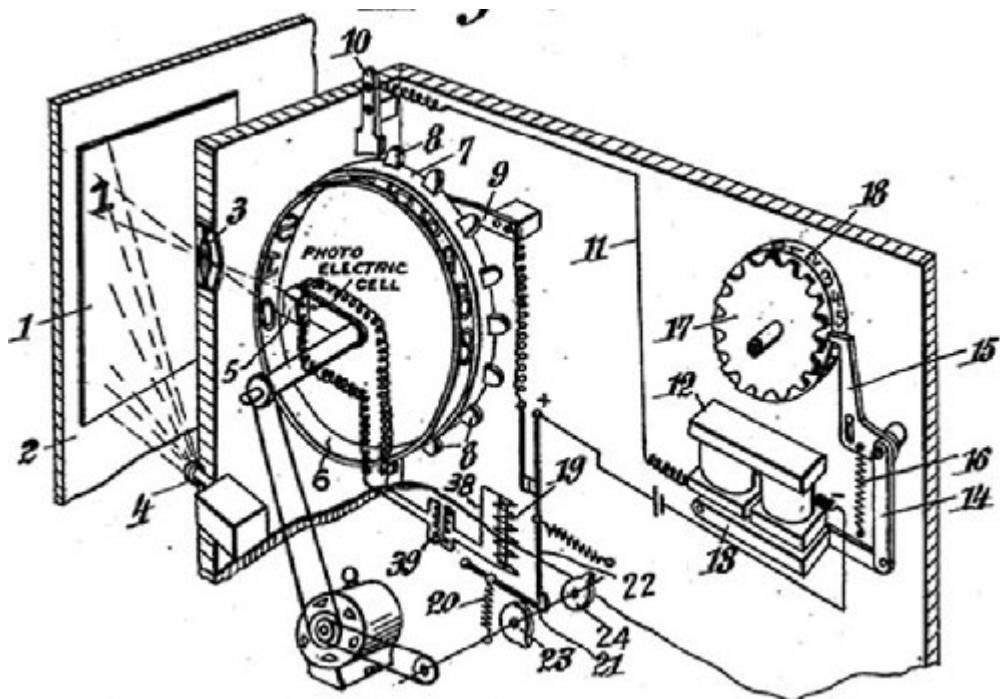


Figura 4.21: Projeto do leitor ótico de Tauschek.<sup>21</sup>

O projeto era simples, mas bem engenhoso. Para reconhecer um símbolo (letra ou algarismo) impresso em uma folha de papel (1), suficientemente iluminada por uma fonte (4), a máquina girava uma roda (6) na qual várias formas de letras e números seriam recortadas de maneira a deixar passar a luz. Uma lente (3) faria a luminosidade do papel alcançar uma célula fotossensível (5). Caso o formato da letra iluminada coincidisse com a forma recortada na roda (6), o nível de energia gerado pela célula ativaría o reconhecimento da letra, o que poderia ser então transmitido para outro mecanismo que a imprimisse ou armazenasse.

O mecanismo, porém, teria algumas falhas que tornariam seu uso muito complicado. Primeiramente, as letras teriam que estar perfeitamente alinhadas e no tamanho exato para serem lidas pela máquina. Em segundo lugar, a máquina poderia ser enganada por falsas letras. Um borrão preto, por exemplo, do tamanho de um

caractere poderia ser identificado como qualquer uma das letras do alfabeto. Assim, a máquina só funcionaria em condições ideais.

## 4.20 Tambor Magnético – 1932

As contribuições de Gustav Tauschek não param por aí. Além da invenção do OCR e de outros dispositivos para trabalhar com cartões perfurados, ele também criou a memória de tambor magnético em 1932.

Tratava-se basicamente de um cilindro giratório e uma série de cabeçotes de leitura e escrita, mostrados no desenho da patente na [Figura 4.22](#). O cilindro era feito de um material que podia ser magnetizado positiva ou negativamente. Os cabeçotes de leitura eram basicamente bobinas magnéticas que podiam ser energizadas para produzir um ou outro estado no material do cilindro. Assim, ele basicamente registrava valores binários, que podiam ser interpretados como 0 e 1.

Ao contrário dos discos magnéticos mais modernos nos quais os cabeçotes se movem para posições específicas de leitura ou gravação, no tambor de Tauschek os cabeçotes eram todos fixos; apenas o cilindro girava. Dessa forma, para gravar ou ler dados do cilindro era necessário indicar qual o cabeçote em questão e qual a posição angular do cilindro que deveriam ser considerados.

O dispositivo construído e patenteado por Tauschek em 1932 tinha capacidade para armazenar 500 mil bits (um bit é um valor que pode ser 0 ou 1), ou seja, aproximadamente 61 kB (kilobytes).<sup>22</sup> Assim, o tambor magnético original podia armazenar cerca de 61 mil números entre 0 e 255, ou 61 mil letras, já que é possível associar um código numérico para cada letra ou símbolo.

A IBM comprou os direitos sobre esta e muitas outras patentes de Tauschek e efetivamente usou estes dispositivos ou suas respectivas evoluções em vários de seus sistemas posteriormente. Afinal, a era da informática estava para começar.

Oct. 4, 1932.

G. TAUSCHEK

1,880,523

SETTING DEVICE FOR CALCULATING MACHINES AND THE LIKE

Filed Oct. 18, 1929

2 Sheets-Sheet 1

FIG.1.

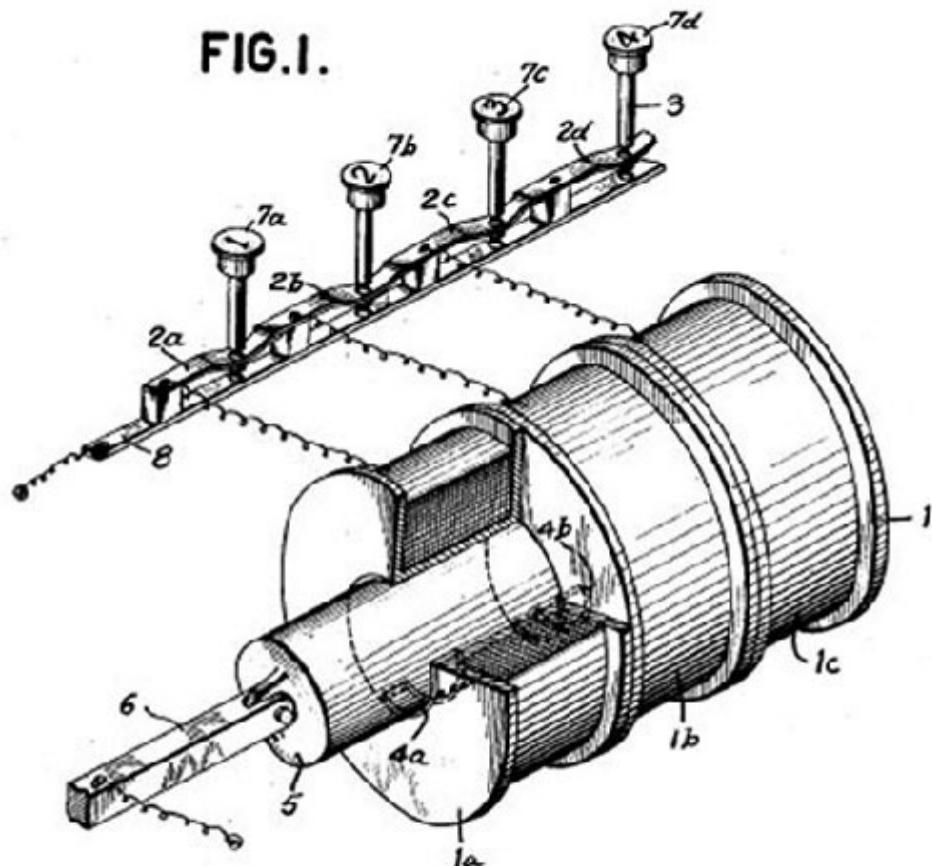


FIG.2.

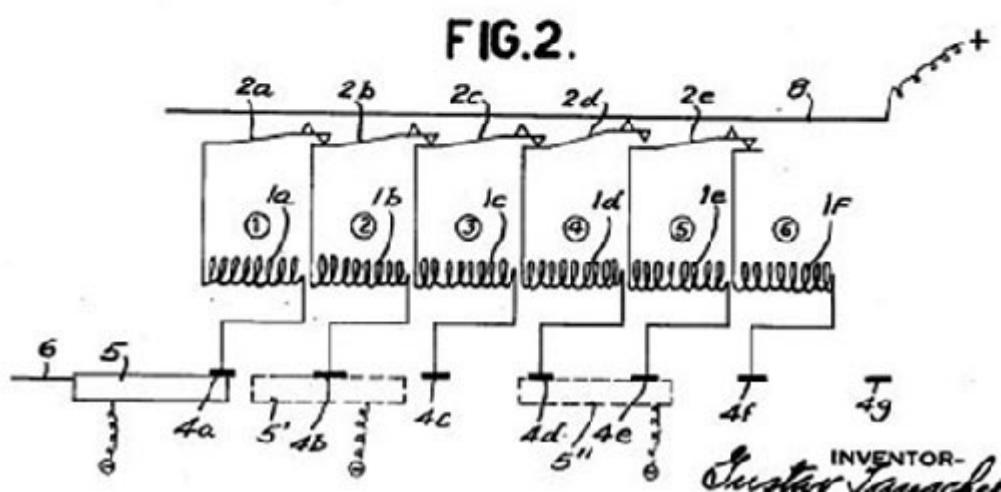


Figura 4.22: Tambor magnético de Gustav Tauschek.<sup>23</sup>

## 4.21 Até Aqui...

Vimos que a virada do século XIX para o XX foi repleta de invenções que mais tarde convergiriam para a construção dos computadores. Essa foi a era em que a palavra “tele” esteve muito em voga, com a invenção do telefone, telefax, televisor e telégrafo.

A automação de escritórios passou a ser um mercado no qual a IBM investia fortemente com suas máquinas tabuladoras e dispositivos de entrada, saída e armazenamento de dados.

Vários componentes que depois serviriam aos computadores foram inventados nesse período, como as memórias magnéticas, calculadoras eletromecânicas, OCR e monitores CRT e LCD. Entretanto, essas invenções ainda não estavam conectadas a uma única máquina. Isso só vai começar a acontecer na fase seguinte, com o efetivo nascimento dos computadores.

---

<sup>1</sup> “CalculatingMachinePatentBurroughs” by William S. Burroughs - Licensed under Public Domain via Commons. Disponível em:<https://commons.wikimedia.org/wiki/File:CalculatingMachinePatentBurroughs.jpg#/media/File:CalculatingMachinePatentBurroughs.jpg>

<sup>2</sup> Domínio público. Disponível em: <http://www.google.com/patents/US348437?hl=pt-BR>

<sup>3</sup> Domínio público. Fonte: Dalakov, G. *History of Computers, Hardware, Software, Internet...* Disponível em: <http://history-computer.com/ModernComputer/thinkers/Peirce.html>

<sup>4</sup> By Elisha Gray(Life time: 1901) - Original publication: United States Patent 386,815 Immediate source: <http://www.freepatentsonline.com/386815.pdf>, Public Domain. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=18595779>

- <sup>5</sup> Domínio Público. Disponível em:  
[http://www.aes.org/aeshc/docs/recording\\_technology.history/tape.html](http://www.aes.org/aeshc/docs/recording_technology.history/tape.html)
- <sup>6</sup> Tradução: Poder-se-ia construir qualquer fórmula?
- <sup>7</sup> “Algebraica1” by MdeVicente - Own work. Licensed under CC0 via Wikimedia Commons. Disponível em:  
<https://commons.wikimedia.org/wiki/File:Algebraica1.JPG#/media/File:Algebraica1.JPG>
- <sup>8</sup> “Braun cathode ray tube” by Eugen Nesper - Retrieved on 11 October 2013 from Eugen Nesper 1921 *Handbuch der Drahtlosen Telegraphie und Telephonie*, Julius Springer, Berlin, p. 78, fig. 46 on Google Books. Licensed under Public Domain via Commons. Disponível em:  
[https://commons.wikimedia.org/wiki/File:Braun\\_cathode\\_ray\\_tube.jpg#/media/File:Braun\\_cathode\\_ray\\_tube.jpg](https://commons.wikimedia.org/wiki/File:Braun_cathode_ray_tube.jpg#/media/File:Braun_cathode_ray_tube.jpg)
- <sup>9</sup> “Telegrafon 8154” by Original uploader was Bitman at hu.wikipedia - Transferred from hu.wikipedia; transferred to Commons by User:Nico-dk using CommonsHelper. Licensed under CC BY-SA 2.5 via Commons. Disponível em:  
[https://commons.wikimedia.org/wiki/File:Telegrafon\\_8154.jpg#/media/File:Telegrafon\\_8154.jpg](https://commons.wikimedia.org/wiki/File:Telegrafon_8154.jpg#/media/File:Telegrafon_8154.jpg)
- <sup>10</sup> Domínio público. Fonte: Dalakov, G. *History of Computers, Hardware, Software, Internet...* Disponível em: <http://history-computer.com/MechanicalCalculators/20thCentury/Rechnitzer.html>
- <sup>11</sup> “Fleming valves” by John Ambrose Fleming - Downloaded July 18, 2013 from John Ambrose Fleming (1919) *The Thermionic Valve and its Developments in Radiotelegraphy and Telephony*, The Wireless Press, London, frontispiece on Google Books. Licensed under Public Domain via Commons -  
[https://commons.wikimedia.org/wiki/File:Fleming\\_valves.jpg#/media/File:Fleming\\_valves.jpg](https://commons.wikimedia.org/wiki/File:Fleming_valves.jpg#/media/File:Fleming_valves.jpg)
- <sup>12</sup> “Fleming valves” by John Ambrose Fleming - Downloaded July 18, 2013 from John Ambrose Fleming (1919) *The Thermionic Valve and its Developments in Radiotelegraphy and Telephony*, The Wireless Press, London, frontispiece on Google Books. Licensed under Public Domain via Commons -  
[https://commons.wikimedia.org/wiki/File:Fleming\\_valves.jpg#/media/File:Fleming\\_valves.jpg](https://commons.wikimedia.org/wiki/File:Fleming_valves.jpg#/media/File:Fleming_valves.jpg)
- <sup>13</sup> Domínio público. Disponível em:  
<http://www.theatlantic.com/technology/archive/2011/06/ibms-first-100-years-a-heavily-illustrated-timeline/240502/>
- <sup>14</sup> Importante mencionar para quem não joga xadrez que a torre pode se movimentar qualquer quantidade de casas em uma única direção, que pode ser horizontal ou vertical. Já o rei pode andar apenas uma casa de cada vez, mas ele

pode ir na horizontal, na vertical ou na diagonal, possibilitando assim um total de 8 movimentos possíveis a cada jogada. Além disso, no xadrez, uma peça “come” a outra se ela consegue, com um movimento, posicionar-se na mesma “casa” que a outra.

15 “Ajedrecista segundo2” de MdeVicente - Trabajo propio. Disponible bajo la licencia CC0 vía Wikimedia Commons - [https://commons.wikimedia.org/wiki/File:Ajedrecista\\_segundo2.JPG](https://commons.wikimedia.org/wiki/File:Ajedrecista_segundo2.JPG#/media/File:Ajedrecista_segundo2.JPG)

16 By Unknown - Bulletin de la Société d'Encouragement pour l'Industrie Nationale - Septembre-Octobre 1920 - page 626, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=30060419>

17 “Eccles-Jordan trigger circuit flip-flop drawings” by Eccles and Jordan - GB 148582 (filed: 21 June 1918; published: 5 August 1920). Licensed under Public Domain via Commons - [https://commons.wikimedia.org/wiki/File:Eccles-Jordan\\_trigger\\_circuit\\_flip-flop\\_drawings.png](https://commons.wikimedia.org/wiki/File:Eccles-Jordan_trigger_circuit_flip-flop_drawings.png#/media/File:Eccles-Jordan_trigger_circuit_flip-flop_drawings.png)

18 Adaptado de: “Transistor Bistable interactive animated EN” by DrJolo - Own work. Licensed under CC BY-SA 3.0 via Commons - [https://commons.wikimedia.org/wiki/File:Transistor\\_Bistable\\_interactive\\_animated\\_EN.svg#/media/File:Transistor\\_Bistable\\_interactive\\_animated\\_EN.svg](https://commons.wikimedia.org/wiki/File:Transistor_Bistable_interactive_animated_EN.svg#/media/File:Transistor_Bistable_interactive_animated_EN.svg)

19 Adaptado de: “Transistor Bistable interactive animated EN” by DrJolo - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Transistor\\_Bistable\\_interactive\\_animated\\_EN.svg#/media/File:Transistor\\_Bistable\\_interactive\\_animated\\_EN.svg](https://commons.wikimedia.org/wiki/File:Transistor_Bistable_interactive_animated_EN.svg#/media/File:Transistor_Bistable_interactive_animated_EN.svg)

20 “Original IBM Logo” by OgilvyOne - Own work. Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Original\\_IBM\\_Logo.png#/media/File:Original\\_IBM\\_Logo.png](https://commons.wikimedia.org/wiki/File:Original_IBM_Logo.png#/media/File:Original_IBM_Logo.png)

21 Domínio público. Disponível em: <http://history-computer.com/ModernComputer/Basis/OCR.html>

22 Um byte é definido como 8 bits. Se cada bit pode ser zero ou um, segundo o sistema de numeração binária, então um byte pode representar números de 0 até 255 ( $=2^8-1$ ). Um quilobyte (kB) são 1.024 bytes. Assim,  $500.000 \times 1.024 = 61$  kB.

23 Domínio público. Disponível em: [http://history-computer.com/ModernComputer/Basis/magnetic\\_drum.html](http://history-computer.com/ModernComputer/Basis/magnetic_drum.html)

# **PARTE V**

## **Os Primeiros Computadores**

Esta sessão inicia e termina com Alan Turing, considerado por muitos um dos pais da computação. Iniciamos com a tese de Turing de 1936, que define o que seriam funções computáveis, e após passar pelo desenvolvimento dos primeiros computadores eletromecânicos e eletrônicos, chegamos a 1950 com o famoso Teste de Turing, que foi proposto como um critério para verificar se uma máquina pode se comportar com a mesma inteligência de um ser humano.

A Segunda Guerra Mundial, entre 1939 e 1945, deu impulso aos governos – especialmente o americano e britânico –, para a construção de máquinas automáticas que decifrassem códigos inimigos e também calculassem trajetórias de mísseis, visto que novas armas estavam sendo criadas e seu comportamento balístico tinha que ser determinado por tabelas de tiro para que pudessem ser úteis. Isso levou à construção de decifradores como a Bomba de Turing, os Colossi e também ao calculador de trajetórias ENIAC, que só ficou pronto depois da guerra.

Essa época foi a era de ouro dos “cérebros eletrônicos”, como eram chamados pela imprensa, o que levou a considerações sobre a capacidade destas máquinas de pensar e mesmo de substituir as pessoas no futuro – coisa que a IBM desde cedo procurou evitar, apresentando o computador como um auxiliar; nunca como substituto do ser humano e seu trabalho. A história destes anos teve que ser reescrita várias vezes ao longo do tempo devido a

revelações de projetos secretos, como os Colossi, ou pouco conhecidos, como a série de computadores Z, desenvolvida na Alemanha por Konrad Zuse.

## 5.1 Máquina de Turing – 1936

Na trilha dos sonhos de Babbage sobre a automatização universal do cálculo, matemáticos ocuparam-se por muitos anos na tentativa de entender ou definir, afinal de contas, o que seria efetivamente computável mecanicamente ou não.

Alan M. Turing (Reino Unido, 1912-1954) foi um matemático que se preocupava exatamente com a questão relacionada ao que era computável e o que não era. Seu trabalho foi apresentado em um artigo, submetido em 1936 e publicado no ano seguinte.

Nessa época, graças à obra dos muitos matemáticos e engenheiros que vieram antes, já se sabia que vários tipos de funções matemáticas eram computáveis, ou seja, podiam ter seus resultados determinados por dispositivos puramente mecânicos, como as engrenagens de Babbage ou os relês de Torres y Quevedo.

Turing, porém, tratou o problema do ponto de vista de uma máquina teórica, que não seria necessariamente construída. Essa máquina, chamada por ele de Máquina A (na qual “A” significa “automática”) tinha uma definição muito simples. Ela consistiria de um cabeçote capaz tanto de ler quanto de escrever símbolos em uma fita. Turing considerou que o conjunto de símbolos deveria ser necessariamente finito porque, como o espaço em cada posição na fita era limitado, se o conjunto de símbolos fosse muito grande, logo haveria símbolos tão parecidos que seria impossível distingui-los, mesmo para uma máquina.

Além disso, ele estabeleceu que a fita onde os símbolos seriam escritos teria um início, mas não necessariamente um fim. Isso não quer dizer que a fita fosse infinita (coisa impossível), mas que sempre seria possível adicionar mais fita ao final da fita corrente, se houvesse necessidade.

O coração da máquina era uma tabela que continha várias instruções que diziam a ela o que deveria fazer se estivesse em um determinado estado e lendo certo símbolo na fita. As ações possíveis eram: mover o cabeçote uma posição para frente ou para trás, manter o cabeçote parado e/ou escrever um novo símbolo na fita na posição atual do cabeçote. Mesmo que já houvesse outro símbolo sobre aquela posição da fita, o novo símbolo substituiria completamente o anterior, não restando nenhum resquício dele.

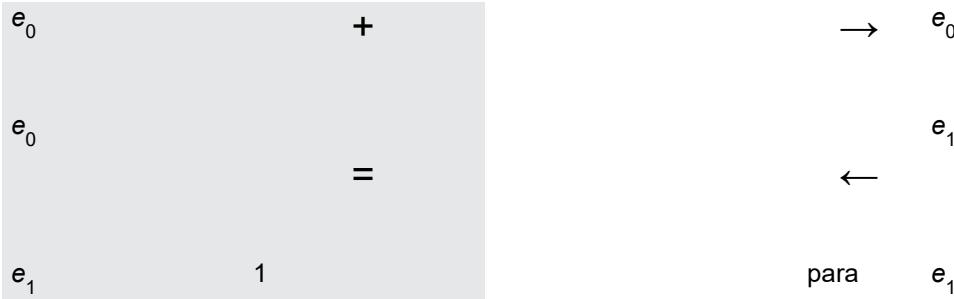
O “estado” da máquina era uma espécie de memória muito simples. Ela estaria em cada instante de tempo em determinado estado que pertencia a um conjunto finito, pois, para que a tabela de instruções fosse finita, o conjunto de estados possíveis da máquina também tinha que ser finito.

Vamos exemplificar o funcionamento da Máquina de Turing definindo uma máquina que soma dois números em unário, como por exemplo  $111+11=11111$  (traduzindo para decimal,  $3+2=5$ ). O que esse equipamento precisa fazer, iniciando a leitura pelo lado esquerdo da expressão “ $111+11=$ ”, é passar pela primeira sequência de 1s, substituir o sinal “+” por “1” e, quando encontrar o sinal “=”, apagá-lo e excluir também o “1” que estiver imediatamente à sua esquerda.

A tabela de instruções da máquina poderia considerar a existência de dois estados:  $e_0$ , no qual a máquina está escaneando os números, e  $e_1$ , no qual ela encontrou o sinal de “=” e vai finalizar a computação apagando o último 1 da sequência. A [Tabela 5.1](#) mostra um possível conjunto de instruções para essa máquina.

**Tabela 5.1: Um conjunto de instruções para uma Máquina de Turing**

O que a máquina vê	O que a máquina faz			
Estado atual	Símbolo lido	Escreve símbolo	Move	Vai para estado
$e_0$	1	1		$e_0$
			→	
		1		



Suponha que se queira somar 3 e 2, como no exemplo. Neste caso, teríamos que preencher a fita com os seguintes símbolos:

1            1            1            +            1            1            =

$\uparrow e_0$

Inicialmente, a máquina está no estado  $e_0$  (chamado de estado inicial), e seu cursor encontra-se sobre o primeiro símbolo 1. Olhando para as duas primeiras colunas da [Tabela 5.1](#), vemos que a máquina, neste caso, deve escrever o símbolo 1 (ou seja, não altera o símbolo na fita) e se mover uma posição para a direita, permanecendo no estado  $e_0$ :

1            1            1            +            1            1            =

$\uparrow e_0$

A máquina novamente está no estado  $e_0$  e visualizando o símbolo 1. Assim ela repete o mesmo procedimento:

1            1            1            +            1            1            =

$\uparrow e_0$

Ainda mais uma vez o procedimento é repetido:

1      1      1      +      1      1      =  
                                ↑e<sub>0</sub>

Agora, vemos que a máquina está no estado  $e_0$ , mas está lendo o símbolo “+”. Segundo a [Tabela 5.1](#), neste caso, ela deve escrever o símbolo 1, mover o cursor para a direita e permanecer no estado  $e_0$ :

1      1      1      1      1      1      =  
                                        ↑e<sub>0</sub>

Novamente no estado  $e_0$ , e visualizando o símbolo 1, a máquina deve imprimir 1, mover-se para a direita e permanecer no estado  $e_0$ :

1      1      1      1      1      1      =  
                                        ↑e<sub>0</sub>

Estando em  $e_0$  e lendo “1”, ela novamente escreve “1”, move-se para a direita e permanece em  $e_0$ :

1      1      1      1      1      1      =  
                                        ↑e<sub>0</sub>

Agora, com a máquina no estado  $e_0$  e o símbolo lido “=”, segundo a tabela, ela deve escrever um espaço em branco e retornar uma posição para a esquerda, além de passar para o estado  $e_1$ , que

significa que a computação está acabando e que só falta apagar o último dígito para ficar com o resultado impresso na fita:

$$1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1$$

$\uparrow e_1$

Com a máquina no estado  $e_1$  e o símbolo 1 na leitura, a máquina deve apagar esse símbolo e permanecer no mesmo estado e posição:

A partir daí a máquina para de se mover e o resultado da computação, 5 em unário, está representado na fita.

A importância dessa máquina simples não está em poder ser ou não construída ou em sua eficiência, mas no fato de que até hoje ninguém conseguiu definir uma máquina capaz de computar funções que essa não possa.

A conjectura de que ela é capaz de computar todas as funções que são efetivamente computáveis é chamada de “Tese de Church-Turing”<sup>1</sup> e costuma ser universalmente aceita, mesmo nunca tendo sido provada. Ela é aceita porque muitas tentativas de definir máquinas mais poderosas do que a descrita acima foram feitas e, invariavelmente todas elas poderiam ser simuladas por uma máquina de Turing simples. Ou seja, tudo o que um engenho mais sofisticado pudesse computar, a Máquina de Turing também poderia. Isso vale inclusive para os nossos mais modernos computadores, os quais são versões limitadas da máquina proposta por Turing, já que sua memória (correspondendo à fita) não é virtualmente infinita.

Por exemplo, pense numa máquina na qual a fita fosse infinita nas duas direções. Ela poderia ser simulada por uma máquina simples na qual as posições pares seriam o lado direito da fita e as posições ímpares o lado esquerdo da fita. Ou seja, a máquina original poderia

simular essa máquina pretensamente mais sofisticada, desde que o conjunto de instruções considerasse a disposição das células como mostrado abaixo:

Máquina com fita duplamente infinita	...	-3	-2	-1	0	1	2	3	...
Máquina original	0	1	-1	2	-2	3	-3	...	

Outro exemplo seria um mecanismo capaz de pular um número arbitrário de casas, em vez de andar apenas uma posição de cada vez. Novamente, foi formalmente demonstrado que uma Máquina de Turing simples poderia simular esta máquina. E assim por diante.

Ele chegou mesmo a propor que uma Máquina A com um conjunto de instruções especialmente programado poderia lê-lo na sua fita de maneira tal que pudesse simular qualquer outra Máquina A. Ele a chamou de “Máquina Universal”. Sua concepção foi importante porque foi a primeira vez que se definiu um conceito importantíssimo para a computação, o qual levou posteriormente à tecnologia de “programa armazenado”, ou seja, a máquina não teria apenas uma programação predefinida codificada em seu próprio mecanismo, como uma calculadora, mas ao ler o programa em uma fita ou qualquer outra forma de entrada de informação, ela seria capaz de computar qualquer função que possa ser computável por qualquer outra máquina de computação.

Uma máquina de computação é considerada Turing-completa se ela for equivalente a uma Máquina de Turing. Máquinas especializadas, como as calculadoras de Leibniz e a máquina diferencial de Babbage, não são Turing-completas, pois existem muitas funções computáveis que elas não são capazes de computar. Já a Máquina Analítica de Babbage é considerada Turing-completa porque seria capaz de computar qualquer função computável por uma Máquina de Turing.

Turing deu importantes contribuições aos Aliados durante a Segunda Guerra Mundial ao ajudar a quebrar os códigos

criptográficos dos inimigos e na década de 1950 ele propôs o famoso Teste de Turing sobre o qual ainda falaremos.

## **5.2 Rapid Arithmetical Machine de Vannevar Bush – 1936**

Vannevar Bush (Estados Unidos, 1890-1974) publicou um artigo muito interessante em 1936 no qual faz um apanhado extenso sobre todas as tentativas até então de realizar computação digital e analógica. O artigo, intitulado “Instrumental Analysis” e publicado pela Sociedade Americana de Matemática, discorria sobre a possibilidade de computar funções matemáticas usando a técnica analítica (o mesmo método da Máquina Analítica de Babbage).

Entre outras tecnologias, ele analisou o uso de cartões perfurados, na época já extensivamente utilizados para registrar informação pela IBM e outras empresas. Bush imaginou, talvez pela primeira vez, que além de ler um cartão perfurado com contatos elétricos – como na tabuladora de Hollerith – ou por ação puramente mecânica – como nos teares de Jacquard –, talvez fosse possível usar raios de luz e células fotoelétricas para obter leituras mais rápidas e sem danificar os cartões, pois não haveria contato físico entre a parte perfurada do cartão e o dispositivo de leitura.

Além disso, especulava que os cartões eram muito pouco aproveitados, pois a maioria tinha 80 colunas e 10 linhas, ou seja, 800 posições nas quais poderiam ser furados. Porém, cada linha representava um dígito de 0 a 9 e apenas um furo poderia ser usado por coluna. Assim, um cartão era capaz de representar apenas 80 dígitos de 0 a 9. Com a possibilidade de se utilizar mais do que um único furo por coluna, fazendo combinações de duas ou mais perfurações em cada uma, seria possível representar uma quantidade muito maior de dados, multiplicando-se assim enormemente a capacidade desses cartões de representar informação.

Bush descreve no artigo já citado uma máquina na qual começou a trabalhar secretamente no MIT nos anos seguintes. Essa máquina seria um computador bastante sofisticado, mas apenas partes dele ficaram prontas, pois a guerra iniciada em 1939 acabou gerando outras prioridades. Se fosse concluída, esta máquina teria um conceito interessante de separação de controle e dados. Ela possuiria três fitas perfuradas: uma para controle, outra para dados e a restante para constantes.

O computador de Bush, chamado “Rapid Arithmetical Machine”, embora nunca tenha sido terminado, foi possivelmente um dos primeiros a ter sua construção iniciada. Embora exista registro de que a construção da máquina a partir de 1936 tenha sido documentada, esses documentos nunca foram encontrados. De fato, sua própria existência foi esquecida até que um processo judicial do início dos anos 1970 que questionava a legitimidade da patente do ENIAC exigiu uma extensa pesquisa histórica sobre os primeiros desenvolvimentos em computação.

Partes do computador foram produzidas, como somadores e contadores. Elas eram construídas com válvulas e relês. Uma parcela da pesquisa consistiu na tentativa de construir válvulas melhores para o projeto em relação às que existiam até então. Os esquemas de dessas partes foram aproveitados ou pelo menos influenciaram outros. Por exemplo, um estudante do MIT chamado Perry Crawford apresentou em sua dissertação de mestrado uma calculadora eletrônica projetada para calcular em tempo real a posição futura de um alvo móvel. Perry usou em seu trabalho pelo menos a ideia do multiplicador de Bush, além do gerador de funções e de um projeto primitivo de disco magnético. Como Crawford acabou trabalhando no computador Whirlwind do MIT, mais tarde, ele levou esse conhecimento para o novo projeto.

### **5.3 Z1 de Konrad Zuse: O primeiro computador binário programável – 1936**

Alguns livros de história ignoram, mas o primeiro computador binário programável foi construído na Alemanha entre 1936 e 1938. Não se tratou, entretanto, de um projeto do governo nazista, mas da iniciativa pessoal de um brilhante engenheiro: Konrad Zuse (Alemanha, 1910-1995).

Zuse construiu, nessa época, uma série de quatro computadores que ele chamou inicialmente de V1, V2, V3 e V4, no qual o "V" procedia de "Versuchmodell 1", ou seja, "Modelo Experimental 1". Porém, após a guerra, e para evitar associações com as bombas de mesmo nome, rebatizou seus inventos para Z1, Z2, Z3 e Z4. Ele também construiu mais duas máquinas de propósito especial, o S1 e o S2, além de vários outros computadores.

O Z1 foi o primeiro computador programável efetivamente construído e completado. Ele ainda não era totalmente confiável, o que levou seu inventor a construir outros modelos, mas foi completado em 1938, bem antes daqueles que foram por anos considerados os primeiros computadores. O ENIAC, por exemplo, que por muito tempo foi creditado como o primeiro computador, só foi completado sete anos mais tarde – e, ao contrário do Z1, ele não era Turing-completo.

O Z1 foi construído na sala de estar dos pais de Konrad apenas com recursos dele próprio e de amigos e parentes que o ajudaram. Ele não era ainda eletrônico, mas um computador com base mecânica. Porém, ao contrário da Máquina Analítica de Babbage, que usaria discos para representar a aritmética decimal, o Z1 trabalhava com base binária, que, como vimos, era de automatização bem mais simples. O computador não tinha relês, mas pinos metálicos que podiam estar em duas posições distintas, representando o valor 0 ou 1.

A única parte elétrica da máquina era o motor que deveria movimentá-la a um "clock" de 4 Hz, ou seja, quatro ciclos de operação por segundo. Entretanto, o modelo reconstruído do Z1 não

era capaz de passar de 1 Hz. Nesta velocidade, uma multiplicação levava cerca de 20 segundos.

O Z1 era capaz de realizar as quatro operações: soma, subtração, multiplicação (por somas sucessivas) e divisão (por subtrações sucessivas). Ele utilizava números de ponto flutuante, ou seja, números representados por uma mantissa e um expoente.

O programa não era armazenado em memória, mas lido a partir de uma fita perfurada. Além disso, ele tinha um dispositivo de entrada de dados na forma de um teclado, e um dispositivo de saída. Tanto o dispositivo de entrada quanto o de saída faziam a conversão do formato binário para o decimal. Ou seja, o usuário podia entrar com números decimais, a máquina convertia para binário, operava, obtinha o resultado em binário, e convertia novamente para decimal.

O Z1 tinha dois registradores internos, R1 e R2, sobre os quais a unidade de aritmética podia operar. A memória da máquina original consistia de 16 posições (64 na reconstruída), cada qual capaz de representar 24 bits: um para o sinal, sete para o expoente e 16 para a mantissa.

Os programas para o Z1 eram perfurados em uma fita capaz de representar 8 bits. Havia oito instruções possíveis divididas em três grupos:

1. Entrada e saída:

- a. Lu – para carregar um valor decimal a partir do dispositivo de entrada.
- b. Ld – para enviar um valor decimal ao dispositivo de saída.

2. Leitura e escrita na memória:

- a. Pr z – para colocar o conteúdo da célula de memória z em R1 se estivesse vago ou R2 caso R1 estivesse ocupado.
- b. Ps z – para colocar o conteúdo do registrador R1 na célula de memória z.

3. Quatro instruções aritméticas:

- a. Ls1 – para adicionar dois números nos registradores R1 e R2.

- b. Ls2 – para subtrair dois números nos registradores R1 e R2.
- c. Lm – para multiplicar dois números nos registradores R1 e R2.
- d. Li – para dividir dois números nos registradores R1 e R2.

Estas instruções eram codificadas em 8 bits. No caso de Pr e Ps, os dois primeiros bits indicavam o código da operação e os seis últimos o endereço de memória. Assim, a máquina tinha uma capacidade de endereçamento de  $2^6 = 64$  palavras.<sup>2</sup> Já no caso das instruções de entrada e saída e das instruções aritméticas eram necessários 3 bits para indicar o código da instrução.

O Z1 original foi destruído em um bombardeio Aliado em 1943, mas depois de se aposentar Zuse construiu uma réplica, que foi doada ao Museu de Tecnologia de Berlin. Infelizmente, a réplica era tão pouco confiável quanto o original e acabou falhando após operar durante algum tempo. Zuse gastou vários meses para reparar o mecanismo, e após sua morte em 1995 ele nunca mais foi ligado.



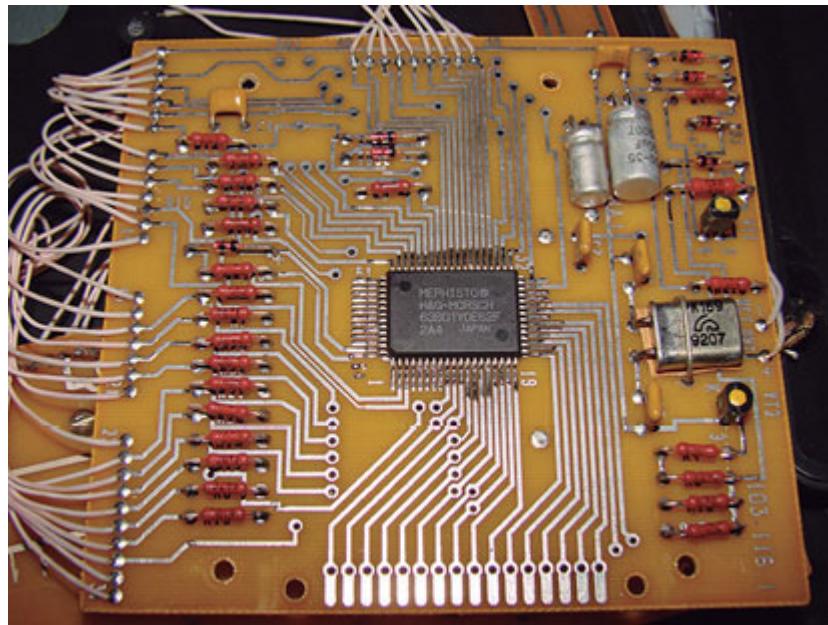
**Figura 5.1: Réplica do Z1 no Museu de Tecnologia de Berlin, exibida pelo filho de Konrad Zuse, Horst.<sup>3</sup>**

## **5.4 Placa de Circuito Impresso – 1936**

Ainda na década de 1930 houve um acontecimento que seria fundamental para a eletrônica e, por conseguinte, para a computação nas décadas seguintes: a invenção da placa de circuito impresso.

A tecnologia de impressão de contatos para ligação de componentes eletrônicos em placas rígidas foi inventada por Paul Eisler (Áustria, 1907-1992) em 1936. Até então, os componentes eletrônicos como relês e válvulas eram ligados uns aos outros através de fios que eram soldados um a um nos contatos dos respectivos componentes. O processo era manual, suscetível a erros e não automatizável.

Eisler teve a ideia de pintar ou imprimir uma fina camada de metal em uma placa rígida, formando as conexões que ligariam os diversos dispositivos. A placa teria furos nos quais seriam inseridos os contatos dos dispositivos eletrônicos, como válvulas, e estas, depois de soldadas na lâmina metálica, estariam corretamente interligadas. Esse processo, sim, era factível de automatização e reprodução em massa. A [Figura 5.2](#) mostra uma versão moderna de uma placa de circuito impresso com vários componentes já soldados a ela.



**Figura 5.2: Placa de circuito impresso moderna.<sup>4</sup>**

No início, a invenção não teve boa aceitação porque o processo de ligação manual era realizado por mulheres, que para os industriais eram mão de obra bastante barata e flexível. Apenas quando os Estados Unidos passaram a usar as placas de circuito impresso em seus equipamentos bélicos é que Eisler passou a ter uma demanda significativa pela sua invenção.

## **5.5 Modelo K de George Stibitz – 1937**

Durante a década de 1930 alguns conceitos que acabaram depois sendo importantes para a construção dos primeiros computadores começaram a ser desenvolvidos, por vezes até em paralelo, como se a evolução dessa ciência fosse um fato inevitável.

Muitas dessas contribuições saíram da Bell Telephone Laboratories. Nos primeiros anos elas foram especialmente relacionadas à computação com relês, dispositivo que uma empresa de telefonia como a Bell dominava.

Em uma bela noite de 1937, um dos pesquisadores da companhia chamado George Stibitz (Estados Unidos, 1904-1995) levou para

casa algumas peças: dois relês telefônicos, duas lâmpadas de lanterna, uma bateria e alguns fios. Sobre a mesa da cozinha, ele montou com essas peças, e mais dois interruptores improvisados a partir de uma lata de tabaco, um circuito somador binário simples. Sua esposa batizou o protótipo como “Modelo K”, no qual “K” vinha de “*kitchen table*” ou “mesa da cozinha”.

Não encontramos nenhuma descrição sobre o circuito ou funcionamento do modelo K, mas, salvo melhor juízo, a partir dos desenhos e fotos da máquina é possível deduzir que as lâmpadas mostrariam o resultado da soma de 2 bits representados por cada um dos interruptores. Se nenhum interruptor fosse acionado, as duas lâmpadas ficavam apagadas, ou no estado 00. Sendo acionado qualquer um dos interruptores e o outro permanecendo não acionado, seria acesa a lâmpada da direita, no estado 01. Sendo acionados os dois interruptores, haveria a transmissão do *carry* para a lâmpada da esquerda, ficando a máquina no estado 10.

No dia seguinte, ele levou o protótipo para a empresa e sugeriu a seus colegas que poderia ser construído um modelo efetivo da máquina que realizasse somas de números grandes. Porém, a ideia não foi muito bem recebida porque alguns consideraram que esse equipamento seria maior e mais caro do que as calculadoras mecânicas existentes, não havendo assim vantagem nele.

Contudo, algum tempo mais tarde os executivos da Bell resolveram investir na ideia de Stibitz pois a empresa vinha se defrontando com problemas de cálculo cada vez mais intrincados, especialmente quando envolviam números complexos (números com uma parte inteira e uma imaginária). Divisões e multiplicações de números complexos exigiam longas sequências de operações, o que estava muito além da capacidade das calculadoras mecânicas. A aposta nos computadores construídos com relês teve início. Mais adiante retornaremos ao trabalho de Stibitz na Bell.

## 5.6 Louis Pierre Couffignal – 1938

Louis Pierre Couffignal (França, 1902-1966) era um matemático que foi um dos pioneiros da informática e cibernética na França. Ele publicou em 1938 uma tese de doutorado na qual avaliava as vantagens de construir máquinas calculadoras com base binária em vez de decimal.

Presume-se que tivesse a intenção de construir a máquina juntamente com a empresa francesa Logabax, mas possivelmente devido à guerra não o tenha feito. Apenas em 1952 ele conseguiu seu intento, ajudando a Logabax a produzir o primeiro computador digital eletrônico francês, com 2 mil válvulas.

## 5.7 Z2 – 1939

Ainda em 1936, Konrad Zuse convidou seu amigo Helmut Schreier (Alemanha, 1912-1984) para conhecer o Z1. Schreier ficou impressionado e disse a Zuse que ele tinha que construir o computador com válvulas. Zuse, entretanto, não foi inicialmente muito favorável à ideia.

Apesar de ter decidido construir seu segundo computador, o Z2, com relês e partes mecânicas, a ideia de Schreier continuou evoluindo. Em 1938, ele defendeu sua tese de doutorado na qual propunha a construção de um computador com 2 mil válvulas. Como os maiores equipamentos a válvula da época não tinham mais do que algumas centenas, a ideia foi considerada absurda.

Mesmo quando Schreier tentou vender seu trabalho para os militares, ele foi ridicularizado por estabelecer um prazo de dois anos para a construção da máquina. Os militares alemães estavam certos de que venceriam a guerra bem antes disso e que então o investimento seria um desperdício. Porém, praticamente na mesma época, o físico americano John Atanasoff também teve a ideia de construir máquinas calculadoras com válvulas, o que acabou levando à construção do ABC, o primeiro computador a válvulas termiônicas, entre 1939 e 1941.

Uma curiosidade em relação a Schreier é que ele migrou para o Brasil e foi professor do Instituto Militar de Engenharia, IME, no Rio de Janeiro, onde residiu até falecer.

Para a construção do Z2, Zuse não estava satisfeito com a confiabilidade das barrinhas de metal usadas na fabricação do Z1. Essas barras entortavam após algum tempo de uso, inutilizando o equipamento. Por outro lado, relês eram caros e uma máquina baseada em milhares de relês estaria muito acima de suas possibilidades financeiras, além de ser um empreendimento que ocuparia muito espaço físico.

Zuse, então, decidiu construir a unidade de aritmética com relês de modo a torná-la mais rápida e confiável, mas manter a memória do Z2 implementada mecanicamente. A memória desta máquina tinha apenas 16 variáveis de 16 bits que representavam números de ponto fixo (ou seja, ao contrário da aritmética de ponto flutuante do Z1, no Z2 a quantidade de casas decimais dos números usados nos cálculos era fixada antes de colocar a máquina em operação).

Como Zuse não tinha muito dinheiro, buscou relês抗igos usados por companhias telefônicas. Ele mesmo e seus ajudantes limparam e recondicionaram cerca de 800 relês para usá-los no Z2. Não há fotos nem reconstruções dessa máquina.

O Z2 era significativamente mais rápido do que o Z1, chegando a operar a 3 Hertz. Além disso, em vez de uma fita de papel, por sugestão de Schreier, ele recebia programas e dados através de um filme de 36mm reaproveitado e perfurado. O conjunto de instruções do Z2 era igual ao do Z1.

Infelizmente, tanto o Z2 quanto seus planos e fotos foram totalmente destruídos em um bombardeio Aliado durante a guerra.

## **5.8 Enigma e a Bomba Criptológica de Turing – 1939**

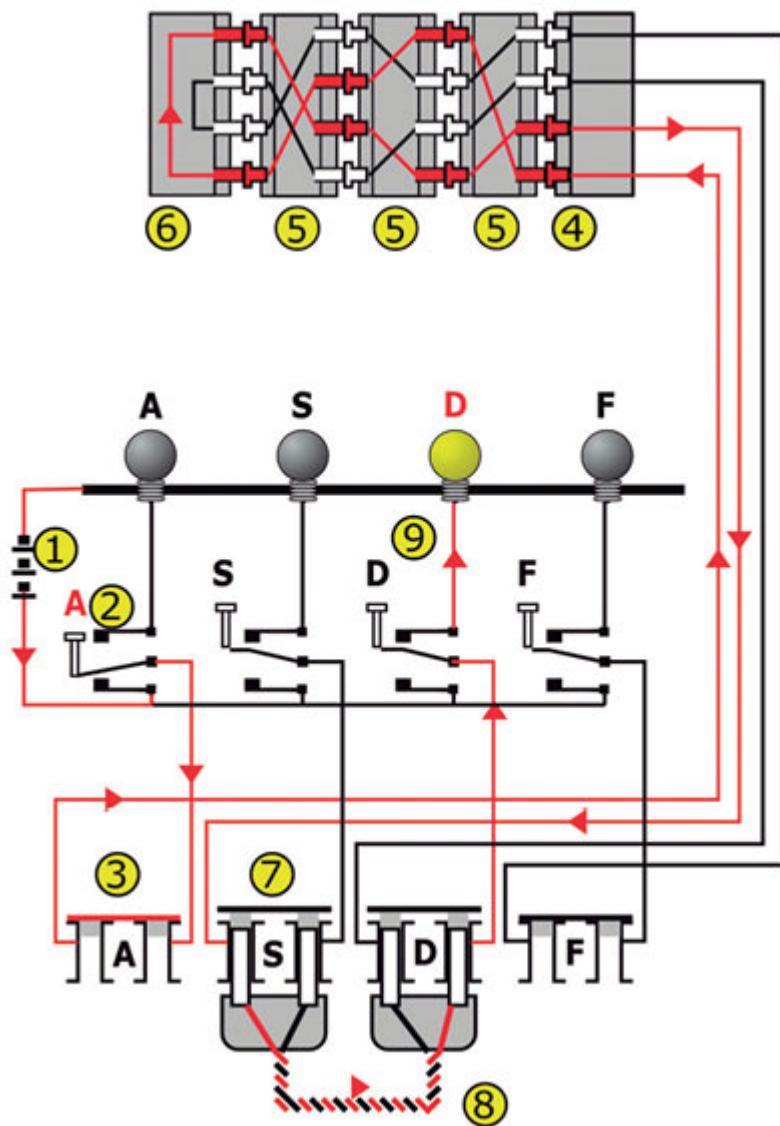
Um dos primeiros grandes embates da história da computação aconteceu durante a Segunda Guerra Mundial e esteve relacionado

ao uso de mensagens criptografadas. Já vimos que técnicas de criptografia e decifração eram conhecidas desde o século IX e foram preservadas pelo trabalho de Al-Kindi.

Porém, a técnica de substituição de símbolos descrita por Al-Kindi era muito fácil de ser quebrada. Como vimos, cada letra é substituída sempre por um mesmo símbolo e, assim, basta fazer uma contagem da frequência de ocorrência de cada símbolo na mensagem criptografada para se ter uma boa ideia de qual letra se trata. Daí em diante é só ir substituindo cada símbolo pela letra correspondente, montando o quebra-cabeças.

As técnicas de criptografia usadas na Segunda Guerra Mundial eram bem mais difíceis de decifrar, entretanto. Entre outras máquinas, os alemães usavam uma que ficou muito famosa: a Enigma, inventada logo depois do fim da Primeira Guerra Mundial por Arthur Sherbius (Alemanha, 1878-1929). Essa máquina fazia girar um conjunto de rotores a cada letra teclada, de forma que a mesma letra fosse substituída por caracteres diferentes em posições diversas do texto. Assim, uma mensagem que originalmente se lia “AAAAAAA” poderia ser criptografada como “JZHSYXGSW”.

A [Figura 5.3](#) mostra como isso era feito. A figura simplificada exibe apenas quatro teclas e seus circuitos. A versão original, no entanto, tinha as 26 letras do alfabeto alemão e um circuito para cada uma. Na parte inferior da figura vemos a tecla “A” sendo pressionada. Como resultado, uma luz é acesa sobre a letra “D” na parte central da figura. Isso indica que neste instante “A” será cifrado como “D”.



**Figura 5.3: Parte do circuito interno da Enigma.<sup>5</sup>**

Mas como o processo ocorre? Observe que a pressão em “A” ativa um circuito elétrico que leva um sinal até um conector (4). Que faz contato com um dos três rotores indicados pelo número 5. Os rotores são cilíndricos e na figura aparecem de perfil. A corrente entra por um conector que está representado bem na parte de baixo do rotor da direita. Esse sinal é então enviado a outro ponto por um fio e o segundo rotor (o do meio) recebe o sinal nessa posição e

novamente envia para uma posição diferente. Finalmente, o terceiro rotor recebe o sinal e repete o processo, passando o sinal elétrico para o refletor (6), que o envia de volta através dos três rotores até que ele saia novamente pelo conector (4), neste caso, através do segundo contato de baixo para cima, o qual corresponde no painel luminoso à letra “D”.

Até aí nada demais. O problema é que a cada toque no teclado, o rotor da direita girava uma posição fazendo com que uma letra diferente fosse acesa caso a letra “A” seja pressionada de novo. E, como na calculadora de Schickard, cada vez que o rotor da direita completava 26 giros, o rotor do meio era girado também e este, por sua vez, após 26 giros fazia girar o da esquerda, gerando com isso  $26^3 = 17.576$  combinações possíveis entre os rotores. Aliando-se isso às diferentes possibilidades de configuração da máquina através de plugues que ligavam o teclado e as lâmpadas aos contatos, chegava-se à ordem de vários trilhões de configurações possíveis, o que era virtualmente impossível de decifrar manualmente.

Porém, essa máquina tinha um ponto fraco. Como ela era baseada na ida e volta do sinal elétrico pelos mesmos rotores, ela nunca poderia criptografar uma letra como ela mesma. Assim, se você visse a letra “A” numa determinada posição em uma mensagem criptografada você saberia que qualquer outra letra poderia estar ali, exceto “A”.

Alan Turing foi contratado pelo governo britânico juntamente com outros cientistas para tentar decifrar as mensagens da Enigma em Bletchley Park. O problema era sério pois todos os dias, à meia-noite, os alemães trocavam as configurações dos plugues da máquina. Assim, se eles levassem mais de 24 horas para adivinhar a configuração atual, o trabalho se tornava inútil, pois neste momento outra disposição já estaria sendo usada.

O ideal seria que conseguissem descobrir a nova configuração o mais rápido possível, pois assim poderiam decifrar um grande

número de mensagens secretas nazistas durante o dia.

Em vez de tentar decifrar manualmente as mensagens, Turing conseguiu apoio para construir uma máquina eletromecânica que, explorando todas as possibilidades por tentativa e erro, poderia em algum momento encontrar a configuração diária utilizada pela Enigma. Essa máquina, mostrada na [Figura 5.4](#), era chama de “Bomba Criptológica”, “Bomba Eletromecânica” ou simplesmente “Bombe”.

O trabalho não ia bem até que Turing divisou um método para explorar a fraqueza da Enigma. Eles poderiam eliminar muitas combinações impossíveis se soubessem que uma determinada frase estaria necessariamente na mensagem. Felizmente para o grupo de Turing, os alemães eram muito formais e suas mensagens seguiam quase sempre um mesmo padrão, iniciando com previsões do tempo pela manhã e terminando praticamente todas com a expressão “Heil Hitler!”.

O método de eliminação que Turing utilizou foi baseado na característica da máquina de nunca cifrar uma letra como ela própria. Uma frase padrão que ele sabia que estaria na frase era chamada de “*crib*”, ou “berço”. Assim, se ele soubesse que a frase “HEILHITLER” estava na mensagem codificada, mas não exatamente onde, só precisava correr a frase sobre a mensagem até encontrar um ponto no qual nenhuma letra do berço ou da mensagem cifrada coincidissem. Ali, provavelmente estaria a frase.

Se a mensagem cifrada fosse “AHJSSKKLAJHSKAL”, eles fariam primeiro um pareamento começando com a primeira letra da mensagem e a primeira letra do berço:

AHJSSKKLAJHSKAL

HEILHITLER



**Figura 5.4: A Bomba Criptológica de Turing.<sup>6</sup>**

Aqui se percebe que essa não pode ser a posição da mensagem (berço) porque existe a coincidência da letra “L”. Passa-se então o berço para a segunda letra da mensagem e repete-se a comparação:

AHJSSKKLAJHSKAL

\_HEILHITLER

Essa também não é uma posição válida porque existe a coincidência entre os “H” nas duas sequências. Passa-se, então, para a terceira letra da mensagem cifrada:

AHJSSKKLAJHSKAL

\_HEILHITLER

Como agora não há coincidências entre as letras é possível que a frase que compõe o berço esteja realmente codificada nesta posição. Se isso for verdade, já se pode saber como a máquina codificava cada uma das letras do berço:

- No instante  $n$ , H era codificado como J.
- No instante  $n+1$ , E era codificado como S.
- No instante  $n+2$ , I era codificado como S.
- No instante  $n+3$ , L era codificado como K.
- E assim por diante.

A partir dessa informação, a Bomba podia ser configurada para encontrar uma configuração que fosse compatível e decifrar qualquer nova mensagem enviada que tivesse sido criptografada pela Enigma com aquela configuração.

## 5.9 ABC: O primeiro computador a válvula – 1939

Por volta de 1937 e após tomar alguns bourbons, o professor de matemática e física do Iowa State College, John V. Atanasoff (Estados Unidos, 1903-1995), concebeu a ideia de um computador binário que funcionasse com válvulas e tivesse uma memória baseada em capacitores. Em 1939, com a ajuda de seu aluno Clifford E. Berry (Estados Unidos, 1918-1963), perito em eletrônica, foi construído o primeiro protótipo – além disso, eles obtiveram financiamento para montar a máquina completa, que viria a ser conhecida como o ABC, ou Atanasoff-Berry Computer.

A máquina construída possuía mais de 300 válvulas e mais de 1,5 km de fios em seu interior. Ela tinha o tamanho de uma mesa. O

original foi desmantelado pela universidade quando resolveram usar o porão, onde estava a máquina, para fazer novas salas de aula. Uma reconstrução pode ser vista na [Figura 5.5](#), entretanto.



**Figura 5.5: ABC – Atanasoff-Berry Computer.<sup>7</sup>**

Apesar de não ter ficado muito conhecido, o ABC é hoje considerado o primeiro computador eletrônico digital efetivamente construído. Porém, ele não era programável: assim como a Máquina Diferencial de Babbage, que era um computador de propósito específico pois apenas calculava polinômios, o ABC foi projetado exclusivamente para calcular sistemas de equações lineares (equações com uma ou mais incógnitas representadas por polinômios de grau 1). Dessa forma, o ABC não era Turing-completo, no sentido de que haveria muitas funções computáveis pela Máquina de Turing que o ABC não seria capaz de computar.

Por algum tempo, o ENIAC, de 1946, foi considerado o primeiro computador eletrônico – pelo menos foi o primeiro a ter uma patente solicitada. Porém, a patente do ENIAC foi questionada por Atanasoff, visto que seus criadores tiveram conhecimento prévio do ABC e utilizaram muitos dos conceitos dele no próprio projeto. A patente do ENIAC só foi concedida em 1964 (quase 20 anos após sua construção) e mesmo assim foi considerada inválida em 1973 por ter utilizado muitos dos conceitos do ABC.

Esse computador foi construído e testado, funcionando razoavelmente como previsto até 1942, mas o mecanismo de armazenamento intermediário de dados baseado em leitura e escrita em cartões de papel não era muito confiável e, em função da guerra, o projeto acabou sendo descontinuado logo depois. Porém, a taxa de erros no armazenamento intermediário levou Atanasoff a propor o uso de dígitos verificadores, técnica que até hoje é usada para assegurar que um valor armazenado não foi indevidamente alterado por problemas técnicos.

Um exemplo de dígito verificador seria, por exemplo, acrescentar a um número um dígito extra que correspondesse às unidades da soma dos demais dígitos. Assim, por exemplo, a representação interna do número 2.387.462 seria 2.387.462-2, onde o dígito verificador 2 é obtido a partir da soma  $2+3+8+7+4+6+2 = 32$ ; no qual 2 é o dígito das unidades de 32. Assim, se por algum erro da máquina o número fosse aleatoriamente modificado para, digamos, 3.387.462-2, o sistema conseguiria detectar que se trata de um erro, pois o dígito verificador é 2, mas a soma dos demais dígitos resulta em 33, e, portanto, se o número estivesse correto o dígito verificador deveria ser 3 e não 2. Essa técnica é muito usada em números de documentos, como o CPF, e de agências e contas bancárias.

Uma das grandes inovações do ABC foi sua memória baseada em capacitores, uma tecnologia que ainda hoje é amplamente utilizada nas memórias DRAM (Dynamic Random Access Memory<sup>8</sup>), na qual cada bit é armazenado em um capacitor individual que pode estar

carregado (1) ou descarregado (0). Atanasoff chamava essa memória de “ábaco”.

A memória do ABC era composta por dois tambores, cada um com 1.600 capacitores que giravam sobre os próprios eixos uma vez por segundo. Os capacitores eram organizados em 32 bandas de 50. Assim, a máquina operava com números de 50 bits em ponto fixo. Ela considerava apenas 30 bandas ou números por tambor, ficando dois de reserva para o caso de queima de capacitores. Como os tambores giravam uma vez por segundo, a máquina seria capaz de realizar até 30 adições ou subtrações por segundo.

As funções lógicas e aritméticas eram implementadas através de válvulas arranjadas de forma a comporem portas lógicas. A entrada de dados se dava através de cartões de 80 colunas da IBM e a saída, através de um painel.

## **5.10 Complex Number Computer ou Modelo 1 de Stibitz – 1940**

Conforme vimos anteriormente, após cerca de um ano da construção do seu somador binário ou Modelo K, Stibitz foi convidado pelos executivos da Bell para um projeto em larga escala. Assim, o Complex Number Computer (Computador de Números Complexos), mais tarde renomeado como Modelo 1, foi construído a partir de 1939 – baseado no projeto de Stibitz –, e concluído em 1940. Não era uma máquina Turing-completa, pois apenas fazia operações aritméticas sobre números complexos, mas esteve em operação ininterruptamente até 1949.

Stibitz provavelmente não conhecia o trabalho de Zuse com relês, até porque o mundo estava em guerra nessa época e a comunicação com a Alemanha não devia ser fácil. Porém, ele entrou em contato com Claude Shannon que, como veremos, trabalhou na formalização do uso de circuitos para a construção de portas lógicas e, portanto, de aritmética baseada em sinais elétricos binários.

O Modelo 1 tinha cerca de 450 relês e fazia as quatro operações aritméticas com números complexos de ponto fixo. Ele tinha notação prefixada (o operador era escrito antes dos operandos) e era comandado a partir de um teclado no qual podia-se escrever, por exemplo:

M+.2+i.3+.4-i.5

Onde M representava a multiplicação e os números complexos em seguida são  $+0.2+0.3i$  e  $+0.4-0.5i$ . O resultado era impresso pela máquina.

O Modelo 1 foi um pioneiro também no sentido de ser operável remotamente. De fato, a máquina em si ficava em uma sala à qual quase ninguém tinha acesso, mas podia ser operada através de uma interface com teclado e impressora (semelhante a um teletipo). Em 11 de setembro de 1940, a Sociedade Americana de Matemática se reuniu no Dartmouth College, a alguns quilômetros do prédio onde estava o Modelo 1. Stibitz conseguiu conectar um teletipo no local da reunião ao computador, que pôde ser operado à distância de alguns quilômetros e certamente impressionou a audiência, no meio da qual estavam pessoas que seriam fundamentais para o desenvolvimento da computação nos anos seguintes, como John von Neumann e Norbert Wiener.

Ao longo dos anos, outros modelos a relês foram construídos pela Bell, embora a maior parte dos computadores do tempo da guerra fossem analógicos. Coube a Stibitz também cunhar o termo “digital” para se referir a computadores que operavam com sinais discretos binários ao invés de grandezas físicas. Até então os computadores desse tipo eram chamados de computadores “de pulso”, mas Stibitz achava o termo pouco adequado.

## 5.11 Claude Shannon – 1940

Construir calculadoras ou máquinas lógicas com circuitos elétricos baseados em relês ou válvulas era coisa que se fazia de forma intuitiva e *ad-hoc* até a publicação da dissertação de mestrado de

Claude Shannon (1916-2001), do MIT, em 1940 – considerada pelos estudiosos do tema a mais importante dissertação de mestrado de todo o século XX.

Esse trabalho simplesmente sistematizou a eletrônica para a construção de computadores ao relacionar circuitos elétricos com a álgebra de Boole. A partir da sistematização de Shannon, não só se poderia construir qualquer circuito que representasse uma função lógica ou aritmética, mas também se poderia aplicar regras da álgebra booleana para simplificar circuitos, obtendo as melhores configurações possíveis para representar essas mesmas operações.

As definições de Shannon poderiam ser executadas tanto por relês quanto por válvulas, não fazia diferença. Ele simplesmente definia que um circuito é um contato entre dois pontos  $a$  e  $b$  que pode estar aberto ou fechado. Um circuito está aberto se ele não permite passagem de eletricidade (resistência infinita) e fechado se permite a passagem da eletricidade (resistência zero), não havendo estados intermediários definidos.

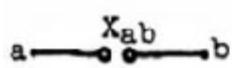


Fig. 1

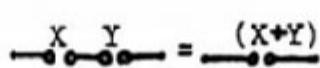


Fig. 2

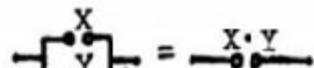


Fig. 3

**Figura 5.6:** Circuito (esquerda), circuitos em série (centro) e circuitos em paralelo (direita).<sup>9</sup>

Na parte central da [Figura 5.6](#), ele define que a soma de dois circuitos ( $X+Y$ ) consiste na colocação deles em série, ou seja, um após o outro. Na parte direita da figura, ele define que o produto de dois circuitos  $X \cdot Y$  é a sua colocação em paralelo, ou seja, a eletricidade pode seguir tanto por um quanto pelo outro, de forma independente.

Para Shannon, um circuito fechado, no qual passa eletricidade, representa o valor booleano 0 e um circuito aberto, no qual não passa eletricidade, representa o valor booleano 1. A partir dessas definições, ele passa a apresentar alguns postulados (afirmações) baseados na lógica booleana e que valem para os tipos de circuitos definidos:

- $0 \cdot 0 = 0$  Um circuito fechado colocado em paralelo com outro circuito fechado equivale a um circuito fechado.
- $1 + 1 = 1$  Um circuito aberto colocado em série com outro circuito aberto equivale a um circuito aberto.
- $1 + 0 = 0 + 1 = 1$  Um circuito aberto em série com outro fechado, em qualquer ordem, equivale a um circuito aberto.
- $0 \cdot 1 = 1 \cdot 0 = 0$  Um circuito fechado em paralelo com outro aberto equivale a um circuito fechado.
- $0 + 0 = 0$  Um circuito fechado em série com outro fechado é um circuito fechado.
- $1 \cdot 1 = 1$  Um circuito aberto em paralelo com outro aberto é um circuito aberto.

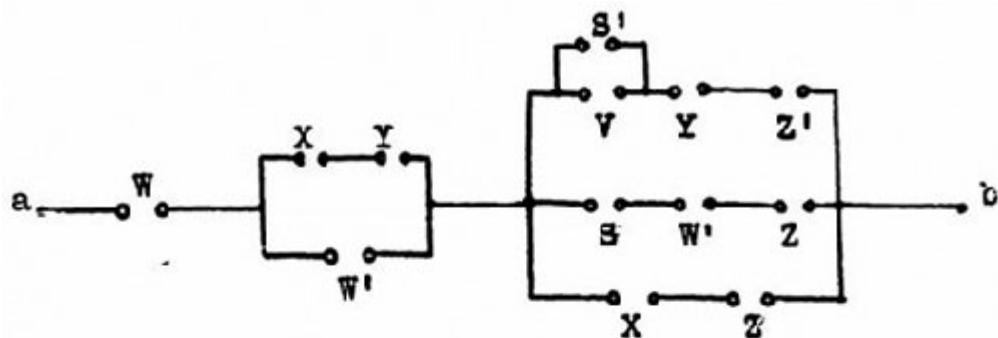
Shannon demonstrou que essas definições eram suficientes para realizar qualquer função lógica booleana e mesmo operações matemáticas.

Ele definiu também um operador de negação  $X'$ , de forma que  $X'$  representa um circuito que se encontra em estado inverso a  $X$ ; ou seja, se  $X$  está aberto,  $X'$  está fechado e se  $X$  está fechado,  $X'$  está aberto. Não era difícil construir um circuito desses com os materiais disponíveis na época de Shannon. Em função dessa operação, ele adicionou novos postulados:

- $X + X' = 1$  Um circuito em série com sua negação equivale a um circuito aberto.
- $X \cdot X' = 0$  Um circuito em paralelo com sua negação equivale a um circuito fechado.
- $X'' = X$  A negação da negação de um circuito equivale ao circuito.

A [Figura 5.7](#) mostra como Shannon aplicava as regras de simplificação da lógica booleana para simplificar circuitos. O circuito mostrado no alto da ilustração, depois de simplificado, fica igual ao circuito bem mais simples mostrado na parte baixa da figura e em relação às variáveis  $X$ ,  $Y$ ,  $W$ ,  $S$ ,  $V$  e  $Z$  ambos os circuitos

representam exatamente a mesma função, ou seja, são equivalentes.



$$\begin{aligned}
 X_{ab} &= w + w'(x+y) + (x+z)(s+w'+z)(z'+y+s'v) \\
 &= w + x + y + (x+z)(s+w'+z)(z'+y+s'v) \\
 &= w + x + y + z(z'+s'v)
 \end{aligned}$$

$$\begin{aligned}
 X_{ab} &= w + x + y + zz' + z's'v \\
 &= w + x + y + z's'v
 \end{aligned}$$

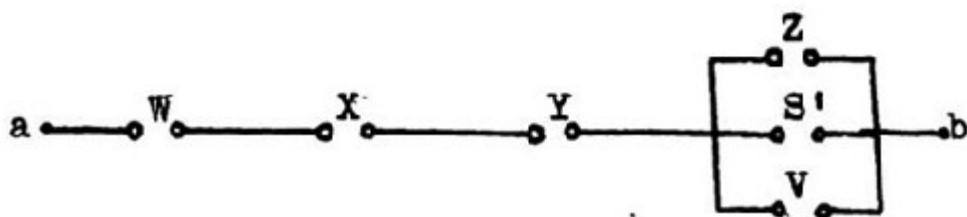


Figura 5.7: Processo de simplificação de circuitos.<sup>10</sup>

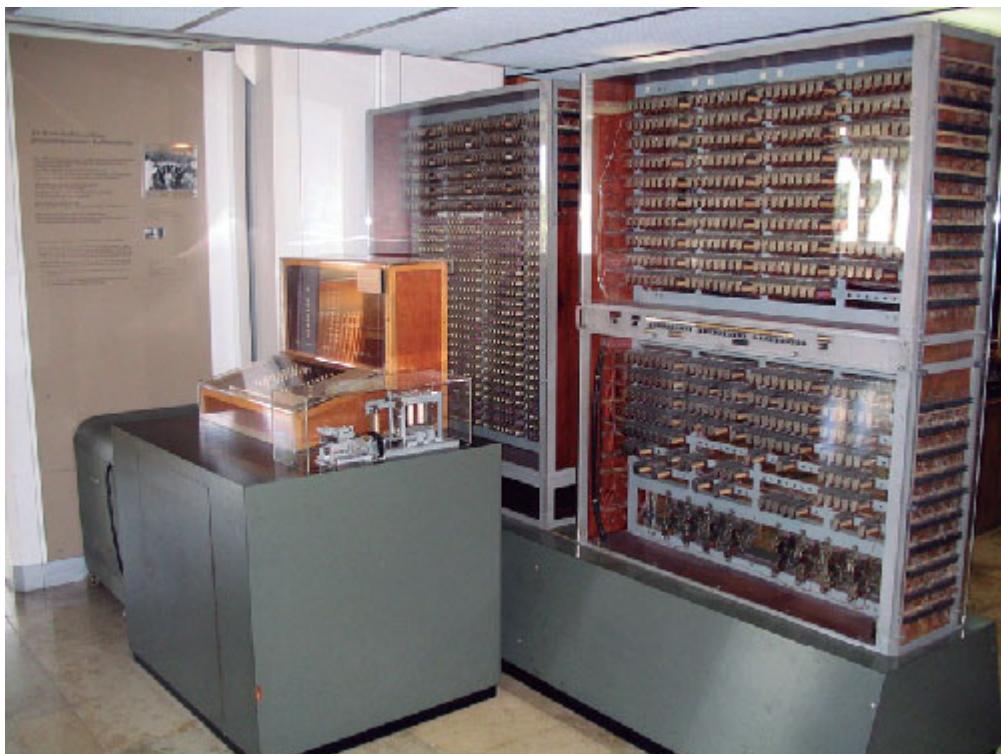
Na parte final da dissertação de apenas 69 páginas, incluindo as seis referências bibliográficas, ele apresenta ainda alguns exemplos de circuitos construídos a partir dessa nova lógica:

- Um circuito que deve ser 1 somente se uma, três ou as quatro variáveis dentre X, Y, Z e W forem 1.

- Uma fechadura eletrônica com cinco botões que só se abre se eles forem pressionados na ordem certa, que é predefinida. Se uma sequência diferente da correta for tentada, um alarme é soado.
- Um circuito de votação eletrônica.
- Uma máquina que imprime números primos entre 1 e 100.000.000 e apresenta o menor divisor para os números que não são primos.

## 5.12 Z3 – 1941

Após o sucesso da demonstração do Z2, Konrad Zuse conseguiu fundos para desenvolver o Z3, um computador totalmente construído com relês. Embora o trabalho de Zuse tenha sido amplamente ignorado fora da Alemanha até recentemente, o Z3 é considerado hoje por muitos como o primeiro computador digital totalmente automático e programável já construído.



**Figura 5.8: O Z3 reconstruído no Deutshes Museum em Munique.<sup>11</sup>**

Tanto Zuse quanto seus amigos pediram ao governo alemão que ele fosse poupado de servir ao exército, de modo que conseguisse trabalhar no projeto do computador – que poderia inclusive servir ao esforço de guerra do país. Entretanto, os pedidos não encontraram respaldo pois os burocratas de plantão não viram o potencial uso de suas máquinas. Assim, Zuse teve que servir ao exército, mas felizmente, por pouco tempo e sem lutar no front entre 1939 e 1940. Quando retornou ao seu estúdio para continuar o desenvolvimento do Z3, ele teve grande dificuldade para conseguir ajudantes visto que muitos morriam na guerra. Assim, ele buscou em escolas por pessoas com deficiência física e conhecimentos de matemática para ajudá-lo.

Desenvolvido a partir de 1939 até ficar pronto em 1941, o Z3 tinha 600 relês na unidade de aritmética (equivalente ao “moinho” de Babbage), 1.400 para a memória e mais 400 relês para a unidade de controle. No mais, suas configurações eram bem semelhantes ao Z1 e Z2. Ele era mais rápido, conseguindo operar a até 5,33 Hz.

Além das oito operações disponíveis no Z1 e Z2, o Z3 também incluía uma operação para extrair a raiz quadrada de um número.

Embora o computador não tivesse comandos de decisão, foi demonstrado em 1998 que ele era uma máquina Turing-completa – ou seja, equivalente em poder de processamento a uma Máquina de Turing –, visto que os ausentes comandos de decisão poderiam ser simulados nele através de operações matemáticas.

## **5.13 RDA2: Rockefeller Differential Analyzer II – 1942**

O Rockefeller Differential Analyzer II foi construído por Vannevar Bush no MIT em 1942. Embora não fosse digital, tecnologia que o MIT só começaria a usar em 1950, ele é considerado como um dos computadores mais importantes da Segunda Guerra Mundial pelo

lado norte-americano, visto que essa máquina conseguia calcular tabelas balísticas.

O computador tem este nome porque sua construção foi financiada pela Fundação Rockefeller. Ele operou normalmente até 1954. A operação era significativamente mais simples do que a de outras máquinas similares da época, pois a sua configuração era definida por fita de papel perfurado e relês eletromecânicos, sem a necessidade de desmontar e remontar partes mecânicas, como acontecia com outras máquinas.

Essa máquina pesava 100 toneladas e consistia de 2 mil peças, entre válvulas e relês, 150 motores e mais de 300 quilômetros de fios elétricos. Como era um computador analógico, boa parte de suas peças consistia de engrenagens que ao se movimentaremumas sobre as outras produziam o cálculo das funções.

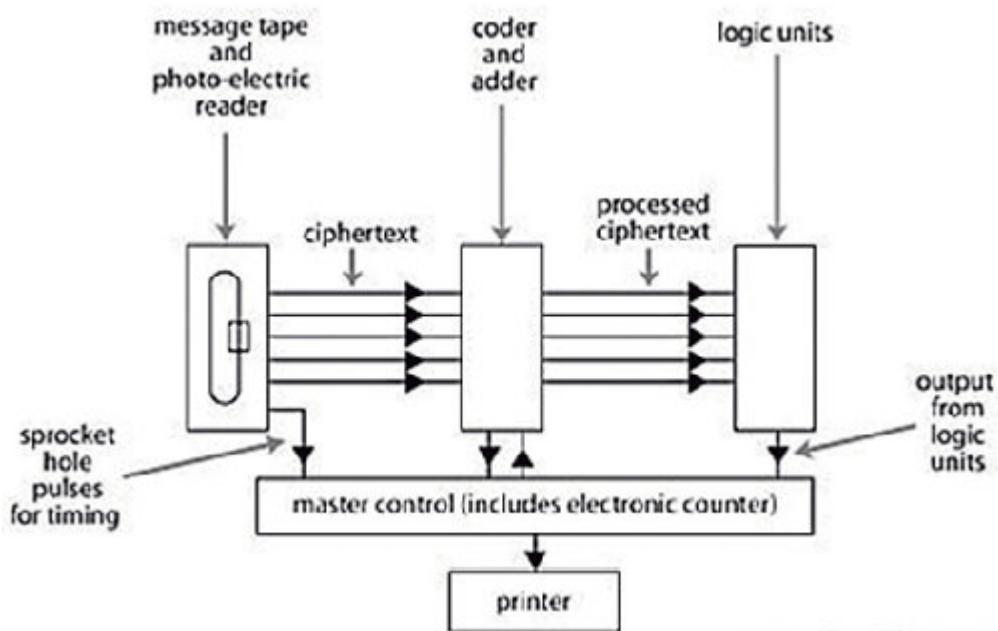
O projeto foi mantido em segredo durante a guerra, mas seu uso foi um sucesso. Talvez a principal aplicação tenha sido a possibilidade de, a partir da observação da rota de um foguete V2 alemão, calcular o ponto de origem de onde ele foi disparado. Dessa forma os Aliados puderam descobrir e destruir os pontos de lançamento dos foguetes.

## 5.14 Colossus Mark I – 1943

Os Colossi foram uma série de computadores desenvolvidos na Inglaterra para quebrar criptografia alemã. O primeiro protótipo, o Colossus Mark I, ficou pronto em dezembro de 1943 e começou a ser usado no início de 1944 em Bletchley Park, local onde Alan Turing trabalhava com a Bomba Criptológica. No total, até o final da guerra, 11 Colossi foram construídos.

O projeto do Colossus foi realizado pelo engenheiro Thomas H. Flowers (Inglaterra, 1905-1998). Alan Turing ajudou em sua execução, e cita-se mesmo que tenha sido ele quem recomendou que Flowers tomasse a frente do projeto do Colossus. Talvez por isso algumas vezes se confunda o Colossus com a Bomba

Criptológica de Turing. No entanto, tratavam-se de duas máquinas que, embora fossem projetadas para quebra de criptografia, trabalhavam com princípios distintos e visavam diferentes sistemas de criptografia. A Bomba de Turing decifrava mensagens da Enigma e o Colossus de Flowers quebrava mensagens cifradas por uma máquina desconhecida, que depois se ficou sabendo ser a máquina Lorenz SZ. A [Figura 5.9](#) apresenta um esquema geral do funcionamento do Colossus, de acordo com Flowers.

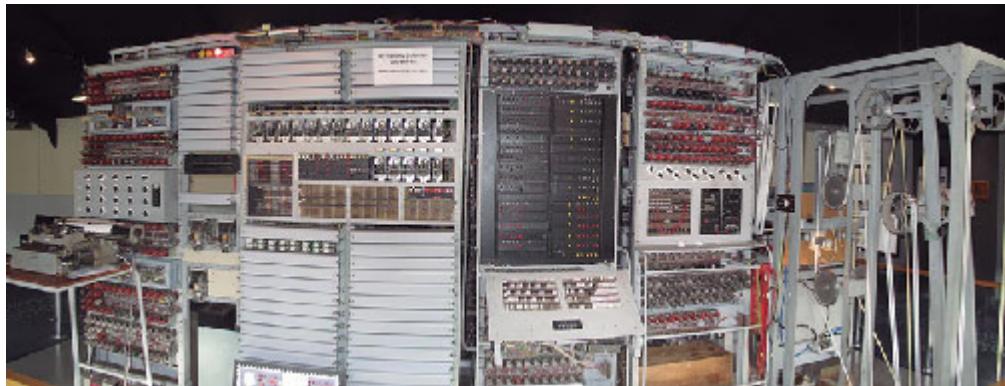


**Figura 5.9: Esquema Lógico do Colossus, realizado por Flowers.<sup>12</sup>**

Após a guerra, o governo inglês ordenou a destruição de todos os Colossi e seus projetos, em um esforço para mantê-los secretos. Restaram apenas partes isoladas dos computadores e algumas notas nos cadernos dos projetistas. Comenta-se também que uma ou outra máquina acabaram sendo poupadadas e continuaram operacionais por mais alguns anos em programas secretos.

Apenas nos anos 1970 o projeto Colossus foi trazido a público e, dessa maneira, ele teve pouca influência direta no desenvolvimento

dos demais computadores do pós-guerra. Mesmo o nome de Flowers ainda é pouco conhecido na história da computação, apesar de sua importante contribuição. Porém, indiretamente, toda a tecnologia e *know-how* desenvolvidos ao longo do projeto acabaram influenciando a construção de outros computadores, visto que muitas pessoas que trabalharam no projeto Colossus acabaram incorporando-se a outras iniciativas, como por exemplo o Manchester Baby, criado por Max H. A. Newman (Reino Unido, 1897-1984), ex-diretor do projeto Colossus. Outro exemplo foi a influência de Turing na construção do Pilot ACE. Uma réplica do Colossus foi construída em 2007 e está em exposição no museu do Bletchley Park ([Figura 5.10](#)).



**Figura 5.10: Colossus reconstruído.<sup>13</sup>**

Cada um dos Colossi foi construído com milhares de válvulas e, embora eles não fossem Turing-completos, eram programáveis através de chaves e plugues, ou seja, ainda não implementavam o conceito de programa armazenado em memória. O uso de válvulas preocupou alguns dos envolvidos, pois era de conhecimento geral que elas costumavam queimar com frequência. Porém, Flowers sabia de sua experiência prévia que elas queimavam devido ao estresse térmico provocado quando eram ligadas frias. Assim, decidiu que as máquinas nunca seriam desligadas, o que aumentaria sua confiabilidade.

Após verificarem que o protótipo Mark I funcionava adequadamente, um modelo mais completo foi construído, o Mark II. Nove unidades do Mark II foram construídas e usadas durante a guerra e uma décima ficou pronta nos últimos dias, mas não chegou a ser aproveitada. O próprio Mark I acabou atualizado e transformado em Mark II, sendo assim possivelmente uma das primeiras operações de upgrade em um computador eletrônico. Enquanto o Mark I tinha 1.600 válvulas, o Mark II tinha 2.400, e era cerca de cinco vezes mais rápido. Além disso, o Mark II era mais fácil de operar do que o protótipo. Essa prática de construir um protótipo, aprender com ele, e depois construir um sistema definitivo acabou sendo uma das mais importantes práticas da indústria de informática moderna.

O processo de otimização da máquina levou em conta também a velocidade com que ela conseguia ler a fita de papel. Foi demonstrado que a partir de 9.700 caracteres por segundo a fita se desintegrava. Assim, foi estabelecido que a leitora de fita funcionaria a uma velocidade mais segura de 5 mil caracteres por segundo. Os Mark II processavam cinco fitas em paralelo, atingindo a velocidade de 25 mil caracteres por segundo.

Como os ingleses não sabiam de que máquina vinham as mensagens cifradas que interceptavam dos alemães, eles chamavam a máquina inimiga de “Tunny”. O funcionamento da Tunny (que, depois soube-se, chamava-se Lorenz) era diferente da Enigma. A Tunny usava um texto embaralhador que era “somado” ao texto que se queria transmitir. Ao contrário da Enigma, que exigia a leitura dos caracteres no painel para posterior transmissão (possivelmente ocasionando erros), a Tunny já transmitia a mensagem cifrada que era recebida do outro lado e impressa sem a criptografia.

O método de “soma” utilizado pela Tunny baseava-se na representação das letras e outros símbolos através de um código binário de 5 bits. Assim, por exemplo, a letra “C” era representada por 01110 (originalmente usava-se “·” no lugar de 0 e “x” no lugar de

1, ficando a representação da letra algo como “·xxx·”), “L” seria representada por 01001 ou “·x·x”, e assim por diante.

A operação de soma de duas letras era feita bit a bit e equivalia ao que modernamente chamamos de operador booleano “XOR” ou “ou-exclusivo”, para o qual dois bits iguais produzem 0, e dois bits diferentes produzem 1:

$$0 \text{ xor } 0 = 0$$

$$0 \text{ xor } 1 = 1$$

$$1 \text{ xor } 0 = 1$$

$$1 \text{ xor } 1 = 0$$

Essa forma de operar com os bits fazia com que a máquina fosse capaz de facilmente codificar e decodificar uma mensagem usando a mesma operação. O XOR tem a seguinte propriedade:  $(x \text{ xor } y) \text{ xor } y = x$ , ou seja, adicionar uma letra a outra e depois adicioná-la novamente, produz a letra original. Por exemplo, se você somar M com N vai obter T (T é a codificação de M usando N como embaralhador). Então para decifrar a mensagem basta somar N novamente a T, já que neste caso o resultado será novamente M. Confira:

Codificando

$$M + N = T$$

$$0 \text{ xor } 0 = 0$$

$$0 \text{ xor } 0 = 0$$

$$1 \text{ xor } 1 = 0$$

$$1 \text{ xor } 1 = 0$$

Decodificando

$$T + N = M$$

$$0 \text{ xor } 0 = 0$$

$$0 \text{ xor } 0 = 0$$

$$0 \text{ xor } 1 = 1$$

$$0 \text{ xor } 1 = 1$$

$$1 \quad \text{xor} \quad 0 \quad = \quad 1 \quad \quad \quad 1 \quad \text{xor} \quad 0 \quad = \quad 1$$

Essa forma de codificação, usando um texto embaralhador em vez de sempre trocar uma letra por outra, faz com que a técnica da identificação da frequência das letras no texto criptografado não seja aplicável, visto que um M em certa posição na mensagem original será somado com uma letra diferente da que seria usada em outra posição. Assim como no caso da Enigma, as letras não são criptografadas sempre da mesma forma.

Imagine que o texto embaralhador (que podia ser qualquer texto, desde que fosse conhecido tanto pelo emissor quanto pelo receptor da mensagem) fosse “WZHI/NR9”. Agora, imagine que queremos codificar a mensagem “COLOSSUS”. As operações de soma, letra por letra, dos dois textos vai produzir “XDIVSDFE”. Agora, somando letra por letra de “XDIVSDFE” novamente com a mesma frase embaralhadora “WZHI/NR9” vai-se produzir novamente “COLOSSUS”.

Um dos problemas com um protótipo anterior ao Mark I (chamado “Heath Robinson”) era que ele trabalhava com duas fitas em paralelo: uma com a mensagem cifrada e outra com a sequência de letras da frase embaralhadora que sabiam que a Tunny era capaz de gerar. A questão era que após rodar por algum tempo, as fitas ficavam fora de sincronia, o que tornava o trabalho inútil e desgastante. Isso foi resolvido com o uso de válvulas para representar as possibilidades de codificação. Assim, bastava ler uma única fita. O que Flowers queria mesmo, originalmente, era substituir a fita da mensagem cifrada por um registro eletrônico representado nas válvulas, mas desistiu quando soube que teria que decifrar mensagens de até 5 mil caracteres (5 kB). A quantidade de

válvulas necessárias para representar tais mensagens com a tecnologia conhecida na época seria absurda.

Hoje, o Colossus Mark I é considerado o primeiro computador totalmente eletrônico e programável (embora de forma limitada) a entrar em funcionamento na história.

## 5.15 IBM ASCC ou Harvard Mark I – 1944

Praticamente ao mesmo tempo em que era projetado e construído o Colossus Mark I na Inglaterra, outros Mark foram produzidos nos Estados Unidos de forma totalmente independente e com objetivos diferentes. O Colossus foi construído com um propósito bem específico em mente: decifrar criptografia. Já o projeto do Harvard Mark I americano nasce sob a proposta de ser um computador de propósito geral.

Seu criador foi o físico da Universidade de Harvard, Howard H. Aiken (Estados Unidos, 1900-1973). Durante seu doutorado, ele se deparou muitas vezes com cálculos de equações não lineares que eram feitos parcialmente a mão e parcialmente com calculadoras eletromecânicas, as quais só faziam as quatro operações aritméticas. Ele chegou à conclusão de que estava na hora e de que já havia os recursos para construir calculadoras totalmente automáticas capazes de calcular a partir de combinações sequenciais das quatro operações aritméticas muitas outras funções mais complexas que eram necessárias para as crescentes demandas da Engenharia, Física e outras ciências. Essas funções seriam, segundo ele, além das aritméticas com números negativos e positivos, expressões com parênteses e colchetes, potências inteiras e fracionárias, logaritmos e antilogaritmos, funções trigonométricas, hiperbólicas e transcendentais. Elas poderiam ser calculadas usando-se séries finitas ou infinitas de operações aritméticas. Por exemplo, o seno de um ângulo  $x$  pode ser calculado através da seguinte série:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

A série em si é infinita. Porém, a quantidade de termos que serão efetivamente usados no cálculo dependerá da precisão que desejamos, ou seja, quanto mais termos adicionarmos à série, maior o número de casas decimais que estarão corretamente representando o seno de  $x$ .

Como elevar um número a uma potência e calcular um fatorial pode ser feito com multiplicações apenas, então se consegue calcular perfeitamente a fórmula acima,  $\sin(x)$ , com as quatro operações – coisa que as calculadoras já faziam há tempos. Porém, elas efetuavam uma operação de cada vez, e um operador humano precisava anotar os resultados e efetuar os cálculos um por um. O que Aiken projetou foi uma máquina que executava sequências dessas operações de forma totalmente automática.

Aiken sabia que a IBM já construía tabuladoras capazes de executar as quatro operações desde o século anterior. Assim, ele propôs a Thomas J. Watson, presidente da IBM, a produção da nova máquina que inicialmente foi chamada de ASCC, Automatic Sequence Controlled Calculator. Aiken queria transformar as máquinas baseadas em cartão perfurado da IBM em calculadoras complexas, adaptadas a problemas mais gerais do que apenas a contabilidade de uma empresa.

Em sua proposta (Aiken, 1937) ele considera que as máquinas IBM da época já eram capazes de calcular automaticamente as seguintes expressões:

- $A+B = F$
- $A-B = F$

- $AB+C = F$
- $AB+C+D = F$
- $A+B+C = F$
- $A-B-C = F$
- $A+B-C = F$

Essas equações eram calculadas da seguinte forma: perfuravam-se até quatro cartões, cada qual com os valores de A, B, C e D, e eles eram colocados na máquina tabuladora. Ela fazia os cálculos e perfurava um cartão com o resultado F. Esse cartão podia então ser usado em outros cálculos subsequentes ou ser levado a uma impressora que o lia e imprimia o número em questão.

A proposta de Aiken foi estender essas tabuladoras para que fizessem internamente o armazenamento dos resultados intermediários dos cálculos de maneira que pudesse ser usados em sequências de cálculos. Isso era muito parecido com a proposta de Babbage, detalhada por Lovelace, 100 anos antes. Aliás, em sua proposta à IBM, Aiken cita Babbage como uma das suas fontes de inspiração.

No ano de 1938 a Universidade de Harvard e a IBM, sob a coordenação de Aiken, fizeram negociações para o projeto que acabou sendo formalizado em 31 de março de 1939. A IBM concordou em construir o ASCC, que mais tarde passou a se chamar Mark I, ou Harvard Mark I, e imaginou que o projeto levaria dois anos. Infelizmente, acabou consumindo cinco, pois o desafio era grande.

Aiken estava ciente de que poderia usar válvulas eletrônicas no seu trabalho, mas possivelmente devido ao alto custo destas (e ao desconhecimento do projeto Colossus), o ASCC acabou sendo construído totalmente com relês eletromecânicos. Mesmo assim, o custo total do projeto em 1944, quando ele se tornou operacional, foi de 200 mil dólares – uma quantia fantástica para a época.

Em abril de 1941, Aiken foi convocado para servir à Marinha americana e não pôde continuar supervisionando pessoalmente o

trabalho no ASCC. Assim, indicou como representante o estudante de Física Robert Campbell, que acabou incumbido da responsabilidade de montar o computador quando este chegou a Harvard, vindo da fábrica da IBM em Endicott. Campbell também rodou no Mark I os primeiros problemas: algumas integrais e tabelas numéricas.

Houve alguns problemas que Campbell teve que resolver, entre eles um que é considerado o primeiro “bug” da história. Literalmente, o termo “bug” já era usado desde o século XIX para descrever defeitos, mas um dia a programadora Grace Hopper (Estados Unidos, 1906-1992), encontrou uma mariposa (um *bug*, “inseto” em inglês) morta dentro do relé 70 do painel F e suas asas impediam o contato ([Figura 5.11](#)). Assim, Grace cunhou o termo “debuggar”, que até hoje significa remover erros de programas.

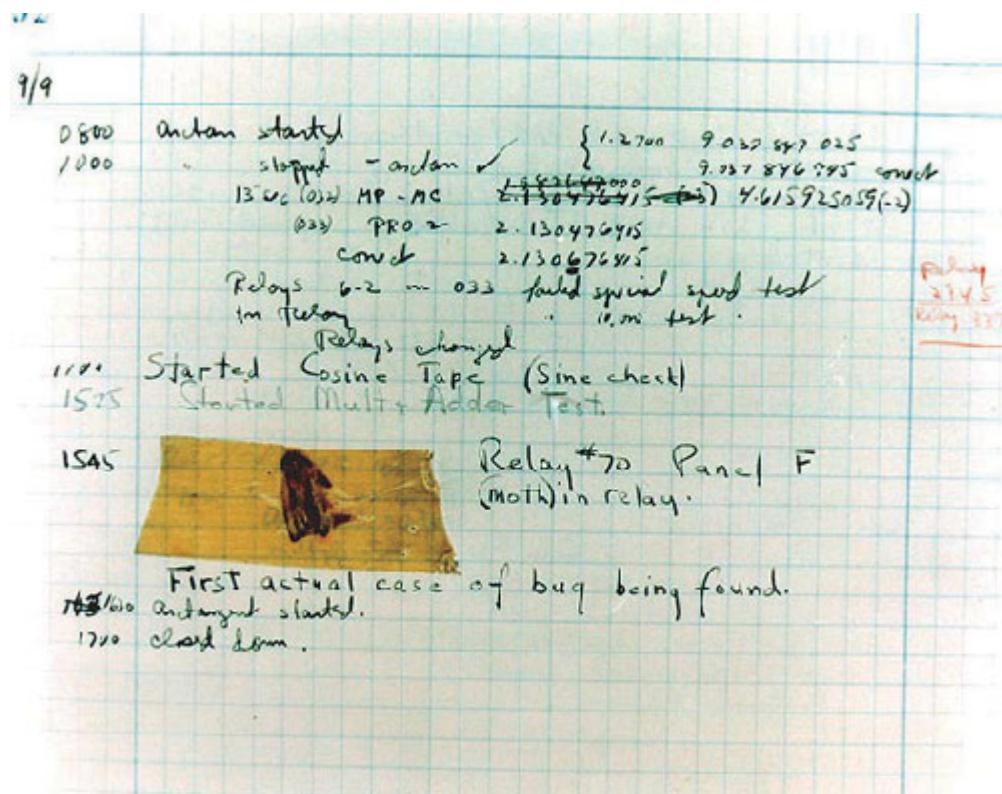
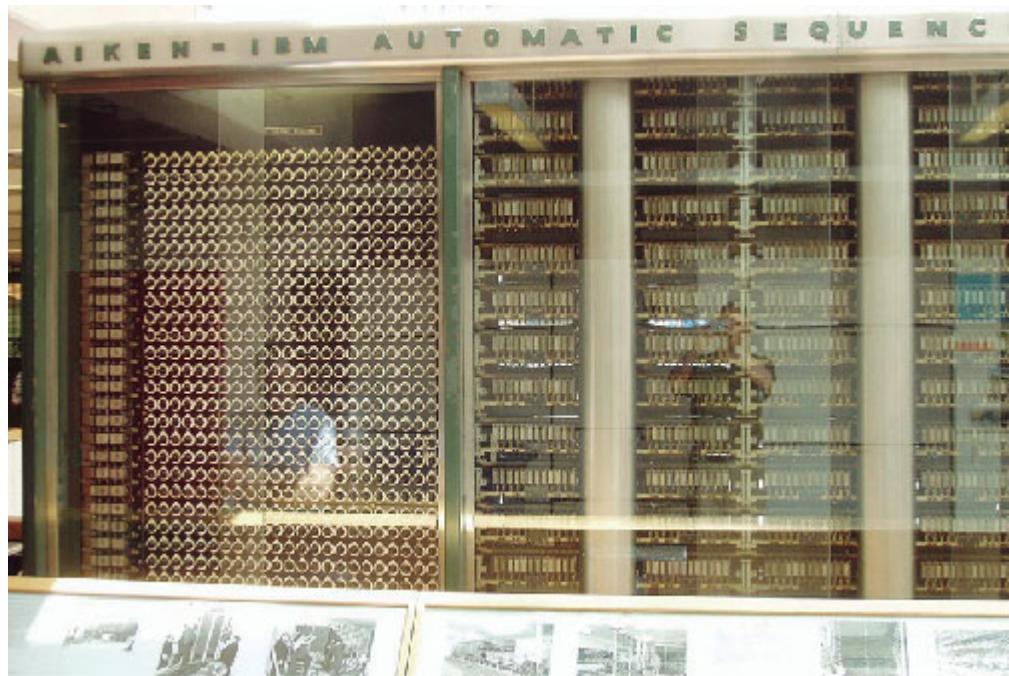


Figura 5.11: O primeiro bug da história.<sup>14</sup>

Aiken, porém, fez uma coisa que enfureceu Thomas Watson, da IBM: apesar de a máquina ter sido construída inteiramente na empresa e de que vários de seus engenheiros projetaram partes físicas que permitiam realizar as ideias de Aiken, ele fez uma chamada para a cerimônia de lançamento do Mark I na qual intitulava-se como o seu único inventor, desprezando completamente a contribuição da IBM. Foi com muita relutância que Watson assistiu à cerimônia de lançamento do Mark I em 7 de agosto de 1944. A partir desse ponto, Aiken e a IBM seguiram por caminhos diferentes.

Entre 1944 e 1945 o Harvard Mark I funcionou ininterruptamente resolvendo os mais complexos problemas, inclusive um conjunto de cálculos de implosões levado por John von Neumann (1903-1957), que mais tarde soube-se tratar de cálculos necessários para o desenvolvimento da bomba atômica.

O Harvard Mark I tinha 2,5 m de altura, 1 m de espessura e 16 m de comprimento. Pesava 5 toneladas e usava cerca de 760 mil partes, com 800 km de fios e 3 milhões de conexões entre fios. Ele possuía 3.500 relês múltiplos com 35 mil contatos e 2.225 contadores. Seu projeto gerou novas tecnologias como subproduto: relês e contadores menores e mais rápidos do que os usados até então foram criados especialmente para o projeto. A [Figura 5.12](#) mostra os dois painéis mais à esquerda da máquina: o da esquerda guarda valores de constantes como  $\pi$  e  $e$ . O da direita é parte da memória. Os demais painéis que não aparecem na figura seriam a unidade de multiplicação e divisão, os contadores funcionais, os interpoladores, o controle de sequência, as máquinas de escrever, as leitoras de cartão e perfuradoras de cartão.



**Figura 5.12: Harvard Mark I.<sup>15</sup>**

Seus registradores guardavam números decimais de 23 dígitos em ponto fixo com uma 24<sup>a</sup> posição extra para o sinal. A [Figura 5.13](#) mostra um detalhe do painel de entrada de dados. Os números 21-23 à esquerda dos botões referenciam-se aos números associados às constantes ali representadas. Por exemplo, o  $\pi$  poderia estar na posição 21. Ao lado, vemos os registradores 24, 23, 22, ... cada qual representando um dígito da constante ali armazenada. Assim, girando os registradores para as posições adequadas poderíamos representar ali diferentes constantes matemáticas necessárias para os cálculos com até 23 dígitos em ponto fixo (lembrando que o 24º registrador era usado para representar o sinal do número).

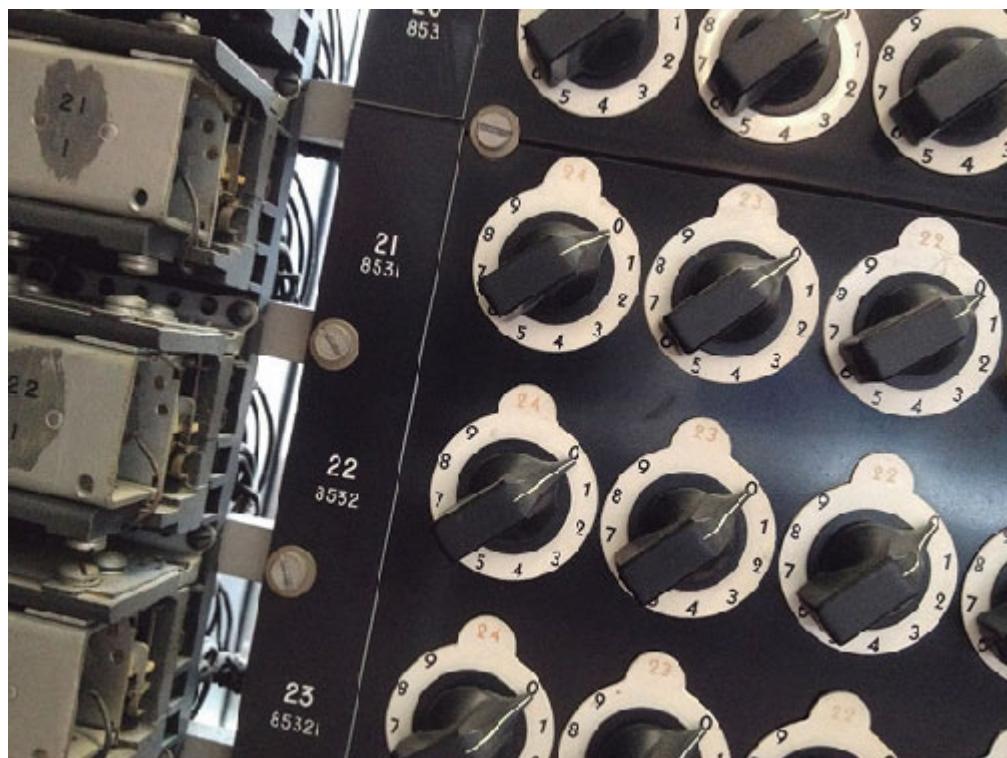


Figura 5.13: Detalhe do painel de entrada de dados.<sup>16</sup>

A máquina era programada por fita de papel perfurada. Ela não tinha comandos de decisão nem de repetição. A falta de comandos de decisão obrigava os programadores a escreverem sequências mais complexas de comandos, mas a repetição era realizada de uma forma bem criativa: eles cortavam a fita de instruções e colavam uma ponta na outra, gerando assim o “laço” ou “loop”, que até hoje é o nome das estruturas de repetição usadas em programação.

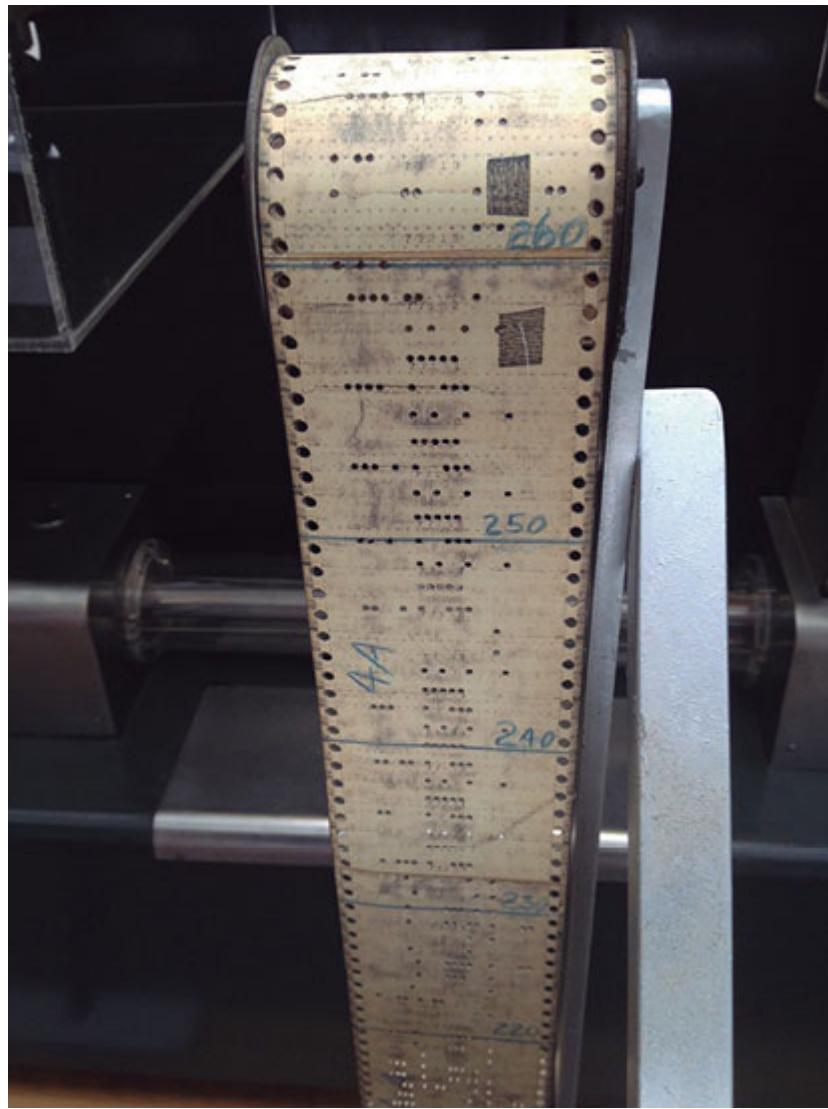


Figura 5.14: Detalhe da fita perfurada do Harvard Mark I.<sup>17</sup>

Assim, para computar a série que calcula o seno de  $x$ , como mostrado acima, eles tinham que carregar o valor de  $x$  em um registrador. Precisavam também de outro registrador para guardar o valor do índice 3, 5, 7, ... . Finalmente, codificariam na fita de papel as seguintes instruções:

- Elevar  $x$  ao índice e armazenar o resultado em V1.
- Calcular o fatorial do índice e armazenar o resultado em V2.
- Subtrair do registrador o valor de V1 dividido por V2.

- Somar 2 ao índice.
- Elevar  $x$  ao índice e armazenar o resultado em V1.
- Calcular o fatorial do índice e armazenar o resultado em V2.
- Somar ao registrador o valor de V1 dividido por V2.
- Somar 2 ao índice.

Se essas operações fossem repetidas uma quantidade suficiente de vezes o resultado seria o seno de  $x$ , dependendo da precisão desejada. A fórmula em questão poderia ser representada assim:

$$-\frac{x^{\text{índice}}}{\text{índice}!} + \frac{x^{\text{índice}+2}}{(\text{índice}+2)!}$$

As instruções seriam então perfuradas em fita e as suas duas pontas unidas em um “loop”. Então, bastaria deixar a máquina rodando a fita até o ponto em que o resultado fosse suficientemente preciso.

Essa máquina tinha uma peculiaridade que nenhuma outra que se saiba teve. Após cada instrução devia ser colocada uma outra que a fizesse continuar processando. Se não houvesse essa instrução especial ela simplesmente parava.

Os primeiros programadores, entre eles Grace Hopper, Dick Bloch e Bob Campbell, costumavam anotar em cadernos essas sequências de comandos para reutilizá-las em outras computações. Por exemplo, se agora você precisasse executar um cálculo que envolvesse senos, já teria pronto o programa necessário para o cálculo do seno. Tempos depois, os programadores se deram conta de que estavam trabalhando com o conceito de sub-rotina, coisa

que Ada Lovelace já havia previsto 100 anos antes e que acabou sendo um dos pilares da programação de computadores.

O Mark I trabalhou ininterruptamente por 14 anos, resolvendo os mais diversos tipos de problemas. Após a guerra foi usado também em um novo curso criado por Aiken na Universidade de Harvard, que viria a se tornar o primeiro curso de Ciência da Computação no mundo. Ele teve três sucessores, Mark II (1947-48), semelhante ao anterior com algumas melhorias, Mark III (1949), com memória de tambor, algumas válvulas e transistores e Mark IV (1952), totalmente eletrônico, com memória de núcleo magnético e transistores.

## 5.16 MEMEX – 1945

O mesmo Vannevar Bush que em 1936 iniciou, mas não terminou, a construção de um computador, publicou em 1945 um artigo que veio a ser um marco na história da computação por sua capacidade de antever e influenciar o que viria futuramente a ser a World Wide Web: “As We May Think”.<sup>18</sup>

Bush foi responsável pela coordenação de mais de 6 mil cientistas americanos durante a Segunda Guerra Mundial. Ao fim a guerra, ele passou a incentivar os cientistas a trabalharem não mais para estender as capacidades de nosso corpo, mas as da mente. Segundo ele, havíamos criado telescópios para aumentar a visão, ferramentas de todo o tipo para aumentar a força, mas nada para aumentar nossa capacidade de pensar ou de acessar o conhecimento disponível.

Ele menciona, por exemplo, que o trabalho de Mendel sobre genética ficou esquecido por quase uma geração porque sua publicação simplesmente não atingiu as poucas pessoas que na época poderiam compreendê-lo e fazê-lo avançar. Bush afirmava que o conhecimento humano já tinha crescido muito e que era necessário criar meios mais adequados para armazená-lo e, principalmente, consultá-lo.

Bush segue descrevendo os avanços tecnológicos dos anos anteriores e especula que, caso a evolução nas técnicas de microfilmes possibilitasse uma redução na proporção de 10 mil para 1 no futuro, seria possível armazenar toda a Encyclopédia Britânica em uma caixa de fósforos e talvez toda a produção literária humana coubesse dentro de um pequeno caminhão.

Ele afirma que a compressão reduz custos. Por exemplo, o material necessário para conter a Britânica microfilmada custaria não mais do que cinco centavos e poderia ser enviado pelo correio a um custo de um centavo.

Mais adiante ele menciona que naquela época se registrava o conhecimento com caneta ou com máquinas de escrever. Mas por que não falar diretamente para a máquina e ela tomar notas como se fosse um estenógrafo? Ele menciona que uma máquina como essa já existia nos laboratórios da Bell. Também lembra que um engenho capaz de falar a partir de texto escrito e sem recorrer a gravações já teria sido apresentado em uma feira mundial naquela época.

Outra visão fantasticamente futurista para 1945, mas natural para os dias de hoje, foi a de um pesquisador em seu laboratório. Com as mãos livres, ele se locomove pelo laboratório fotografando e fazendo comentários que são gravados, juntamente com o registro do tempo, por um gravador especial. Se ele se afastar dali, poderá continuar fotografando e comentando porque estará ligado ao gravador por ondas de rádio.

Finalmente, Bush propõe um dispositivo, o Memex (acrônimo de *memory index*, ou índice de memória), que armazenaria todos os livros, registros e as comunicações, e poderia ser consultado rapidamente ([Figura 5.15](#)).

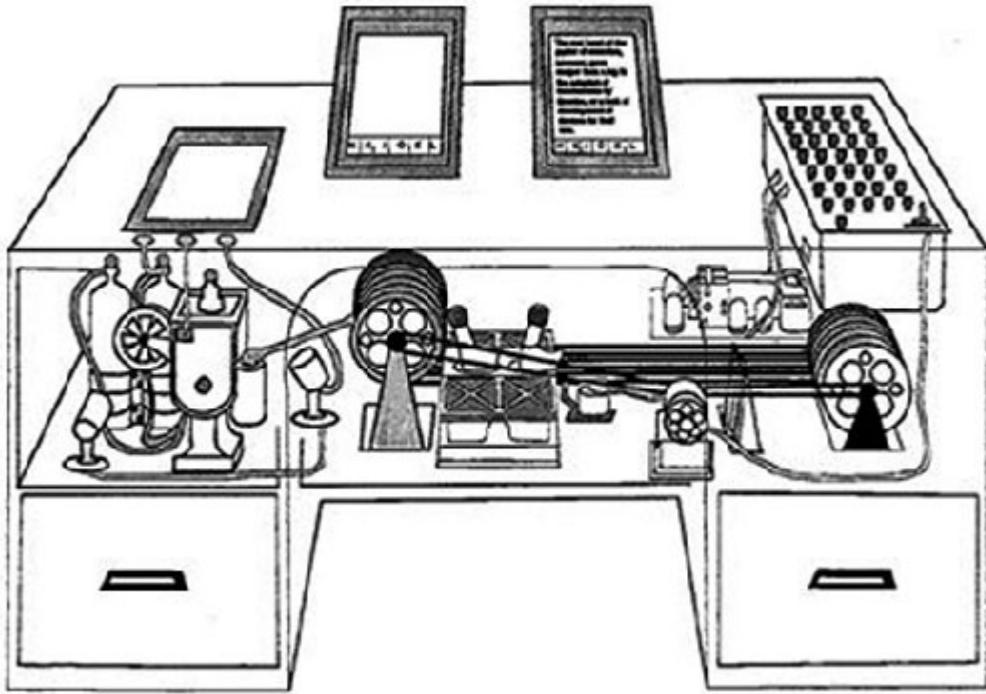


Figura 5.15: Memex.<sup>19</sup>

Ele seria do tamanho de uma mesa, no topo da qual haveria telas translúcidas onde os microfilmes poderiam ser projetados. Para examinar um conteúdo, o usuário deveria teclar seu código de acesso, um mnemônico para ser mais facilmente lembrado, de forma que o usuário não precisasse ficar várias vezes consultando o livro de códigos.

O Memex é considerado um precursor dos sistemas de hipertexto e, de fato, consta que suas ideias tenham influenciado os primeiros criadores da técnica. Segundo Bush, o Memex poderia ser implementado com a tecnologia de microfilmes. Toda a informação seria microfilmada e assim ficaria armazenada dentro da máquina. Novas imagens poderiam ser obtidas pois ela teria um mecanismo de fotografia semelhante a um scanner.

Entretanto, o principal é que ele propunha que as diferentes imagens tivessem ligações entre si, de forma que a partir de uma informação, por associação de ideias, fosse possível obter outras

informações. Assim, as trilhas associativas, que poderiam inclusive ser criadas pelo próprio usuário, permitiriam o acesso à informação de uma forma diferente da sequência na qual ela estivesse armazenada.

Esse mecanismo, porém, ainda permitia apenas a navegação sequencial. O usuário tinha que abrir uma espécie de índice, no qual uma lista de páginas apareceria e assim ele poderia visualizar as páginas na sequência de acordo com o índice em questão. Os modernos sistemas de hipertexto, inclusive Web, são bem mais flexíveis e sofisticados do que isso. Porém, esses sistemas por muito tempo deixaram de implementar uma das características que Bush considerava importante: a habilidade de o usuário criar suas próprias ligações. Além disso, o Memex permitiria que o utilizador colocasse anotações próprias em material preexistente. Essas características só foram disseminadas recentemente com a criação das wikis.

No início de 2014, a DARPA (Defense Advanced Research Projects Agency) anunciou o nascimento do Projeto Memex, que consiste no desenvolvimento de um buscador (Web Crawler) capaz de localizar informação da *deep web*, ou seja, toda a parte da Internet que está escondida e usualmente não é acessível por buscadores como o Google.

## 5.17 ENIAC – 1945

John Mauchly (Estados Unidos, 1907-1980) trabalhava como físico assistente e computador humano na Carnegie Institution's Department of Terrestrial Magnetism. Em 1938, ele submeteu um artigo sobre análise de dados meteorológicos para o *Journal of Terrestrial Magnetism and Atmospheric Electricity*. Seu artigo foi rejeitado com a justificativa de que o período analisado teria sido muito curto. Considerando a complexidade dos cálculos que precisavam ser feitos ele se motivou a procurar meios para realizá-los de forma automática.

Este novo interesse e, em 1941, o conhecimento do trabalho de John Atanasoff levaram-no à Moore School da Universidade da Pensilvânia, onde aprendeu sobre circuitos eletrônicos e iniciou uma parceria muito frutífera com John Presper Eckert (Estados Unidos, 1919-1995).

Quando os Estados Unidos entraram na Segunda Guerra Mundial, o Exército americano encomendou à Moore School o trabalho de calcular trajetórias para mísseis, um serviço de geração de tabelas semelhante ao que era realizado pelos computadores humanos havia gerações. O problema é que as fábricas estavam produzindo muitas novas armas e, para colocá-las em funcionamento, os artilheiros precisavam usar tabelas que estabeleciam os ângulos necessários para configurá-las – até porque muitas vezes eles não eram capazes de visualizar o alvo, o que tornava o tiro direto ou por tentativa e erro impossível.

Uma equipe de 176 pessoas trabalhava nos cálculos usando calculadoras de mesa e computadores analógicos. Ainda assim, levavam mais de um mês para completar todas as tabelas necessárias para disparar uma única arma.

Mauchly e Eckert resolveram investir no projeto de uma máquina que fizesse cálculos automaticamente (já viram essa história antes?). Mauchly escreveu um pequeno memorando para seus superiores, no qual defendia o uso de válvulas eletrônicas para construir um aparelho capaz de computar. O ponto principal era o fato de que ele seria muito mais rápido do que as máquinas mecânicas e eletromecânicas. Mauchly não encontrou respaldo dos superiores, mas o memorando circulou entre seus colegas e, especialmente aqueles que trabalhavam como computadores, ficaram tentados a ver construída uma máquina capaz de fazer em segundos o trabalho de horas. Mauchly viu o Exército como financiador em potencial para sua ideia, e estava certo. Inaugurou-se então o inicialmente secreto “Projeto PX”.

O trabalho no ENIAC, ou Electronic Numerical Integrator and Computer, começou em junho de 1943. O computador, entretanto, só foi terminado um pouco depois do final da guerra em 1945 e, portanto, não chegou a cumprir a tarefa para a qual foi criado.

O ENIAC foi por muito tempo considerado o primeiro computador eletrônico. Porém, apesar de sua importância, ele já não detém este título, visto que o ABC e o Colossus Mark I são mais antigos. O Colossus era um projeto secreto e por isso seu pioneirismo não chegou a público até recentemente. Já o ABC foi reconhecido como primeiro computador após um processo movido por Atanasoff contra Mauchly, que terminou com a vitória de Atanasoff. Durante o extenso processo foi demonstrado que a ideia de um computador eletrônico não provinha de Mauchly, mas que teria sido derivada do trabalho pioneiro de Atanasoff. Mauchly queria a patente do computador, o que lhe foi negado pela Justiça e, a partir daquele momento, o computador como máquina foi sacramentado como de domínio público.

O ENIAC pode não ter sido o primeiro computador a ser construído, mas certamente foi o primeiro a ficar famoso, pois já em 1946 sua existência foi revelada ao grande público. Ele foi anunciado como o “cérebro gigante”. Com o final da guerra e do contrato com o Exército, Eckert e Mauchly queriam ganhar dinheiro e iniciaram uma empresa que fabricaria computadores. Logo, precisavam de publicidade.

O ENIAC era ainda maior do que o Harvard Mark I e sua construção custou quase meio milhão de dólares. Ele pesava 30 toneladas e preenchia uma sala de  $10 \times 15$  m ([Figura 5.16](#)). Continha 17.468 válvulas, 1.500 relês, 70 mil resistores e 10 mil capacitores. A energia necessária para mantê-lo funcionando era de 174 quilowatts (equivalente a quase 2 mil lâmpadas incandescentes). O consumo do ENIAC era tão grande que gerou uma lenda urbana na cidade da Filadélfia: toda vez que o ENIAC era ligado, as luzes da cidade diminuíam.

Esse computador fazia um trabalho de cálculo equivalente a 50 mil pessoas. Uma trajetória simples de míssil era calculada em cerca de 20 horas com uma calculadora de mesa, em 15 minutos com um analisador diferencial ou em 20 segundos com o ENIAC.

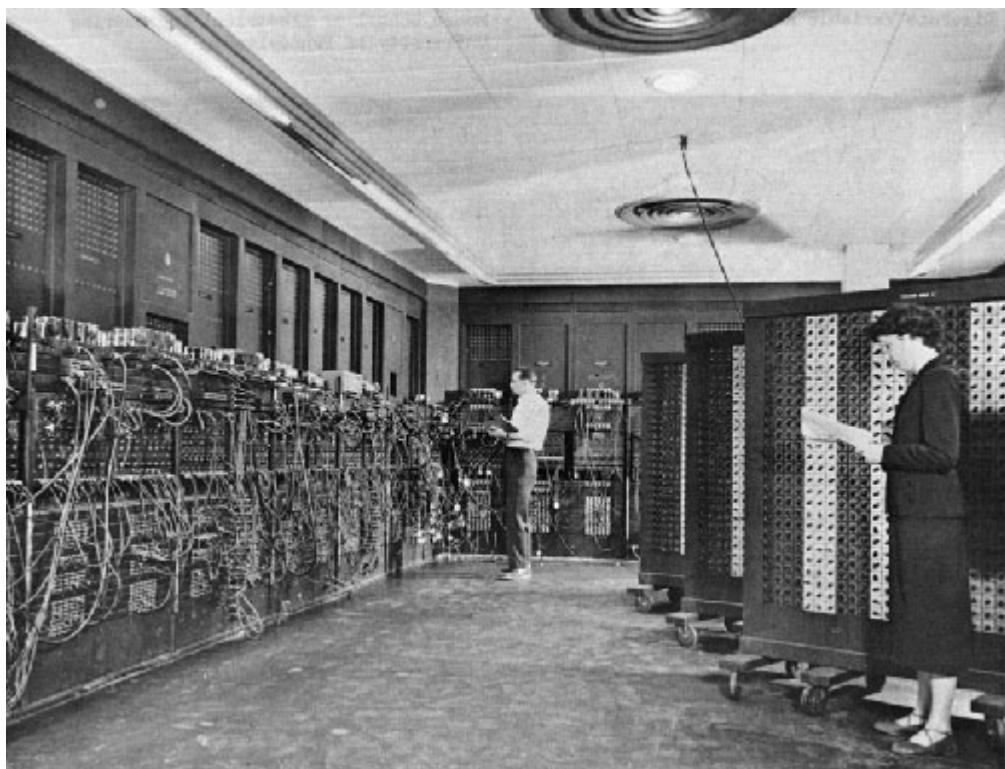


Figura 5.16: ENIAC.<sup>20</sup>

Seu sistema de entrada e saída era composto por uma leitora e uma perfuradora de cartões da IBM. A programação era feita a partir de chaves e plugues. Não havia ainda a noção de programa armazenado, mas era possível programar usando repetição, seleção e sub-rotinas.

A forma como o ENIAC tomava decisões influenciou a definição da linguagem Fortran, anos mais tarde. Em vez de testar uma condição lógica como “ $x > y?$ ” para decidir qual operação executar em seguida, o ENIAC subtraía  $x$  de  $y$ , armazenando o resultado em um

registrador; dependendo do sinal do número armazenado, ele decidia executar uma instrução ou outra.

O trabalho de programação era demorado. Seis mulheres foram escolhidas dentre os melhores computadores humanos para programar o ENIAC, coisa que costumava durar algumas semanas.

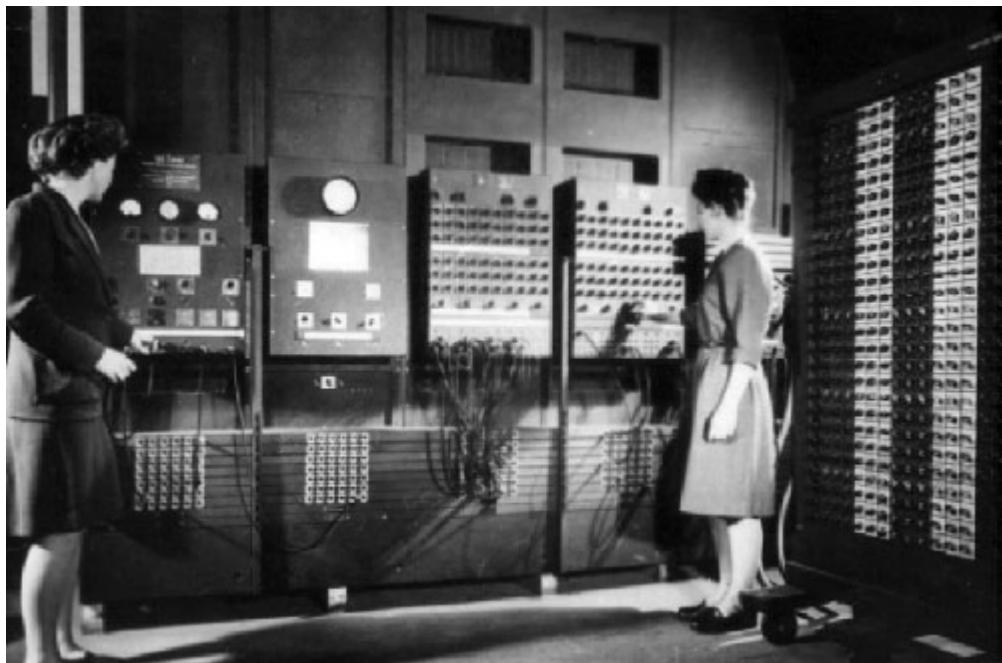


Figura 5.17: As programadoras do ENIAC.<sup>21</sup>

A programação era realizada através dos seguintes passos:

1. O problema a ser resolvido precisava ser representado através de um conjunto de equações.
2. As operações mais complexas relacionadas nessas equações precisavam ser reduzidas aos operadores aritméticos básicos que o ENIAC era capaz de executar. Processo semelhante à programação de Ada Lovelace.
3. Para cada passo da computação era necessário planejar e plugar da forma correta os acumuladores dos quais viriam os dados e onde os resultados seriam armazenados. Além disso,

era necessário definir os pontos de entrada e saída de dados (I/O, ou *input-output*, “entrada e saída” em inglês).

4. Os passos de computação precisavam ser plugados na sequência correta para que cada um fosse executado pelo controle principal no momento certo.

O ENIAC era formado por vários módulos, sendo 20 deles acumuladores, cada um capaz de armazenar um número de 10 dígitos. A arquitetura do ENIAC não era, entretanto, muito eficiente, como depois se descobriu; para representar cada dígito de um número eram necessárias pelo menos 20 válvulas, já que ele trabalhava com base decimal. O uso da base binária e outras formas de representação em computadores posteriores permitiram reduzir muito o número de componentes necessário para representar números.

Um dos problemas enfrentados era o fato de que nos anos 1940 era muito difícil conseguir válvulas de boa qualidade, ainda mais nesta quantidade. Portanto, todo dia uma ou mais válvulas do ENIAC queimavam, especialmente nos momentos em que era ligado. Assim, o computador ficava inoperante praticamente metade do tempo. Consta que os engenheiros conseguiram, com algum esforço, reduzir essa taxa para uma válvula queimada a cada dois dias, o que era mais aceitável. Segundo Eckert, o maior período de funcionamento contínuo do ENIAC foi em 1954, quando ele ficou ligado por mais de cinco dias sem queimar nenhuma válvula.

John von Neumann também usou os serviços do ENIAC para resolver simulações relacionadas ao projeto da bomba de hidrogênio. Diz-se que os cartões usados para as entradas e saídas desta simulação alcançaram a ordem de um milhão de unidades.

Como mencionamos antes, Mauchly e Eckert criaram uma empresa direcionada à produção de computadores: a Eckert and Mauchly Computer Corporation. Seu primeiro computador comercial foi o UNIVAC. Infelizmente, a dupla não era muito boa em negócios e acabou tendo problemas financeiros, pois diversas vezes gastavam

mais para construir suas máquinas do que recebiam por elas. No final, acabaram vendendo a empresa para a Remington.

## 5.18 Plankalkül – 1946

Durante o final da Segunda Guerra Mundial, enquanto britânicos e americanos construíam seus primeiros computadores, Konrad Zuse, na Alemanha continuava cheio de planos e ideias geniais. Porém, não conseguia recursos físicos para colocá-las em prática. Assim, entre os anos 1942 e 1946 ele trabalhou no projeto de uma linguagem de programação, a Plankalkül, que é considerada hoje a primeira linguagem de programação de alto nível concebida.

A diferença entre uma linguagem de programação de alto nível para uma de baixo nível, ou linguagem de máquina, está no tipo de expressões que o programador vai usar. Na linguagem de máquina, que vimos até agora, os comandos disponíveis ao programador são aqueles que a máquina executa diretamente. Por exemplo, para calcular

$$x = \frac{2y + 5w}{k}$$

usando linguagem de máquina, o programador deve escrever a sequência de comandos que faz uma operação de cada vez:

1. Armazene em  $t$  o valor de  $2 \times y$ .
2. Armazene em  $u$  o valor de  $5 \times w$ .
3. Armazene em  $v$  o valor de  $t+u$ .
4. Armazene em  $x$  o valor de  $v/k$ .

Essa era a forma de programação de todos os computadores, desde Babbage até a criação do Fortran em 1957.

Já em uma linguagem de alto nível, o programador simplesmente escreveria o comando:

- Armazene em  $x$  o resultado de  $(2y+5w)/k$ .

A linguagem Plankalkül não chegou a ser implementada até 1975. O próprio Zuse comentou que considerava ela mais um exercício mental do que uma aplicação prática. De fato, a notação seria considerada estranha por programadores modernos, pois a linguagem era bidimensional, assim como a antiga notação lógica de Frege. A [Figura 5.18](#) mostra uma “linha” de comando em Plankalkül que define que dois quadrados são adjacentes se não são o mesmo quadrado e se a distância vertical ou horizontal entre eles é 1 ( $L$ ). Essa linha originalmente fazia parte de um programa para jogar xadrez.

	$V \neq V \wedge  V - V  \leq L \wedge  V - V  \leq L \Rightarrow R\Delta.17$
$V$	0 1 0 1 0 1
$K$	0 0 1 1

**Figura 5.18:** Uma linha de programa em Plankalkül.<sup>22</sup>

A linguagem de Zuse também não teve influência na criação dos primeiros compiladores de linguagens de alto nível nos anos 1960 porque somente foi amplamente publicada e conhecida a partir de 1972.

A linguagem tinha, entre outras, as seguintes características:

- Instruções de atribuição.
- Sub-rotinas.
- Comandos de decisão (condicionais).
- Comandos de repetição (iteração).
- Aritmética de ponto flutuante.
- Arrays (vetores).
- Estruturas de dados hierárquicas.
- Asserções.
- Tratamento de exceções.

Em 1944 Zuse idealizou um dispositivo que poderia ler os comandos dessa linguagem de alto nível e, a partir de sua interpretação, perfurar em uma fita de papel os comandos em linguagem de máquina. Talvez tenha sido a primeira concepção de um compilador na história da humanidade, visto que nessa época ainda nem se pensava em linguagens de alto nível na Inglaterra e Estados Unidos. Zuse, contudo, não chamou a máquina de compilador, mas de *Planfertigungsgerät*, ou seja, “dispositivo de construção de planos”. Ele também usava a palavra “plano” no lugar de “programa”.

## 5.19 Transistor – 1947

Enquanto o mundo construía seus primeiros computadores efetivos com relês e válvulas, a próxima tecnologia para construção dessas máquinas já estava sendo criada. Em dezembro de 1947, os físicos John Bardeen (Estados Unidos, 1908-1991) e Walter Brattain (Estados Unidos, 1902-1987), na Bell Labs, fizeram funcionar o primeiro transistor sólido.

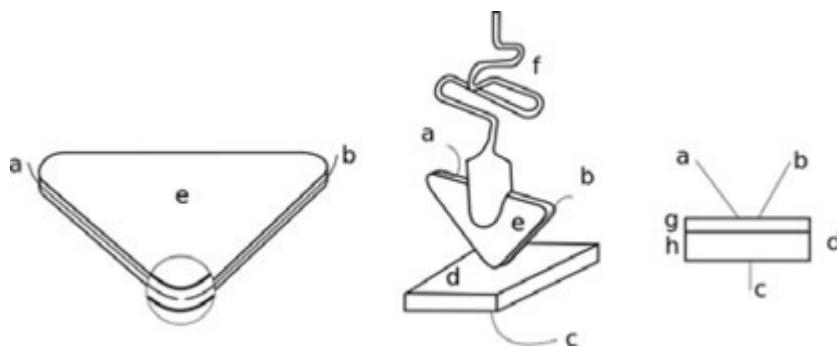
De forma simplificada, o transistor basicamente realiza as mesmas funções da válvula-diodo. Ele tem três terminais, chamados de “emissor”, “coletor” e “base”. Se nenhuma corrente elétrica for aplicada na base, a corrente entre o emissor e o coletor também não flui. Porém, a aplicação de uma pequena corrente na base faz com que uma corrente eventualmente até maior flua entre o emissor e o coletor. Dessa forma, o transistor pode funcionar tanto como um amplificador quanto como um retificador de corrente, exatamente como a válvula-diodo.

No entanto, é a possibilidade de usar esses componentes em grandes quantidades e em combinações adequadas que os torna interessantes para a computação. Como vimos, Claude Shannon, sete anos antes, sistematizou os modelos de criação de circuitos lógicos e aritméticos a partir de contatos do tipo “aberto-fechado”. Isso possibilitava a construção de computadores muito mais poderosos com relês e válvulas. Porém, os relês ainda eram muito

lentos e as válvulas, além de queimarem com frequência, consumiam muita energia. Com o transistor surgia uma nova opção: relês em estado sólido – extremamente rápidos e confiáveis. Eles não evoluíram da noite para o dia, entretanto.

A pesquisa na Bell Labs sobre transistores sólidos iniciou durante a Segunda Guerra devido à necessidade de trabalhar com recepção de micro-ondas (ondas de rádio com frequência extremamente alta). As válvulas de tubo de vácuo simplesmente não davam conta dessa velocidade. Assim, eles começaram a trabalhar com cristais de Germânio, que tem a propriedade de ser isolante exceto se um pequeno fluxo de elétrons for aplicado ao elemento; neste caso, os elétrons ocupam determinados espaços ou “buracos” na estrutura cristalina do germânio e ele passa a ser um condutor em vez de isolante.

Vários experimentos foram feitos sem sucesso, até que fixaram uma fina lâmina de ouro sobre uma cunha de plástico e então a cortaram em duas partes bem na aresta da cunha. Essa aresta foi apoiada sobre um cristal de germânio e dois contatos foram colocados em cada uma das lâminas de ouro. Aplicando-se uma corrente sobre o lado oposto do cristal de germânio, o circuito se fechava e a corrente fluía entre as lâminas de ouro. Sem essa corrente, o contato se abria e a corrente parava de fluir. A [Figura 5.19](#) mostra esquematicamente o processo. Na figura, (a) e (b) são o coletor e emissor ligados às lâminas de ouro; (e) é a cunha; (d) é o cristal de germânio e (c) a corrente base.



**Figura 5.19: O primeiro transistor.<sup>23</sup>**

Esse primeiro tipo de transistor foi chamado de “transistor de ponto de contato”. Outros modelos foram criados depois. Um dos principais avanços foi o uso do silício no lugar do germânio a partir de 1954. O silício é mais fácil de purificar e resiste mais a altas temperaturas, ao passo em que transistores de germânio falhavam a meros 75°C.

Em 1956, Bardeen e Brattain juntamente com William B. Shockley (Reino Unido, 1910-1989) ganharam o Prêmio Nobel de Física “por suas pesquisas com semicondutores e a descoberta do efeito transistor”. Pouco antes disso, os primeiros computadores transistorizados começavam a ser construídos.

## 5.20 ACM - Association for Computing Machinery – 1948

A ACM, ou Association for Computing Machinery é hoje a maior sociedade científica do mundo. Ela reúne pesquisadores, educadores e profissionais da área de computação, e foi fundada em 15 de setembro de 1947 como Eastern Association for Computing Machinery.

A criação da ACM foi resultado de várias conferências sobre máquinas calculadoras que reuniram cientistas e profissionais antes de sua fundação. Entre eles:

- Um simpósio sobre máquinas calculadoras digitais de grande escala realizado na Universidade de Harvard em janeiro de 1947.
- Uma série de seis encontros sobre máquinas calculadoras digitais e analógicas, organizados entre 1946 e 1947 pelo capítulo nova-iorquino do American Institute of Electrical Engineers.
- Seis encontros sobre máquinas computadoras eletrônicas, ocorridos entre março e abril de 1947 no Departamento de Engenharia Elétrica do MIT.

Em janeiro de 1948 a palavra “Eastern” foi retirada do nome da associação, e em 1949 sua constituição foi aprovada pelos membros. O nome “ACM”, apesar de se referir a uma sigla que remete aos primórdios da computação, foi mantido até hoje como era em 1948. Seu propósito, estabelecido em 1947, era: “*...to advance the science, development, construction, and application of the new machinery for computing, reasoning, and other handling of information.*”<sup>24</sup>

Esses objetivos foram reescritos e atualizados ao longo do tempo, embora mantivessem sua essência. Atualmente, são descritos assim: “*The Association is an international scientific and educational organization dedicated to advancing the art, science, engineering, and application of information technology, serving both professional and public interests by fostering the open interchange of information and by promoting the highest professional and ethical standards.*”<sup>25</sup>

Para ser membro profissional da ACM basta ter um diploma de graduação na área de computação ou afim, ou comprovar pelo menos dois anos de atividades em tempo integral em áreas relacionadas à informática. Estudantes de computação também podem ser membros, mas de uma categoria diferente. Atualmente a associação possui mais de 100 mil membros em vários países do mundo.

## 5.21 Z4 – 1948

Em março de 1945, Konrad Zuse conseguiu fugir de Berlim para a cidade de Hinterstein, localizada na Baviera, levando consigo a esposa Gisela, que então estava grávida, e seu novo computador, o Z4, semiacabado. Na Baviera, escondeu o Z4 em um celeiro enquanto obtinha comida em troca de lenha que ele cortava e vendia aos fazendeiros e tropas americanas.

O Z4 só foi finalizado em 1948 ([Figura 5.20](#)), mesmo ano em que foi apresentado ao professor Eduard Stiefel de Zurique, o qual, embora considerando que a máquina baseada em relês era um tanto antiquada (o anúncio público do ENIAC já havia ocorrido dois anos antes), percebeu ser um engenho muito simples e aparentemente poderoso e confiável. Stiefel resolveu comprar uma dessas máquinas para sua instituição. Zuse, então, passou a construir uma nova versão do Z4, adicionando a capacidade de lidar com instruções de decisão e de imprimir resultados em uma máquina de escrever Mercedes ou perfuradora de cartões. O Z4 era realmente muito confiável; especialmente sua memória, que surpreendentemente era formada por milhares de tiras metálicas, parafusos e pinos. Ele era capaz de funcionar sem supervisão mesmo durante a noite, coisa que o ENIAC ainda não fazia.

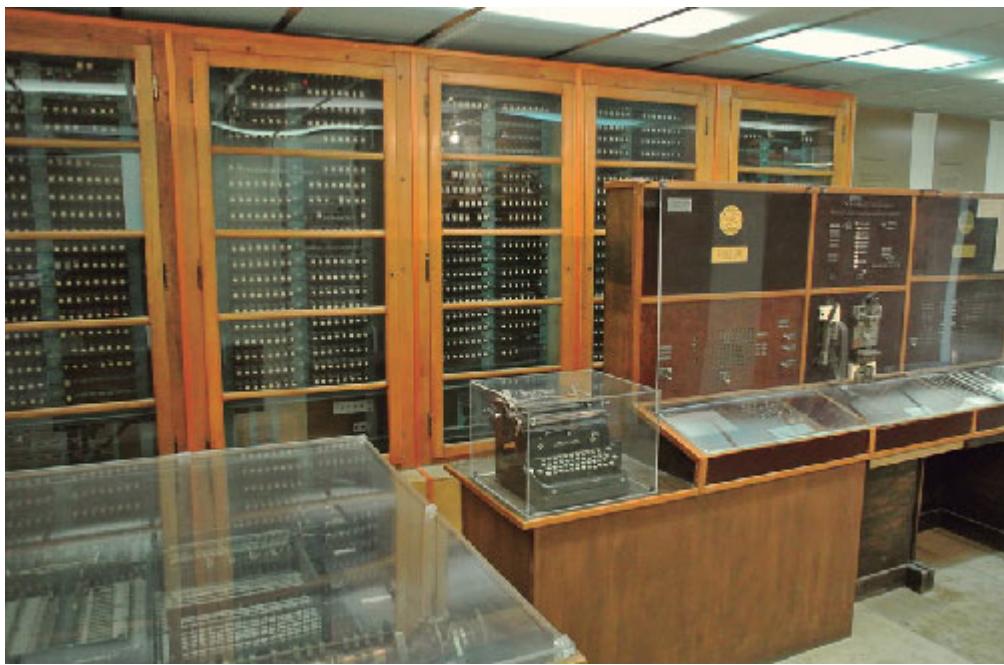


Figura 5.20: Z4 finalizado.<sup>26</sup>

O Z4 continha 2.200 relês. Sua memória era totalmente mecânica e consistia de 64 posições, cada qual com 32 bits. Ele tinha uma unidade de perfuração de fita que permitia programá-lo muito mais facilmente do que os computadores americanos e britânicos da época. Comenta-se que era possível aprender a programá-lo em um curso de apenas três horas. O conjunto de instruções do Z4 incluía:

- Entrada de dados, <-: Permitia a leitura de dados a partir da fita perfurada.
- Saída de dados, ->: Permitia a conversão dos números binários em saídas decimais na máquina de escrever ou na perfuradora de fita.
- Instruções para ler da memória, A n: Por exemplo, A 5, fazia a leitura da posição de memória 5 no registrador R1. Se R1 estivesse ocupado o número seria lido em R2 (lembrando que os computadores Z todos tinham dois registradores para operar com números).

- Instruções para escrever na memória,  $S n$ : Por exemplo,  $S 5$  escrevia o conteúdo do registrador R1 na posição de memória 5.
- Operadores binários:  $+$ ,  $-$ ,  $\times$ ,  $/$ , max e min.
- Operadores unários:  $x^2$ ,  $\text{sqr}(x)$ ,  $1/x$ ,  $|x|$ ,  $\text{sign}(x)$ ,  $x/2$ ,  $2x$ ,  $-x$ ,  $10x$ ,  $3x$ ,  $x/3$ ,  $x/5$ ,  $x/7$ ,  $\pi x$ ,  $x/\pi$ .
- Instruções de comparação,  $x=0$ ,  $x>=0$ ,  $|x|=\infty$ : Essas instruções testam o valor atualmente no registrador R1; se a condição for satisfeita, R1 é setado para 1, senão para -1.
- Instrução de desvio condicional: SPR. Ao encontrar uma instrução SPR, o computador examina o conteúdo do registrador R1. Se contém 1, então ele pula todas as instruções na fita até encontrar a instrução ST. Por outro lado, se R1 contiver -1 não há nenhum efeito.
- Instruções para trocar a leitora de fita. O Z4 tinha duas leitoras de fita perfurada.

Zuse foi muito bem-sucedido financeiramente e fundou uma empresa, a Zuse KG, a partir da qual continuou inovando e construindo novos computadores por muitos anos. Apesar de ter vendido 250 computadores entre 1949 e 1969 e faturado mais de 100 milhões de marcos, não era um administrador muito bom e acabou com dificuldades financeiras. Em 1964 sua empresa foi vendida para a siderúrgica Rheinstahl, mas continuou produzindo computadores até 1969, quando foi adquirida pela Siemens.

## 5.22 Cibernética – 1948

Em 1948, um professor do MIT chamado Norbert Wiener (Estados Unidos, 1894-1964) reinventou o termo “cibernética” ao propor uma ciência que estudasse os fenômenos biológicos, humanos, sociais e mecânicos a partir dos mesmos princípios: feedback, controle e comunicação. Desde então, a cibernética é considerada como a filosofia da tecnologia e Wiener é creditado como fundador dessa

ciência por ter publicado em 1948 o livro *Cybernetics: or Control and Communications in the Animal and the Machine*.<sup>27</sup>

Sob a direção de Wiener, nos anos 1940 foi formado um grupo chamado “Sociedade Teleológica”, que se reunia em conferências organizadas pela Fundação Macy, conhecidas como Conferências Macy. Entre outros, reuniam-se com o grupo Claude Shannon e John von Neumann.

Wiener usou o termo “cibernética” para essa nova forma de se enxergar os sistemas, proveniente do grego “kybernetikē”, que significa “piloto” ou “guia”. Assim, a cibernética seria a teoria que guiaria sistemas orgânicos, mecânicos ou de qualquer outro tipo em seu funcionamento.

Ele considerou que, através da cibernética, quase tudo pode ser considerado um sistema. Sistemas, por sua vez, podem ser quebrados em componentes ou “caixas pretas” com entradas e saídas. Assim, os sistemas poderiam ser estudados e construídos a partir de fluxos de informação, ruído, feedback, estabilidade, controle, e assim por diante. Essas ideias tiveram impacto profundo na maneira como a Ciência da Computação se organizou nos anos seguintes.

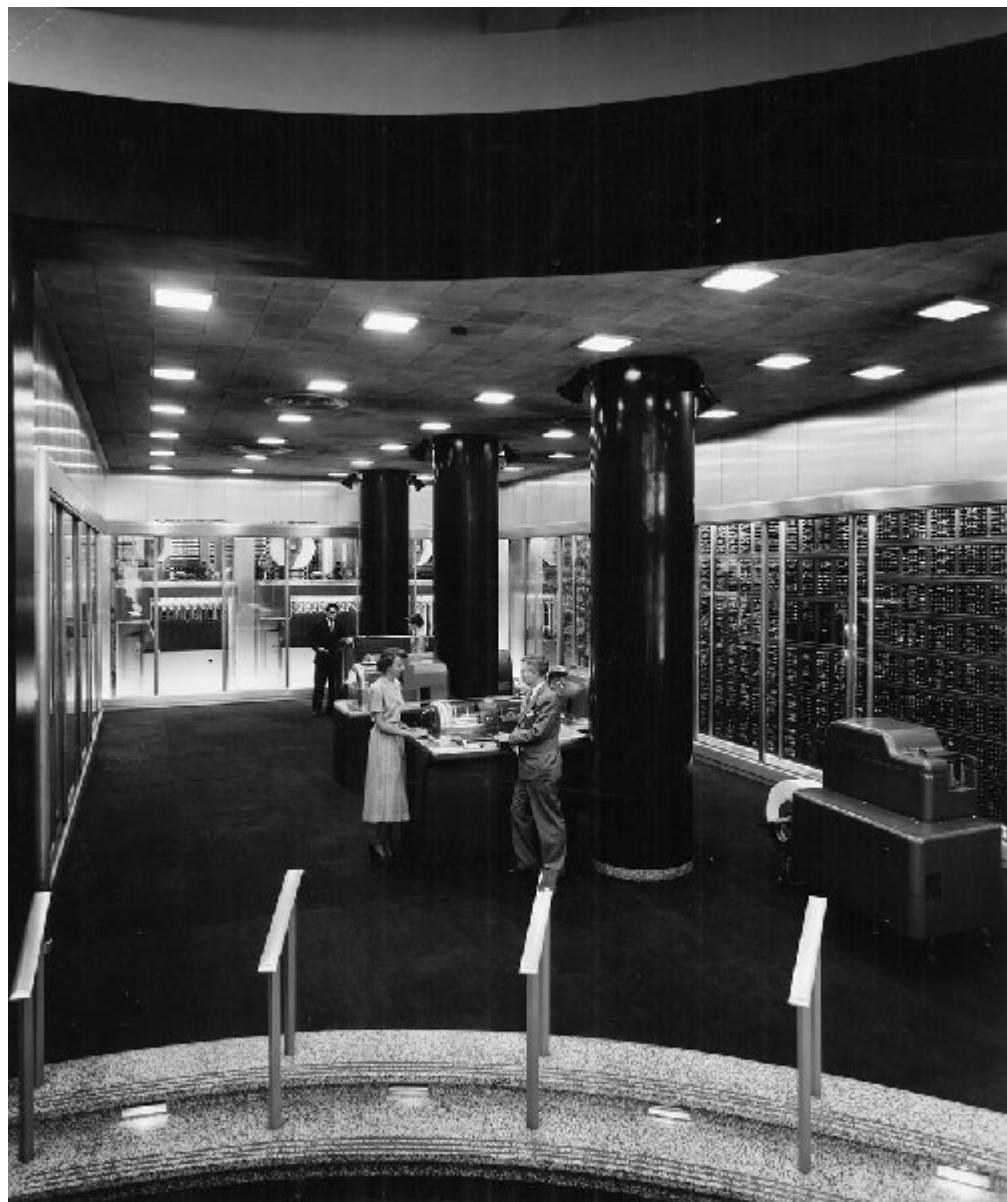
## 5.23 IBM SSEC – 1948

Não custa relembrar que em 1944 o presidente da IBM, John Watson, rompeu com Howard Aiken após o anúncio do Harvard Mark I, no qual Aiken não deu qualquer crédito à IBM pela sua construção. Vimos que Aiken seguiu lançando computadores e comentamos que a IBM seguiu por outro caminho. Esse caminho foi um projeto iniciado em 1944 e apresentado em janeiro de 1948, o IBM Selective Sequence Electronic Calculator (SSEC). Watson quis chamar a máquina de “calculadora” em vez de “computador” porque na época “computador” ainda era o nome de uma profissão. Ele não queria dar a ideia de que as máquinas substituiriam as pessoas,

mas sim que nos auxiliariam fazendo a parte repetitiva e mecânica do trabalho.

O SSEC foi o maior computador eletromecânico já construído e teve grande impacto para a imagem da IBM na área. De fato, ele foi produzido em uma sala na Avenida Madison em Nova York. O SSEC ocupava a parede dos fundos e mais as duas paredes laterais ([Figura 5.21](#)). A quarta parede era toda de vidro e dava diretamente para a rua, de onde as pessoas podiam ver a máquina maravilhosa (e barulhenta). Na figura, pode-se observar que as instalações do SSEC já utilizavam o conceito de “piso erguido”, o que evitava que os operadores ficassem tropeçando nos cabos, que neste caso passavam pelo fundo falso abaixo do piso.

Um de seus programadores foi Edgar F. Codd (Reino Unido, 1923-2003), que depois ficou conhecido por desenvolver as bases do modelo relacional para bancos de dados. Outro de seus programadores foi John Backus (Estados Unidos, 1924-2007), famoso pela construção do primeiro compilador de linguagem de alto nível, o Fortran.



**Figura 5.21: IBM SSEC.<sup>28</sup>**

O SSEC usava cerca de 12.500 válvulas em sua unidade de aritmética com oito registradores. Além disso, mais de 21 mil relês foram usados para a memória da máquina, um sistema muito semelhante ao do ASCC (Harvard Mark I). A máquina fazia adições em 283 microssegundos e multiplicações em 20 milissegundos, sendo, portanto, muito mais rápida que o Harvard Mark I.

Robert Rex Seeber Jr. (Estados Unidos, 1910-1969) foi contratado do grupo de Harvard para ser o designer-chefe da arquitetura do SSEC. Ele incluiu nesse projeto o conceito de “programa armazenado”, que embora na prática não tivesse chegado a ser muito usado no SSEC, passou a ser a base para as arquiteturas de computadores que o sucederam até os dias de hoje. A ideia era que o programa e os dados fossem tratados da mesma forma, armazenados, lidos, modificados etc. Máquinas anteriores armazenavam apenas dados em sua memória, sendo os programas lidos de fitas, cartões ou inseridos diretamente por plugues e interruptores.

Com programas armazenados dentro da memória das máquinas, o processamento destes podia ser muito mais rápido, pois não mais se dependia da velocidade das leitoras de fita. Assim, cada instrução poderia tomar seus dados de qualquer fonte, registradores eletrônicos, memória mecânica ou leitores de fita. Da mesma forma, a instrução também podia armazenar os resultados em registradores, memória, perfuradoras de cartão ou impressoras. Além disso, a instrução poderia em seguida indicar o endereço da próxima instrução, que também poderia estar em memória, registradores ou leitora de fita.

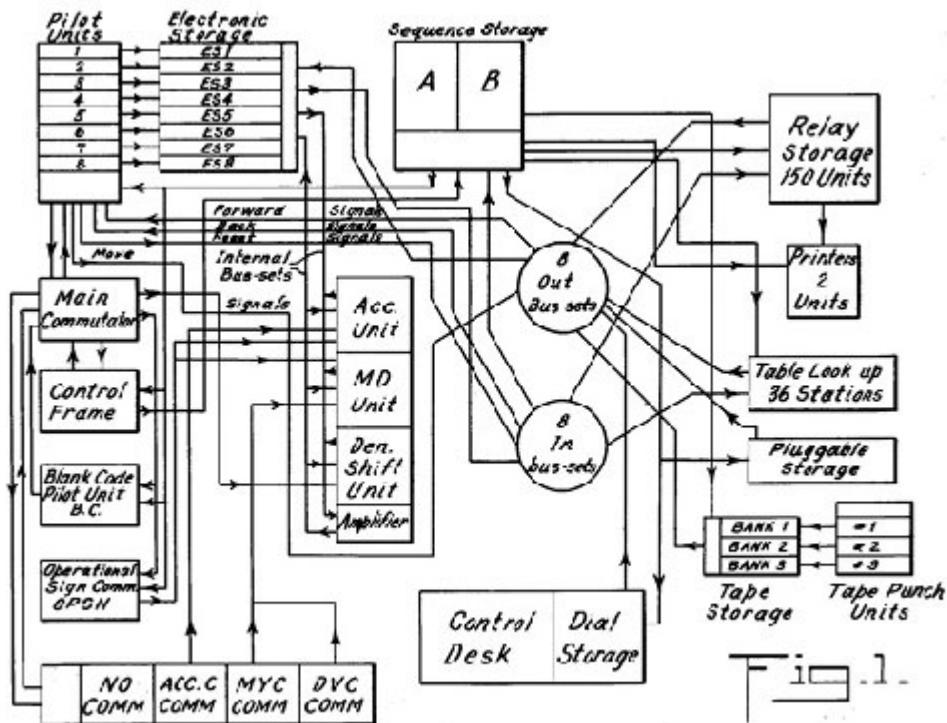


Figura 5.22: Diagrama esquemático do SSEC.<sup>29</sup>

Na prática, porém, apesar dessa potencialidade de rodar programas em memória, a máquina tinha pouquíssima memória e os comandos acabavam sendo executados a partir da leitora de fita, o que mantinha a velocidade média bem abaixo do seu potencial. Laços repetitivos eram feitos da mesma forma que com computadores anteriores: o fim da fita era colado ao seu início.

A primeira aplicação prática do SSEC foi calcular as efemérides, ou movimentos da Lua e dos planetas. O cálculo de cada posição da Lua exigia 11 mil adições, 9 mil multiplicações e 2 mil consultas a tabelas, o que tomava cerca de sete minutos do SSEC. Esse trabalho ocupou a máquina em seus primeiros seis meses de atividade. Depois seus serviços passaram a ser vendidos para outras empresas, das quais a primeira foi a General Electric.

O SSEC também foi um dos primeiros computadores reais a aparecer em um filme, *Caminhe para o Leste* (*Walk East on*

*Beacon*) de 1952, levado às telas como o computador do Prof. Albert Kafer. Ainda no mesmo ano o SSEC, totalmente obsoleto em relação aos novos computadores eletrônicos, foi desmantelado e sua sala passou a ser ocupada por outro projeto.

## 5.24 Manchester SSEM (Baby) – 1948

O Manchester Small-Scale Experimental Machine (SSEM), também conhecido como Manchester Baby, foi construído pela Universidade Victoria de Manchester não para ser um computador usável, mas como protótipo para testar a efetividade dos tubos Williams ([Figura 5.23](#)) como células de memória.

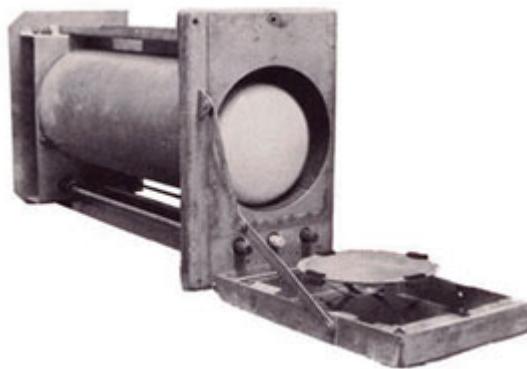


Figura 5.23: Um tubo Williams.<sup>30</sup>

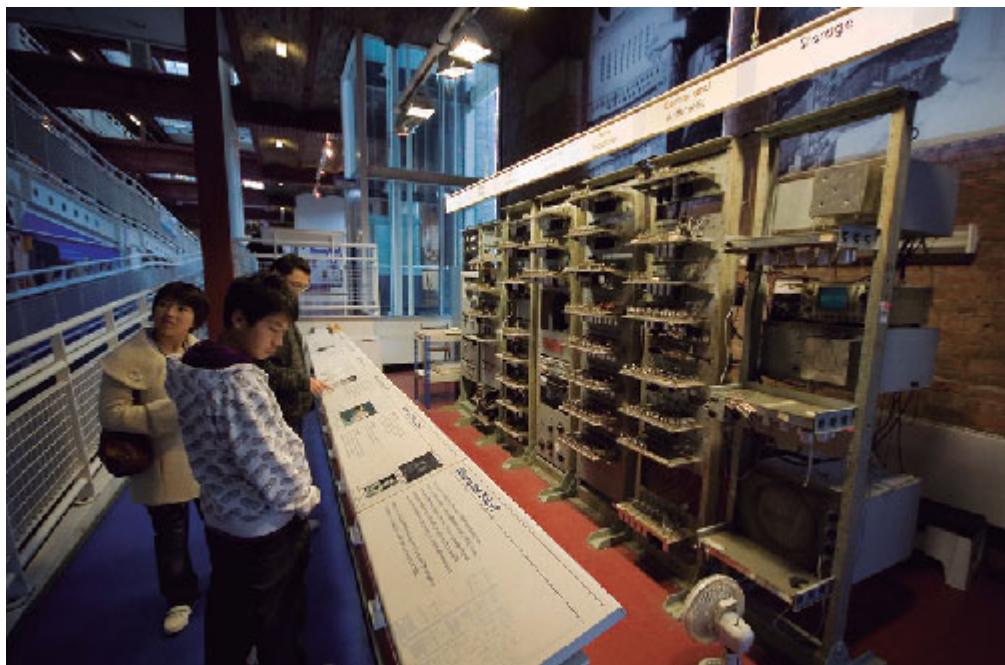
Frederick C. Williams (Reino Unido, 1911-1977), um dos construtores do Baby, foi também o criador de uma forma primitiva de memória de acesso aleatório que levou seu nome: os tubos William. Na verdade, um tubo William nada mais era do que um tubo de raios catódicos, ou CRT. Williams observou que o ponto da tela fosforescente tocado pelo raio ficava positivo, enquanto a área em volta dele ficava negativa, criando-se assim um anel de carga negativa que poderia ser lido por um sensor. Para inverter a carga bastava transmitir um sinal numa área pouco maior do que o ponto, por exemplo, um hífen “-”.

Dessa forma, podia-se armazenar bits de informação numa tela CRT. O mais interessante é que, ao contrário de outras formas de memória desenvolvidas na época, que tinham acesso sequencial (para chegar na posição  $n$  era necessário passar por todas as posições anteriores, como numa fita de papel ou fita magnética), essa forma de memória era de acesso aleatório, ou seja, qualquer posição poderia ser alcançada independentemente das demais.

Porém, a carga magnética só ficava na tela CRT por uma fração de segundo, cerca de 0,2 s. Felizmente era possível usar o próprio sensor que lia as cargas para gerar um pulso de reforço em cada uma das posições de memória. Ou seja, a um intervalo dado de tempo, o computador fazia uma varredura de todas as posições da memória e reenviava um impulso de forma a reforçar a carga positiva ou negativa armazenada em cada espaço.

O Manchester Baby, embora fosse pequeno, tinha todos os elementos de um computador completo, com uma memória formada por um único tubo Williams de 32 posições de 32 bits cada. Outros três tubos serviam para armazenar instruções, como acumulador e saída visual. As únicas operações implementadas em hardware eram a subtração e a negação. As demais operações aritméticas eram executadas em software.

Em junho de 1948, o Baby começou a funcionar. Ele tinha “apenas” 5,2 m de comprimento e 2,2 m de altura; bastante pequeno se comparado a seus contemporâneos. Seu consumo era de somente 3,5 kW; quase nada se comparado aos 174 kW do ENIAC. A [Figura 5.24](#) mostra uma réplica do computador atualmente em exibição no Museu da Ciência e da Indústria em Manchester.



**Figura 5.24:** Réplica do Manchester Baby.<sup>31</sup>

Cada uma das palavras de 32 bits podia conter tanto um número quanto uma instrução. Caso fosse uma instrução, os primeiros bits (0-12) representavam o endereço de memória do operando a ser usado pela instrução; os bits 13-15 especificavam qual a operação a ser executada e os demais 16 bits não tinham uso. Caso um número fosse representado, todos os 32 bits tinham significado e podiam representar números inteiros de  $-2^{31}$  até  $+2^{31}-1$ , ou seja, de -2.147.483.648 até +2.147.483.647.

Os três bits de instrução do Baby permitiam então representar até 8 ( $2^3$ ) instruções. Elas são representadas na [Tabela 5.2](#). O código binário usado era invertido em relação ao que se usa normalmente. Assim, o número 1 seria representado por 100 e não por 001.

**Tabela 5.2: Conjunto de instruções do Manchester Baby**

Binário	Notação original	Mnemônico	Significado
		moderno	
000	S, CI	JMP S	Pula para a instrução no endereço obtido na posição de memória S. Equivale ao “go to” absoluto incondicional.

100	ADD S, CI	JRP S	Pula para a instrução no endereço obtido pela soma do endereço atual com o valor obtido na posição de memória S. Equivale ao “go to” relativo incondicional.
010	-S, C	LDN S	Pega o número da posição de memória S, inverte seu sinal e o carrega no acumulador.
110	c, S	STO S	Armazena o número do acumulador na posição de memória S.
001 ou 101	SUB S	SUB S	Subtrai do acumulador o número guardado na posição de memória S e armazena o resultado no acumulador.
011	Test	CMP	Pula a próxima instrução se o acumulador contiver um número negativo.
111	Stop	STP	Para.

Assim, como o computador implementava em hardware apenas a negação e a subtração, para somar dois números era necessário escrever um programa. O programa se baseava no fato de que  $x+y = -(-x-y)$ . Digamos que  $x$  está na posição de memória 12 e  $y$  na posição 18. A posição 30 será usada para guardar  $t$ , um resultado intermediário. Assim, o programa poderia ser escrito da seguinte forma:

1. -12, C Carrega  $-x$  no acumulador.
2. SUB 18 Subtrai  $y$  do acumulador (o acumulador fica com  $-x-y$ ).
3. C, 30 Armazena o resultado do acumulador ( $-x-y$ ) em  $t$ .
4. -30, C Carrega  $-t$  no acumulador, ou seja,  $-(-x-y)$  ou  $x+y$ .

Os programas eram carregados na memória através de um painel com 32 interruptores e uma tecla de entrada. O Baby não tinha leitora de cartão ou fita perfurada.

O primeiro programa escrito para este computador tinha 17 instruções e foi feito por Tom Kilburn (Reino Unido, 1921-2001). Ele calculava o maior divisor próprio de  $2^{18}$ . Após 52 minutos durante os quais o Baby rodou 3,5 milhões de operações, ele encontrou a resposta correta: 131.072. Seu processamento, assim, tinha uma velocidade de cerca de 1.100 instruções por segundo. Alan Turing rodou o terceiro programa no Baby, para realizar divisões de números longos.

O sucesso do projeto foi relatado em uma carta publicada pela revista *Nature* em setembro de 1948.

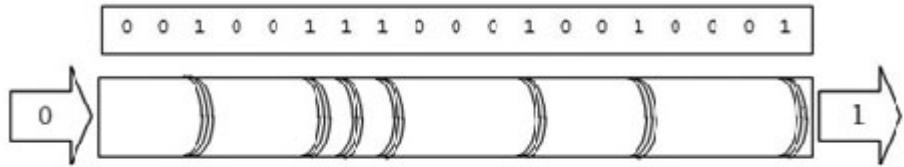
## 5.25 EDVAC e a Arquitetura von Neumann – 1949

Ainda durante a construção do ENIAC, Mauchly e Eckert perceberam que o sistema de memória decimal baseado em flip-flops com válvulas era muito dispendioso. Eram necessárias 20 válvulas para representar um único dígito decimal e, por este motivo, a memória do ENIAC precisava ser muito pequena.

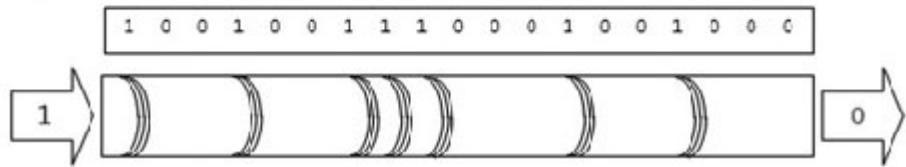
Entretanto, desde 1944 eles tinham planos de construir outro computador. Sua nova máquina usaria um tipo de memória diferente, que eles inventaram a partir de sistemas usados em radares. Tratava-se da memória serial ultrassônica.

A ideia era a seguinte: imagine um tubo longo com algum tipo de preenchimento (no caso, usavam mercúrio). Se você emitir um pulso sonoro em um dos lados do tubo, ele vai se propagar até o lado oposto, onde pode ser captado por um sensor. O pulso levaria algum tempo para viajar do início ao final do tubo, porém. Durante esse tempo, você pode emitir outros pulsos que estarão viajando na mesma velocidade e logo atrás do primeiro. Imagine que você seja capaz de emitir 100 pulsos antes que o primeiro atinja o outro lado do tubo. Neste caso, escolhendo entre as opções de emitir ou não emitir um pulso a cada instante, você estará passando para o outro lado um código binário, no qual pulso emitido significa 1 e pulso não emitido significa 0. Se você adaptar ao sensor no outro lado do tubo um circuito que ative novamente o emissor se um pulso chegar do outro lado, então a mesma sequência de 100 bits ficará indefinidamente circulando por esse tubo; ou seja, você terá uma célula de memória capaz de armazenar 100 bits em som! A [Figura 5.25](#) mostra esquematicamente dois instantes desse processo.

Instante  $n$ :



Instante  $n+1$ :



**Figura 5.25: Memória serial ultrassônica.**

Como se pode ver na figura, no instante  $n$  há um “0” entrando no tubo de mercúrio e um “1” saindo e sendo captado pelo receptor. No instante seguinte ( $n+1$ ), todos os bits se moveram uma posição para a direita e o “1” captado pelo receptor está sendo novamente realimentado no emissor. Esse tipo de memória exige boa sincronização entre os sistemas porque é necessário saber o momento exato em que a palavra está iniciando para fazer a leitura.

Assim, Mauchly e Eckert começaram em 1944 e finalizaram em 1949 o projeto do EDVAC, ou Electronic Discrete Variable Automatic Computer, o qual faria uso desse tipo de memória. Ao contrário do ENIAC, completado em 1945, o EDVAC seria um computador binário e não decimal. Assim como o ENIAC, o EDVAC foi construído e pago pelo Laboratório de Pesquisa Balística do Exército americano.

O uso da memória linear de mercúrio no lugar de válvulas ou relês permitiu que com recursos relativamente escassos fosse construído um computador com uma memória gigantesca para a época: 1.000 palavras de 44 bits, armazenadas em dois conjuntos com 64 tubos de mercúrio com capacidade para armazenar oito palavras cada.

Apesar disso, a unidade de controle e aritmética do EDVAC ainda exigia válvulas e, no caso, 6 mil delas foram usadas, além de 12 mil diodos. Ele foi entregue ao Exército em agosto de 1949, mas apresentou alguns problemas que só foram corrigidos em 1951. Desde então funcionou bem, recebendo inclusive algumas atualizações de periféricos, memória e capacidade de processamento de ponto flutuante. Foi desativado apenas em 1961, substituído por um modelo mais moderno, o BRLESC, ou Ballistic Research Laboratories Electronic Scientific Computer.

Lembra-se que John von Neumann usou os serviços do ENIAC de Mauchly e Eckert para fazer cálculos relacionados ao projeto da bomba de hidrogênio? Pois bem, von Neumann conhecia pessoalmente Alan Turing e sabia de seu artigo de 1936 sobre a máquina automática universal, isto é, uma máquina que podia simular qualquer Máquina de Turing.

Conversando com Mauchly e Eckert sobre o assunto, logo perceberam o potencial dessa ideia e contrataram von Neumann como consultor para ajudá-los a projetar o EDVAC como computador de programa armazenado. O IBM SSEC já tinha essa característica, mas, como vimos na Seção 5.23, ela foi desvalorizada e pouquíssimo explorada. Basicamente, a memória do SSEC era muito pequena para ser gasta com instruções que poderiam estar externamente armazenadas em fita.

O EDVAC teria uma memória muito grande, porém, e programa armazenado.

Assim, o projeto foi feito. von Neumann começou a escrever um relatório sobre essa forma de organizar computadores, que evitava que se precisasse a cada novo programa reconfigurar ou reconstruir a máquina, como no caso dos Colossi e do ENIAC. Esse relatório, o “First Draft of a Report on the EDVAC”<sup>32</sup> escrito por von Neumann, quando ainda estava incompleto, foi distribuído em 1945 por um de seus colegas a dezenas de pesquisadores nos Estados Unidos e Europa.

O trabalho teve uma repercussão tremenda e passou a inspirar praticamente todos os construtores de computadores desde então. O problema foi que somente o nome de von Neumann estava no relatório. Eckert e Mauchly ficaram, com razão, muito aborrecidos. Até hoje, a arquitetura projetada para o EDVAC ([Figura 5.26](#)) é conhecida como “arquitetura von Neumann” e não “arquitetura Mauchly-Eckert-von Neumann”.

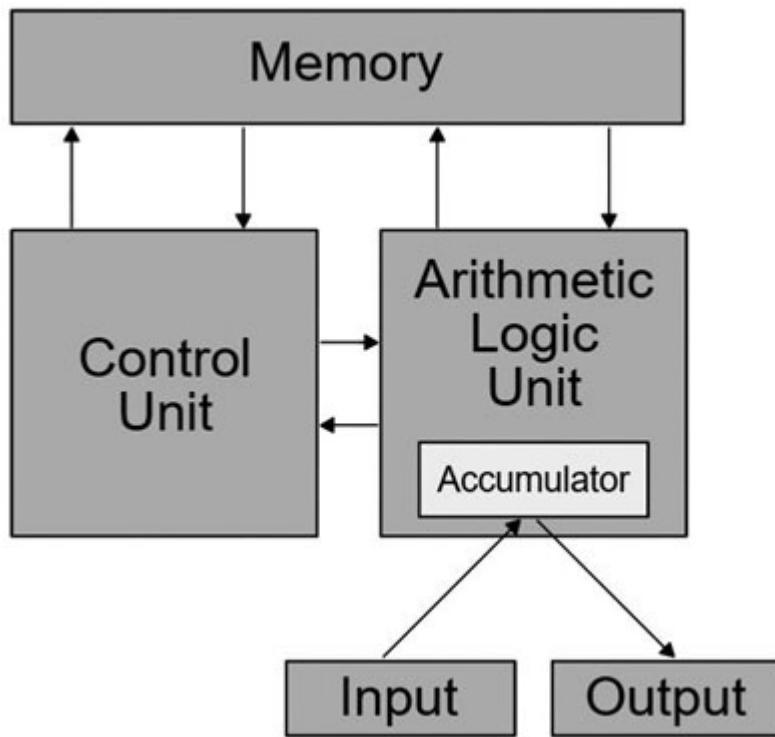


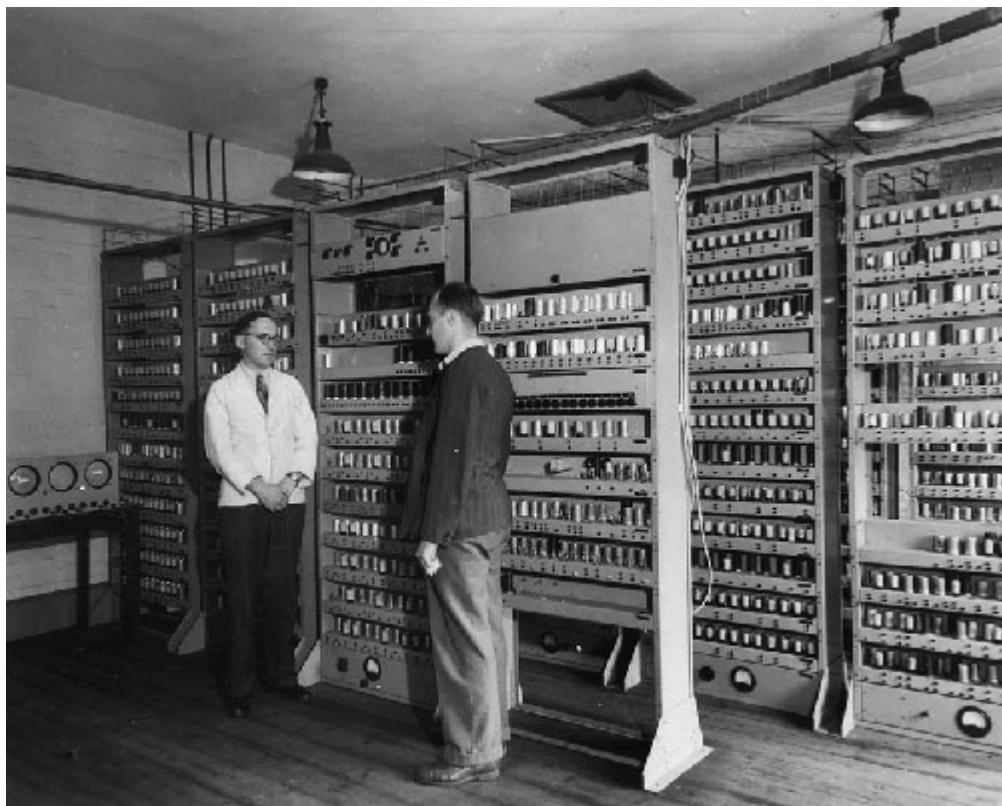
Figura 5.26: Arquitetura von Neumann.<sup>33</sup>

Outra desavença entre eles deveu-se ao fato de que Eckert e Mauchly eram engenheiros e queriam patentear suas invenções para faturar com elas. Já von Neumann era um acadêmico que queria universalizar o conhecimento para que todos pudessem usufruir livremente dessas ideias e construir seus próprios computadores.

## 5.26 EDSAC e o Assembly – 1949

O relatório de von Neumann teve como consequência direta o projeto e construção de um computador de programa armazenado na Inglaterra. O projeto foi chamado de EDSAC, ou Electronic Delay Storage Automatic Calculator ([Figura 5.27](#)). Ele foi construído pela Universidade de Cambridge por uma equipe liderada por Maurice Wilkes (Reino Unido, 1913-2010), sendo iniciado em 1946 e concluído em 1949, pouco depois do EDVAC.

A memória do EDSAC era formada por colunas de mercúrio, como as do EDVAC. Ela tinha inicialmente apenas 512 posições, mas depois foi estendida para 1.024. Cada palavra continha 18 bits, sendo que um não era usado devido a restrições de sincronia da memória. Ele trabalhava tanto com números de uma palavra (17 bits) quanto com números de palavra dupla (34 bits).



**Figura 5.27: Maurice Wilkes e W. Renwick em frente ao EDSAC.<sup>34</sup>**

As instruções consistiam em 5 bits e eram projetadas de forma que o código da instrução correspondesse a uma letra mnemônica. Por exemplo, o código para a instrução ADD era o mesmo código da letra A. Além dos 5 bits de instrução, outros 10 bits eram usados para indicar o endereço de memória onde estava o operando da instrução. Mais 1 bit de controle era usado para indicar se o operando era um número de palavra simples ou palavra dupla.

O uso destas letras mnemônicas para escrever programas no EDSAC deu origem ao que hoje é conhecido como Linguagem Assembly, ou seja, uma linguagem de programação de baixo nível, na qual cada comando corresponde, via de regra, a um comando diretamente executável pela máquina. A vantagem do Assembly (chamado de “Orders” pelo grupo do EDSAC) é que é mais fácil lembrar de letras que possuem relação com o nome do comando a ser executado, por exemplo, A para ADD ou G para GOTO, do que lembrar dos códigos binários para estes comandos.

O EDASC também possuía um conjunto de instruções gravadas em hardware, as “initial orders”, que inicializavam a máquina. Esse conjunto era uma espécie de programa que hoje chamamos de “boot”.

## 5.27 MADM ou Manchester Mark I – 1949

Após o sucesso do protótipo Baby, a Universidade Victoria de Manchester deu continuidade ao projeto, desenvolvendo o Manchester Mark I, também conhecido como MADM, ou Manchester Automatic Digital Machine. O trabalho ocorreu em tempo recorde para a época, de agosto de 1948 a abril de 1949.

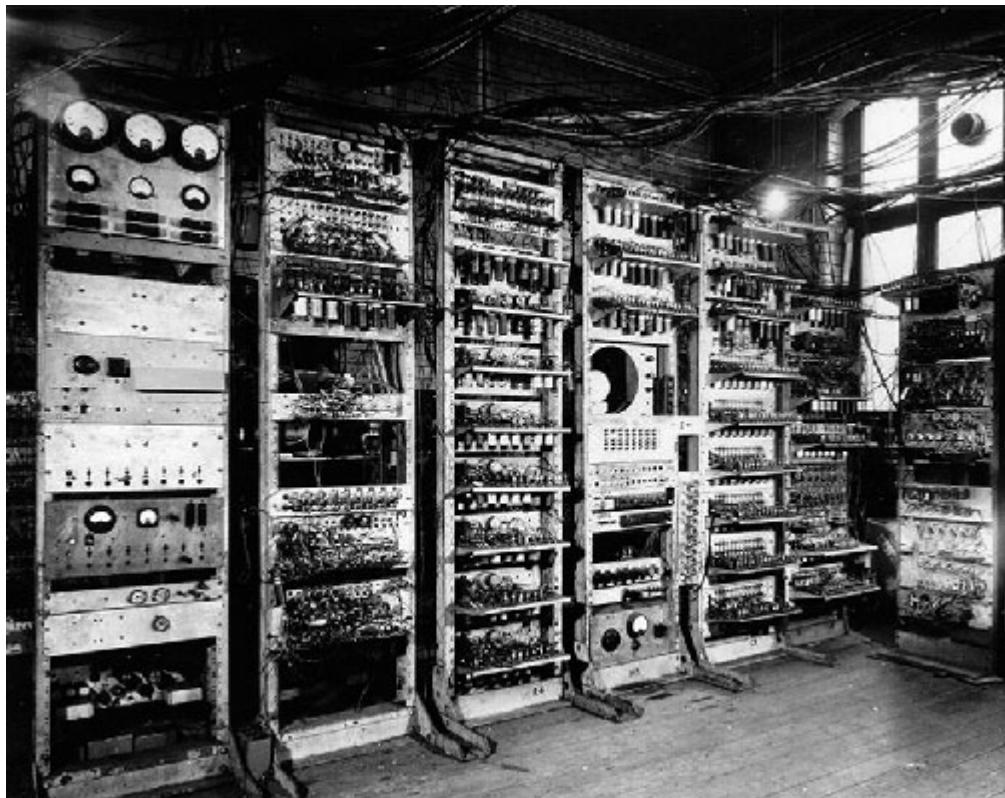


Figura 5.28: Manchester Mark I.<sup>35</sup>

Logo de início, a máquina operou alguns programas complexos sem apresentar erro e foi aclamada pela imprensa britânica como um verdadeiro “cérebro eletrônico”. Essa comparação da máquina com um cérebro gerou reações do chefe do departamento de neurocirurgia da universidade, o qual iniciou um longo debate sobre se um computador eletrônico poderia algum dia ser criativo. Possivelmente esse debate inspirou Alan Turing a propor seu famoso “Teste de Turing”, sobre o qual falaremos na Seção [5.29](#).

A máquina, porém, não ficou em uso por muito tempo, pois à medida que os pesquisadores se acostumavam com ela, foram percebendo possibilidades de melhoria e ela própria tornou-se um protótipo para um computador que viria depois: o Ferranti Mark I – o primeiro computador eletrônico de propósito geral disponível comercialmente (para compra e aluguel). No final de 1950, o MADM

foi desmantelado e substituído por um Ferranti Mark I, sua versão comercial.

Tecnicamente falando, uma das novidades introduzidas por este projeto foi o uso de registradores de índice, que facilitavam a leitura e escrita de informações em arrays na memória.

O tamanho de palavra do MADM era de 40 bits (contra 32 do Baby). A palavra podia conter um número de 40 bits ou duas instruções de 20 bits. A memória principal consistia de dois tubos Williams de dupla densidade especialmente projetados para o MADM. Cada tubo era capaz de armazenar dois arrays de 32 palavras de 40 bits cada.

O conjunto de instruções foi ampliado das sete do Baby para 26 durante o projeto e finalmente 30 na versão final.

## 5.28 Pilot ACE – 1950

Alan Turing também fez um projeto de computador, o que pode não ter sido muito bem visto pelos engenheiros da época, considerando que ele era um matemático se intrometendo na área dos engenheiros. O projeto do ACE (Automatic Computing Engine) foi batizado desta forma em homenagem ao Analytic Engine (Máquina Analítica) de Charles Babbage, a quem Turing reconhecia a influência.

Ele iniciou o projeto logo após encerrar o trabalho com os Colossi em 1945. Infelizmente, como o projeto Colossus era segredo de Estado e a maioria dos engenheiros não podia saber dos progressos obtidos por Turing neste projeto, muitos ficaram céticos em relação à possibilidade de realização do ACE.

Ainda em 1945, Turing publicou o relatório “Proposed Electronic Calculator” pouco depois do texto inacabado de von Neumann sobre o EDVAC. Considera-se hoje que o trabalho de Turing era muito mais completo e menos abstrato do que o de von Neumann, que pouco falava sobre eletrônica e programação. Assim, o trabalho de

Turing, embora menos famoso, foi um verdadeiro marco na criação do conceito de computador com programa armazenado.

Cansado de ver o progresso de seu projeto postergado pelo Laboratório Nacional de Física (NPL), Turing acabou se demitindo. O projeto foi, então, assumido por James H. Wilkinson (Reino Unido, 1919-1986), que o completou em 1950. Na verdade, o que foi finalizado nesse ano foi simplesmente um protótipo do projeto original, que ficou conhecido como Pilot ACE ([Figura 5.29](#)).



**Figura 5.29: Pilot ACE.<sup>36</sup>**

O computador tinha uma memória de tubos de mercúrio com capacidade para 128 palavras de 32 bits. Originalmente ele trabalhava com aritmética de números inteiros, mas como os cálculos começaram rapidamente a extrapolar a capacidade de representação de inteiros com 32 bits, Wilkinson adaptou a máquina para trabalhar com números de ponto flutuante, obtendo assim um

campo muito maior de números que poderia representar. Por algum tempo, este computador foi o mais rápido da Terra.

O Pilot ACE era um protótipo completamente funcional. Assim, acabou sendo usado em aplicações reais, pois havia pouquíssimos recursos computacionais disponíveis no mundo em 1950. Como foi um sucesso, acabou tendo uma versão comercial, o DEUCE (Digital Electronic Universal Computing Engine).

## 5.29 Teste de Turing – 1950

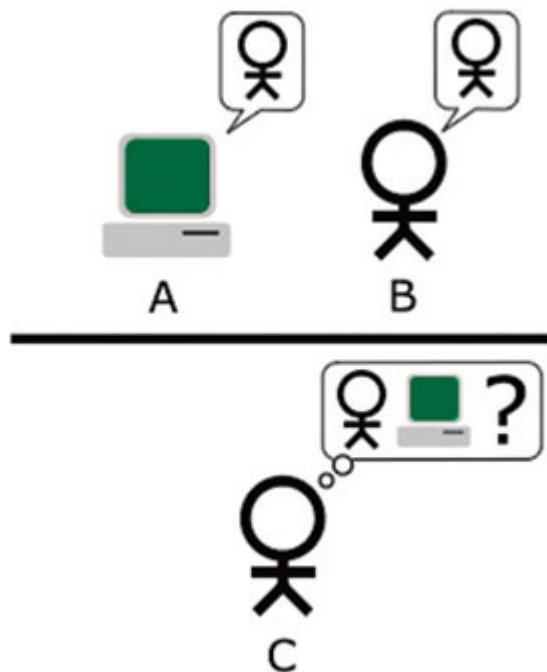
Como mencionamos anteriormente, houve certa celeuma na Inglaterra quando os primeiros computadores foram apresentados como “cérebros eletrônicos”. Esse fato provavelmente levou Alan Turing a pensar sobre o caso e tentar responder à pergunta: “As máquinas podem pensar?” O problema com o qual ele se deparou foi o de definição das palavras “máquina” e “pensar”. Afinal, o cérebro humano é ou não uma máquina? E o que significa pensar?

Dessa forma, esta parecia uma questão inútil de se tentar responder, pois qualquer resposta daria margem a amplas discussões e possivelmente nenhum acordo.

Turing resolveu reformular a questão, perguntando se haveria algum computador capaz de se passar por um ser humano, enganando um juiz no “jogo da imitação”. Esse jogo era bastante comum em festas na época e ocorria da seguinte forma: um homem e uma mulher estariam em um quarto escrevendo em papel respostas a perguntas feitas por um juiz. O juiz não saberia quem era o homem e quem era a mulher; saberia simplesmente que eram os participantes A e B. Assim, fazendo as perguntas que quisesse, o juiz deveria tentar descobrir quem era o homem e quem era a mulher.

A proposta de Turing consistia em substituir um dos participantes por um computador capaz de escrever as respostas. Assim, o trabalho do juiz passaria a ser adivinhar quem era o humano e quem era a máquina ([Figura 5.30](#)). Se o juiz se enganasse cerca de 50%

das vezes é porque a máquina seria indistinguível do humano e, assim, ela teria igualado a nossa capacidade de pensar.



**Figura 5.30: Representação esquemática dos papéis no teste de Turing; o juiz é “C”.  
Ele não tem contato direto com A e B.<sup>37</sup>**

Turing publicou essa ideia em um artigo na revista *Mind* em 1950. Ele previu que os computadores seriam capazes de passar no seu teste por volta do ano 2000, mas isso ainda não aconteceu até hoje. De fato, consta que apenas uma única vez um juiz foi enganado, e isso porque o humano estava fingindo ser uma máquina. O cientista e futurista Ray Kurtzweil (Estados Unidos, 1948) prevê, de forma otimista, que a máquina a ser capaz de passar no Teste de Turing só será construída por volta de 2029.

A história de Alan Turing, infelizmente, teve um desfecho trágico. Em janeiro de 1952 ele iniciou uma relação homossexual com Arnold Murray, de 19 anos. Após um arrombamento na casa de Turing, ele confessou em depoimento à polícia a relação homossexual e foi julgado, pois na época isso era considerado

crime na Inglaterra. Ele foi condenado por indecência e pôde escolher entre a prisão e a castração química. Segundo o próprio Turing, para poder continuar seu trabalho ele recorreu à castração química.

No dia 8 de junho de 1954 ele foi encontrado morto pela sua empregada. Uma maçã parcialmente comida jazia ao lado de sua cama. A causa da morte foi estabelecida como envenenamento por cianeto e, embora a maçã não tenha sido examinada, presumiu-se que foi a forma utilizada para introduzir o veneno no corpo. Sua morte foi então atribuída a suicídio.

Apesar disso, autores como Jack Copeland questionam esta conclusão, afirmindo que uma análise mais detalhada do caso mostra mais indícios de inalação accidental do cianeto do que ingestão proposital. Seu pequeno laboratório contava com experimentos que produziam cianeto gasoso. Além disso, há relatos de que ele estaria de bom humor e cheio de planos para o futuro, o que não condiz com um tipo suicida. Por outro lado, o biógrafo Andrew Hodges sugere que Turing possa ter encenado uma morte aparentemente acidental com o experimento para poupar sua mãe da dor de ter um filho que se suicidou.

### **5.30 Até Aqui...**

Acabamos de ver o surgimento da era da computação eletrônica. Várias máquinas e experimentos foram feitos e a história dos primeiros computadores foi muitas vezes reescrita. Pode-se assumir que o primeiro país a iniciar a construção de um computador foram os Estados Unidos com a máquina de Vannevar Bush, mas o primeiro a concluir um computador foi um alemão, Konrad Zuse – que não tinha apoio nem de seu governo. Atanasoff e Berry construíram um computador a válvula em 1939 que também ficou esquecido porque Mauchly e Eckert não lhes deram crédito quando anunciaram o ENIAC em 1945.

Enquanto na Alemanha Zuse continuava lançando novos modelo de computadores, a Inglaterra também fazia seus desenvolvimentos e por pouco a arquitetura de von Neumann para computadores com programa armazenado não se chamou arquitetura Turing, porque ambos desenvolveram o conceito quase que ao mesmo tempo.

Ao final da década, a tecnologia para construção de computadores já era de domínio público. Os relês começavam a ser substituídos por válvulas, bem mais rápidas, nos processadores, e as memórias basicamente eram tubos Williams ou tubos de mercúrio.

Porém, os computadores ainda eram muito grandes e somente os governos, algumas universidades e poucas grandes empresas podiam pensar em ter algum. Inicialmente na Inglaterra com a Ferranti e logo depois nos Estados Unidos com a Remington Rand e IBM, iniciou-se a exploração comercial da computação. Computadores passaram a ser produzidos em série e podiam ser comprados ou alugados por quem pudesse pagar o preço. Isso nos leva aos anos 1950.

---

<sup>1</sup> Alonzo Church (Estados Unidos, 1903-1995) definiu computabilidade de forma independente e simultânea a Turing através do seu Cálculo Lambda. Posteriormente, Turing demonstrou que o conjunto de funções computáveis pelo Cálculo Lambda coincidia com o da Máquina de Turing, reforçando ainda mais a tese de que esse é, de fato, o conjunto das funções efetivamente computáveis.

<sup>2</sup> Em computação usa-se o termo “palavra” frequentemente para identificar uma posição de memória que é acessível a partir de um endereço. Em um computador de 24 bits como o Z1, uma palavra seria então formada por uma sequência de 24 dígitos “0” ou “1”.

<sup>3</sup> “Horst Zuse and Z1” by Steve Parker - originally posted to Flickr as Z1. Licenced under CC BY 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Horst\\_Zuse\\_and\\_Z1.jpg#/media/File:Horst\\_Zuse\\_and\\_Z1.jpg](https://commons.wikimedia.org/wiki/File:Horst_Zuse_and_Z1.jpg#/media/File:Horst_Zuse_and_Z1.jpg)

<sup>4</sup> “Frans Morsch name on the ICs” by Sergei Frolov, Soviet Digital Electronics Museum, <http://www.leningrad.su/museum/> - Own work. Licenced under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Frans\\_Morsch\\_name\\_on\\_the\\_ICs.jpg#/media/File:Frans\\_Morsch\\_name\\_on\\_the\\_ICs.jpg](https://commons.wikimedia.org/wiki/File:Frans_Morsch_name_on_the_ICs.jpg#/media/File:Frans_Morsch_name_on_the_ICs.jpg)

<sup>5</sup> “Enigma wiring kleur” by MesserWoland - own work by user:HandigeHarry based on previous version based on Image:Enigma wiring kleur.png by Matt Crypto originally nl:Afbeelding:Enigma\_wiring\_kleur.png by nl:User:Drdefcom. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Enigma\\_wiring\\_kleur.svg#/media/File:Enigma\\_wiring\\_kleur.svg](https://commons.wikimedia.org/wiki/File:Enigma_wiring_kleur.svg#/media/File:Enigma_wiring_kleur.svg)

<sup>6</sup> “Bletchley Park Bombe4” by Antoine Taveneaux - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Bletchley\\_Park\\_Bombe4.jpg#/media/File:Bletchley\\_Park\\_Bombe4.jpg](https://commons.wikimedia.org/wiki/File:Bletchley_Park_Bombe4.jpg#/media/File:Bletchley_Park_Bombe4.jpg)

<sup>7</sup> “Atanasoff-Berry Computer” by User:Manop - Own work. Licenced under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Atanasoff-Berry\\_Computer.jpg#/media/File:Atanasoff-Berry\\_Computer.jpg](https://commons.wikimedia.org/wiki/File:Atanasoff-Berry_Computer.jpg#/media/File:Atanasoff-Berry_Computer.jpg)

<sup>8</sup> Tradução: Memória de Acesso Aleatório Dinâmico.

<sup>9</sup> © MIT. Cortesia. Fonte: Shannon, C. E. (1936). *A Symbolic Analysis of Relay and Switching Circuits*. Thesis M. S. MIT. Disponível em: <http://dspace.mit.edu/bitstream/handle/1721.1/11173/34541425-MIT.pdf>

<sup>10</sup> © MIT. Cortesia. Fonte: Shannon, C. E. (1936). *A Symbolic Analysis of Relay and Switching Circuits*. Thesis M. S. MIT. Disponível em: <http://dspace.mit.edu/bitstream/handle/1721.1/11173/34541425-MIT.pdf>

<sup>11</sup> “Z3 Deutsches Museum” by Venusianer at the German language Wikipedia. Licenced under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Z3\\_Deutsches\\_Museum.JPG#/media/File:Z3\\_Deutsches\\_Museum.JPG](https://commons.wikimedia.org/wiki/File:Z3_Deutsches_Museum.JPG#/media/File:Z3_Deutsches_Museum.JPG)

<sup>12</sup> Jack Copeland. Cortesia. Fonte: Copeland, B. J. (2010). *Colossus: The Secrets of Bletchley Park’s Codebreaking Computers*. Oxford University Press. Disponível em: <http://www.colossus-computer.com/colossus1.html>

<sup>13</sup> “Frontal view of the reconstructed Colossus at The National Museum of Computing, Bletchley Park” by TedColes - Own work. Licensed under CC BY-SA 4.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Frontal\\_view\\_of\\_the\\_reconstructed\\_Colossus\\_at\\_The\\_National\\_Museum\\_of\\_Computing,\\_Bletchley\\_Park.jpg#/media/](https://commons.wikimedia.org/wiki/File:Frontal_view_of_the_reconstructed_Colossus_at_The_National_Museum_of_Computing,_Bletchley_Park.jpg#/media/)

[File:Frontal\\_view\\_of\\_the\\_reconstructed\\_Colossus\\_at\\_The\\_National\\_Museum\\_of\\_Computing,\\_Bletchley\\_Park.jpg](#)

<sup>14</sup> “H96566k” by Courtesy of the Naval Surface Warfare Center, Dahlgren, VA., 1988. - U.S. Naval Historical Center Online Library Photograph NH 96566-KN. Licensed under Public Domain via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:H96566k.jpg#/media/File:H96566k.jpg>

<sup>15</sup> “Harvard Mark I Computer - Left Segment” by Original uploader was Daderot at en.wikipedia - Originally from en.wikipedia; Licenced under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Harvard\\_Mark\\_I\\_Computer\\_-Left\\_Segment.jpg#/media/File:Harvard\\_Mark\\_I\\_Computer\\_-Left\\_Segment.jpg](https://commons.wikimedia.org/wiki/File:Harvard_Mark_I_Computer_-Left_Segment.jpg#/media/File:Harvard_Mark_I_Computer_-Left_Segment.jpg)

<sup>16</sup> “Harvard Mark I constant switches detail” by ArnoldReinhold - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Harvard\\_Mark\\_I\\_constant\\_switches\\_detail.jpg#/media/File:Harvard\\_Mark\\_I\\_constant\\_switches\\_detail.jpg](https://commons.wikimedia.org/wiki/File:Harvard_Mark_I_constant_switches_detail.jpg#/media/File:Harvard_Mark_I_constant_switches_detail.jpg)

<sup>17</sup> “Harvard Mark I program tape.agr” by ArnoldReinhold - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Harvard\\_Mark\\_I\\_program\\_tape.agr.jpg#/media/File:Harvard\\_Mark\\_I\\_program\\_tape.agr.jpg](https://commons.wikimedia.org/wiki/File:Harvard_Mark_I_program_tape.agr.jpg#/media/File:Harvard_Mark_I_program_tape.agr.jpg)

<sup>18</sup> Tradução: “Como poderíamos pensar.”

<sup>19</sup> “Memex” by RotaruDan - Own work. Licenced under CC0 via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:Memex.jpg#/media/File:Memex.jpg>

<sup>20</sup> “Eniac” by Unknown - U.S. Army Photo. Licensed under Public Domain via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:Eniac.jpg#/media/File:Eniac.jpg>

<sup>21</sup> “Two women operating ENIAC” by United States Army - Image from [1]. Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Two\\_women\\_operating\\_ENIAC.gif#/media/File:Two\\_women\\_operating\\_ENIAC.gif](https://commons.wikimedia.org/wiki/File:Two_women_operating_ENIAC.gif#/media/File:Two_women_operating_ENIAC.gif)

<sup>22</sup> Fonte: Bauer, F. L., Wössner, H. (1972). *The “Plankalkül” of Konrad Zuse: A Forerunner of Today’s Programming Languages.* ACM. Disponível em: <http://www.catb.org/retro/plankalkuel/>

<sup>23</sup> “First transistor 2” by MdeVicente - Own work. Licensed under CC0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:First\\_transistor\\_2.svg#/media/File:First\\_transistor\\_2.svg](https://commons.wikimedia.org/wiki/File:First_transistor_2.svg#/media/File:First_transistor_2.svg)

<sup>24</sup> Tradução: “... avançar a ciência, desenvolvimento, construção e aplicação das novas maquinarias para computação, raciocínio e outras formas de manipulação

da informação.”

<sup>25</sup> Tradução: “A Associação é uma organização internacional educacional e científica dedicada a avançar a arte, ciência, engenharia e aplicação da tecnologia da informação, servindo tanto aos interesses públicos quanto profissionais, fomentando a troca livre de informação e promovendo os mais altos padrões éticos e profissionais.”

<sup>26</sup> “Zuse Z4 deutsches-museum” by Dasbloeckendeschaf - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Zuse\\_Z4\\_deutsches-museum.png#/media/File:Zuse\\_Z4\\_deutsches-museum.png](https://commons.wikimedia.org/wiki/File:Zuse_Z4_deutsches-museum.png#/media/File:Zuse_Z4_deutsches-museum.png)

<sup>27</sup> Tradução: *Cibernética, ou controle e comunicações no animal e na máquina.*

<sup>28</sup> © IBM. Cortesia dos “IBM Archives”.

<sup>29</sup> “IBM SSEC block diagram” by Joseph Robbins was patent attorney for IBM - Figure 1 from US Patent 2,636,672. “Selective Sequence Electronic Calculator” F.E. Hamilton, R.R. Seeber, R.A. Rowley, and E.S. Hughes (January 19, 1949). Issued April 28, 1953. Licensed under Public Domain via Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:IBM\\_SSEC\\_block\\_diagram.jpg#/media/File:IBM\\_SSEC\\_block\\_diagram.jpg](https://commons.wikimedia.org/wiki/File:IBM_SSEC_block_diagram.jpg#/media/File:IBM_SSEC_block_diagram.jpg)

<sup>30</sup> “Williams-tube” by Sk2k52 - Licensed under GFDL via Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:Williams-tube.jpg#/media/File:Williams-tube.jpg>

<sup>31</sup> “SSEM Manchester museum” by Parrot of Doom - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:SSEM\\_Manchester\\_museum.jpg#/media/File:SSEM\\_Manchester\\_museum.jpg](https://commons.wikimedia.org/wiki/File:SSEM_Manchester_museum.jpg#/media/File:SSEM_Manchester_museum.jpg)

<sup>32</sup> Tradução: “Primeiro rascunho de um relatório sobre o EDVAC.”

<sup>33</sup> “Von Neumann architecture”. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Von\\_Neumann\\_architecture.svg#/media/File:Von\\_Neumann\\_architecture.svg](https://commons.wikimedia.org/wiki/File:Von_Neumann_architecture.svg#/media/File:Von_Neumann_architecture.svg)

<sup>34</sup> “EDSAC (12)”. Licensed under CC BY 2.0 via Wikimedia Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:EDSAC\\_\(12\).jpg#/media/File:EDSAC\\_\(12\).jpg](https://commons.wikimedia.org/wiki/File:EDSAC_(12).jpg#/media/File:EDSAC_(12).jpg)

<sup>35</sup> Copyright © The University of Manchester. Cortesia. Disponível em: <http://curation.cs.manchester.ac.uk/computer50/www.computer50.org/kgill/mark1/mark1.jpg>

<sup>36</sup> “London Science Museum by Marcin Wichary - Pilot ACE, pt. 3 (2289287317)” by Marcin Wichary from San Francisco, U.S.A. - Pilot ACE, pt. 3. Licensed under CC BY 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:London\\_Science\\_Museum\\_by\\_Marcin\\_Wichary\\_-\\_Pilot\\_ACE,\\_pt.\\_3\\_\(2289287317\).jpg#/media/File:London\\_Science\\_Museum\\_by\\_Marcin\\_Wichary\\_-\\_Pilot\\_ACE,\\_pt.\\_3\\_\(2289287317\).jpg](https://commons.wikimedia.org/wiki/File:London_Science_Museum_by_Marcin_Wichary_-_Pilot_ACE,_pt._3_(2289287317).jpg#/media/File:London_Science_Museum_by_Marcin_Wichary_-_Pilot_ACE,_pt._3_(2289287317).jpg)

<sup>37</sup> “Turing Test version 3” by Bilby - Own work. Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Turing\\_Test\\_version\\_3.png#/media/File:Turing\\_Test\\_version\\_3.png](https://commons.wikimedia.org/wiki/File:Turing_Test_version_3.png#/media/File:Turing_Test_version_3.png)

# **PARTE VI**

## **Da Válvula ao Transistor**

A década de 1950 viu o surgimento da computação comercial. De um lado, a Remington Rand, que adquiriu a empresa de Mauchly e Eckert, e passou a produzir os UNIVAC comercialmente; de outro lado a IBM, iniciando com o modelo 701 e depois o 704, que foi um grande sucesso. Na Inglaterra, a Ferranti passou a produzir comercialmente os sucessores do Manchester Mark I.

Em termos de hardware, essa década viu a passagem gradual dos computadores a válvula para computadores transistorizados. As memórias dos computadores evoluíam dos tubos CRT e mercúrio para as memórias de núcleo magnético, que podiam ser muito maiores, rápidas e confiáveis a um custo relativamente baixo.

Em termos de software, a década viu o surgimento das primeiras linguagens de alto nível, em especial FORTRAN e LISP, que ainda hoje são usadas, e o ALGOL, inspiração para a maioria das linguagens de programação utilizadas atualmente. Nos anos 1950 foram também desenvolvidos os primeiros jogos eletrônicos, embora ainda fossem mera curiosidade e não a lucrativa indústria que são hoje. A década de 1950 também viu os computadores começarem a ser usados para tocar músicas e digitalizar imagens.

### **6.1 Human Use of Human Beings – 1950**

Considerando o quanto incipiente era a computação em 1950 é de se admirar que tenham existido visionários tão fantasticamente

influentes em nossa história quanto Norbert Wiener (Estados Unidos, 1894-1964). Depois de redefinir o termo “cibernética” em seu livro de 1948, ele brindou o mundo com uma nova e inspiradora visão de um futuro no qual a sociedade passa a ser uma mescla de homens e máquinas, ao publicar em 1950 o livro *The Human Use of Human Beings*.<sup>1</sup>

Wiener estabelece que se os séculos XVII e XVIII foram a era dos relógios e o século XIX a era das máquinas a vapor, o século XX deu início à era da comunicação e do controle.

Apesar de ter trabalhado em projetos militares durante a Segunda Guerra Mundial, Wiener recusou-se a continuar nessa área durante a Guerra Fria. Ele preferiu pesquisar e apresentar novos modos de organizar a sociedade.

É interessante notar a influência que seus pensamentos tiveram sobre o desenvolvimento posterior da computação visto que ela efetivamente nasceu a partir dos esforços relacionados à guerra. Colocar a computação para trabalhar para a consolidação de uma nova sociedade baseada em informação e controle era algo novo. Com exceção de poucos pensadores como Vannevar Bush e Norbert Wiener, muitos viam os computadores apenas como “mastigadores de números”, que poderiam ser usados em poucas aplicações além dos cálculos das tabelas para aplicações militares, financeiras e científicas.

Wiener considerava que a partir daquele momento da história, a integração dos seres humanos à tecnologia era inevitável. Ele disse: “*We have modified our environment so radically that we must now modify ourselves in order to exist in this new environment.*”<sup>2</sup>

## **6.2 Simon, o Primeiro Kit de Computador Pessoal – 1950**

Se perguntarmos a qualquer profissional de computação hoje quando foi inventado o computador pessoal, a maioria provavelmente responderá que foi na década de 1970 e lembrará de

máquinas como o Altair ou o Xerox Alto. Porém, em 1950 já existia um computador pessoal, o Simon. Ele era anunciado como o “menor cérebro eletrônico do mundo” e qualquer pessoa podia comprar um kit e montá-lo em casa.

Edmund C. Berkeley (Estados Unidos, 1909-1988), criador do Simon, chegou a conhecer alguns computadores a partir de 1939, como o Complex Number Computer e o Harvard Mark I. Desde 1934 ele trabalhava para a companhia de seguros Prudential e chegou a escrever a especificação de um computador UNIVAC que foi encomendado à empresa de Eckert e Mauchly. Em 1949, porém, a empresa o proibiu de participar de qualquer projeto contrário à guerra atômica, mesmo em seu tempo livre, o que o fez pedir demissão. Passou então a trabalhar como consultor independente.

Logo depois disso ele escreveu um livro que o deixou famoso *Giant Brains, or Machines That Think*.<sup>3</sup> No livro, ele descreve de forma acessível ao grande público como funcionavam essas máquinas elétricas e mecânicas. Em um determinado ponto ele diz, prevendo o futuro, que um dia o usuário poderia escrever algo como “receita de biscoitos” e a máquina iria acessar um longo rolo de filme até chegar ao ponto exato no qual exibiria em uma tela três ou quatro livros com receitas de bolos. Sem saber, ele estava prevendo a existência dos atuais mecanismos de busca da Internet.

Neste mesmo livro propõe a construção de um “minicérebro eletrônico”, que ele chamou de “Simon”. Ele deveria ser mais ou menos do tamanho de uma caixa de produtos de supermercado. Esse projeto, até onde se sabe, é a mais antiga concepção de computador pessoal, visto que desde Babbage até os dias de Berkeley, todos os computadores eram máquinas gigantescas que uma vez construídos jamais sairiam de suas salas inteiros. O Simon podia ser carregado com uma única mão, desde que a fonte fosse carregada com a outra.

Planos detalhados para a construção do computador foram publicados em uma série de artigos na revista *Radio-Electronics*. A

capa original da revista com uma foto do computador pode ser visualizada em <http://www.vintagecomputer.net/simon.cfm>. Ele teria uma arquitetura semelhante à do Harvard Mark I, com 129 relês e programação por fita perfurada de cinco colunas (tamanho padrão para os teletipos da época).

Os registradores eram ora de 2 ora de 4 bits e a unidade de processamento era de 2 bits. As saídas eram visualizadas em um painel com cinco lâmpadas. Apesar de trabalhar apenas com números binários de 2 bits, podendo representar assim apenas os algarismos de 0 a 3, ele era capaz de realizar adição e negação de números, verificar qual o maior dentre dois deles, além de outras operações.

Berkeley não esperava que a máquina tivesse qualquer propósito comercial ou científico. Para ele, Simon era como um kit de experiências com o qual alguém poderia aprender sobre o funcionamento de cérebros eletrônicos. Era, assim, um instrumento que aproximava a nova tecnologia que estava surgindo da pessoa comum. Um kit de construção do Simon custava cerca de 600 dólares, e até 1959 mais de 400 foram vendidos.

Ainda em 1950, Berkeley fundou e editou uma revista que no ano seguinte se tornaria a *Computers and Automation*, possivelmente a primeira publicação do tipo sobre computação no mundo.

## **6.3 Ferranti Mark I, o Primeiro Computador Comercial – 1951**

O Ferranti Mark I, produzido na Inglaterra e sobre o qual já falamos um pouco, foi o sucessor do Manchester Mark I. Ele é considerado o primeiro computador comercial do mundo efetivamente produzido. A primeira unidade foi entregue à Universidade de Manchester em fevereiro de 1951, enquanto que seu competidor americano, o UNIVAC, só foi vendido em 31 de março do mesmo ano e entregue apenas no final do seguinte.

Assim como o seu antecessor, o Ferranti também tinha memória principal baseada em tubos Williams, mas uma memória secundária baseada em tambores (cilindros) magnéticos foi adicionada.

Outro avanço que a máquina obteve foi um circuito multiplicador cerca de cinco vezes mais rápido do que o de seu antecessor. Alec Robinson projetou e construiu um circuito que fazia a multiplicação em paralelo, isso é, em vez de multiplicar um bit depois do outro, esse circuito conseguia multiplicar vários bits ao mesmo tempo, acelerando o processo.

O Ferranti tinha uma característica interessante: um alto-falante ligado aos seus circuitos, que emitia um som cuja tonalidade podia ser modificada através da programação da máquina. Não demorou até que os técnicos fizessem ele tocar algumas músicas reconhecíveis. De fato, a música foi até gravada e permanece como o mais antigo registro gravado de música tocada por computador. Porém, essa não foi a execução mais antiga, porque o CSIRAC, um computador australiano sobre o qual ainda falaremos, havia tocado músicas antes do Ferranti, embora não tenham sido gravadas na época.

O Ferranti era programado de uma maneira pouquíssimo intuitiva. Ele usava um código de base 32 (5 bits) para representar valores de 0 a 31. Os engenheiros resolveram manter o mesmo mapeamento entre os valores binários de 00000 a 11111 com caracteres de acordo com o mapeamento do teclado da máquina. Assim, os “dígitos” de 0 a 31 seriam representados pela seguinte sequência:

/ E @ A : S I U ½ D R J N F C K

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

T Z L W H Y P Q O B G “ M X V £

16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Dessa forma, o número 4, por exemplo, seria representando por “:”. Já o número 20 seria representado por “H” e 32 por “E/”. Os comandos da máquina também eram representados pelos mesmos caracteres. Pode-se concluir, portanto, que os programas do Ferranti eram verdadeiras “sopas de letrinhas”.

Apesar disso, ele era usável. Um de seus programadores chegou até a desenvolver um jogo limitado de xadrez para o qual, se houvesse possibilidade de xeque-mate em até duas jogadas, a máquina conseguiria examinar todos os movimentos praticáveis e escolher aquele que a levaria à vitória. Esse programa é considerado o mais antigo jogo implementado em um computador de propósito geral.

## 6.4 UNIVAC – 1951

O UNIVAC ou Universal Automatic Computer foi o sucessor do ENIAC e o primeiro computador comercial a ser vendido nos Estados Unidos. A máquina foi projetada e sua construção iniciada pela empresa Eckert-Mauchly Computer Corporation, mas o projeto só foi concluído depois que Eckert e Mauchly venderam a empresa para a Remington Rand. Após o surgimento de outras versões, a original passou a ser conhecida como UNIVAC I.

O primeiro UNIVAC foi encomendado pelo escritório do Censo americano em 1951. Já a quinta máquina pertencia à Comissão Americana de Energia Atômica. Em 1952, em uma jogada de marketing, a Remington pediu à rede de televisão CBS que ela usasse a máquina para prever o resultado da eleição presidencial daquele ano. Os especialistas previam uma vitória apertada de Adlai Stevenson, candidato Democrata, mas o UNIVAC, com uma amostra de apenas 1% da população votante, previu uma ampla

vitória para Dwight Eisenhower. Em função dessa discrepância, a CBS e a Remington preferiram não divulgar resultado da previsão do computador, o que teria sido feito em escala nacional, caso os resultados tivessem sido aceitos. Os programadores do UNIVAC possivelmente foram procurar as falhas no código ou nos dados, porém após a eleição de Eisenhower por uma ampla margem (442 a 89), viu-se que o computador de fato tinha acertado. Só então a história foi divulgada.

Mauchly e Eckert criaram o UNIVAC especificamente para ser uma máquina de gerenciamento de empresas, não para fins científicos. Assim, ela não precisava de muitas das funções científicas que foram implementadas em suas antecessoras, especialmente as do tempo da guerra. A Remington Rand, assim, como UNIVAC, competia diretamente com a IBM, que até então dominava o mercado de automatização de empresas com suas máquinas tabuladoras.

Porém, a Remington não chegou a ameaçar o domínio da IBM, pois faltava ao UNIVAC um recurso muito importante: boas leitoras e gravadoras de cartões perfurados. As empresas já tinham muita informação armazenada nos cartões IBM e dificilmente estariam dispostas a converter tudo isso em novos formatos como a fita magnética que o UNIVAC fornecia.

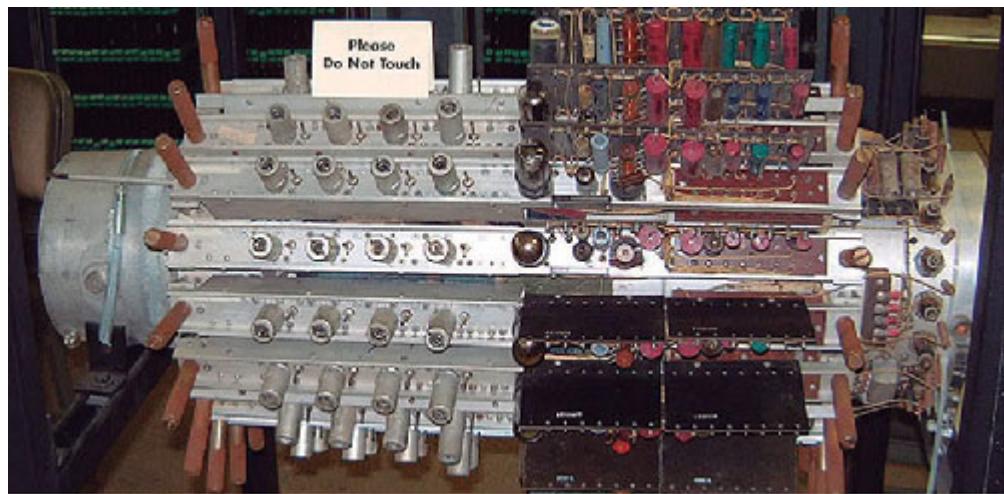
Embora inicialmente Eckert e Mauchly previssem que ele custaria cerca de 150 mil dólares, os custos finais acabaram ficando na casa de 1,2 a 1,5 milhões. Ainda assim, 46 UNIVAC foram vendidos pela Remington.

A [Figura 6.1](#) mostra o complexo painel de controle do UNIVAC, que mais parece a cabine de um antigo avião a jato.



**Figura 6.1: Painel de controle do UNIVAC.<sup>4</sup>**

Ao todo o UNIVAC tinha 5.200 válvulas, medindo 4,3 x 2,4 m com uma altura de 2,6 m. Ele pesava cerca de 13 toneladas e consumia 125 mil Watts. Também tinha uma memória grande para a época, com 1.000 palavras de 12 caracteres. Essas 1.000 palavras eram armazenadas em tubos de mercúrio semelhantes aos do EDVAC. Tratavam-se de 100 tubos, cada qual capaz de armazenar 10 palavras de 12 caracteres. Um dos módulos de memória de mercúrio do UNIVAC é mostrado na [Figura 6.2](#).



**Figura 6.2: Memória de mercúrio do UNIVAC.<sup>5</sup>**

Na tentativa de resolver os problemas de compatibilidade com os cartões perfurados da IBM, a Remington criou duas máquinas para converter cartões em fita magnética e vice-versa. Conta-se que uma destas máquinas tinha um compartimento dentro do duto de refrigeração que podia ser acessado por uma portinhola e que serviria, a pedido de um dos primeiros programadores do UNIVAC, para gelar cerveja quando ele estivesse trabalhando à noite.

## **6.5 LEO I – 1951**

Praticamente ao mesmo tempo que a Remington lançava o UNIVAC nos Estados Unidos, o Reino Unido também lançava um computador para uso empresarial. O LEO I, ou Lyons Electronic Office I, foi construído por uma empresa de alimentos, a J. Lyons and Co. Após a guerra, em 1947, a instituição enviou dois de seus gerentes sênior, Oliver Standingford e Raymond Thompson, para os Estados Unidos para conhecerem novos métodos de negócio desenvolvidos lá. Eles tiveram contato com o ENIAC de Mauchly e Eckert e rapidamente perceberam seu potencial para ajudar na administração. Ficaram sabendo também que em Cambridge um outro computador eletrônico, o EDSAC, estava sendo construído.

Desta forma, em visita a Cambridge a empresa decidiu primeiramente investir para ajudar a Universidade a finalizar o EDSAC, que foi finalizado em 1949. Após o projeto ela decidiu construir seu próprio computador, que seria uma expansão do EDSAC. Entre outras coisas, a Lyons usou o LEO I para processar sua folha de pagamento, controle de patrimônio, controle de vendas e *call center*. Pode-se afirmar que foi a primeira vez na história que uma única máquina integrou significativamente o gerenciamento de uma empresa.

## 6.6 CSIRAC – 1951

Lembram que no século XIX Ada Lovelace especulou que eventualmente os computadores poderiam compor e tocar músicas desde que instruídos com as regras da harmonia? Pois bem, essa visão se realizou pela primeira vez em 1951.

E surpreendentemente isso aconteceu na Austrália, onde foi construído um computador chamado CSIRAC, ou Commonwealth Scientific and Industrial Research Organisation Automatic Computer, inspirado na arquitetura do Manchester Baby. Ele continua intacto e é reputado como o mais antigo computador de primeira geração ainda existente. A [Figura 6.3](#) é uma foto dele tirada em 2008.

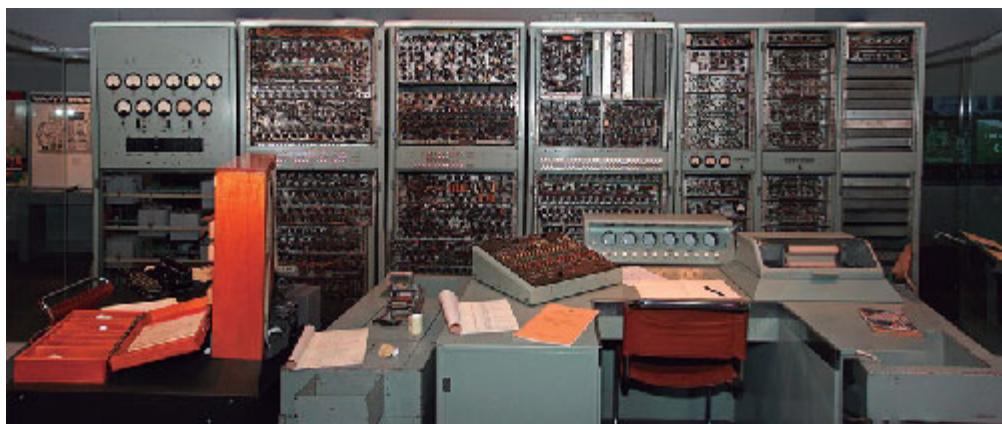


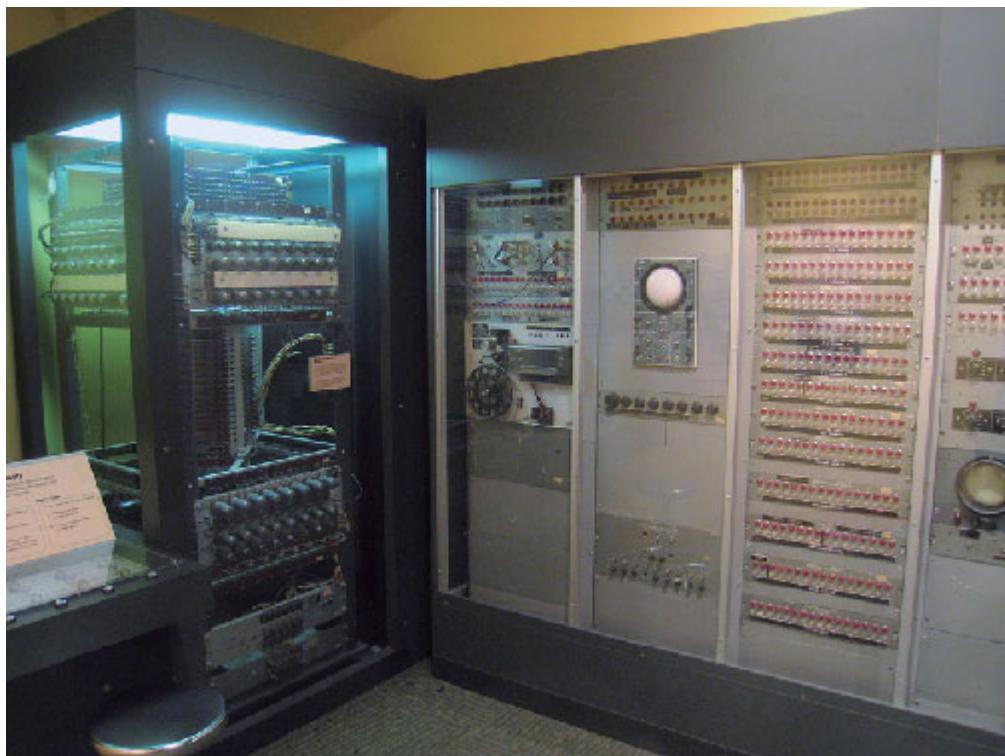
Figura 6.3: CSIRAC.<sup>6</sup>

Ele foi desenvolvido por Trevor Pearcey (Reino Unido, 1919-1998) e Maston Beard, do Conselho de Pesquisa Científico e Industrial (CSIR) da Austrália. De acordo com os construtores, ele rodou o primeiro programa – cuja função era multiplicar dois números – no final de 1949. Porém, ninguém se lembra do dia exato; dizem que apenas gritaram “hurra!” e voltaram a trabalhar.

Ao contrário da maioria dos computadores, que transferem todos os bits de uma palavra em paralelo através de vários condutores, o CSIRAC transmitia um bit de cada vez. Esses meios de condução são hoje chamados de “barramentos”. Assim, o CSIRAC tinha um barramento de 1 bit. O computador usava também uma memória de tubo de mercúrio e, dessa maneira, de fato apenas 1 bit de cada vez era lido em cada tubo. Os construtores aproveitaram essa característica do CSIRAC para ligar um alto-falante a um dos barramentos, o qual produzia uma tonalidade sonora cada vez que certa configuração de bits passava por ali. Assim, foram capazes de programar o computador de forma que as tonalidades produzidas por ele se parecessem com músicas conhecidas. Desta forma, o CSIRAC além de ser o primeiro computador do Hemisfério Sul, também foi o primeiro computador a cantar.

## 6.7 MIT Whirlwind, o Primeiro Computador de Tempo Real – 1951

O Whirlwind (tornado), mostrado na [Figura 6.4](#), foi um computador a válvula desenvolvido pelo MIT (Massachusetts Institute of Technology) para a Marinha americana. Ele foi o primeiro na história a ser projetado para trabalhar em tempo real. Até então todos os computadores que existiam processavam dados de entrada e produziam uma saída. Assim, se ele fosse mais rápido, ótimo; e se fosse mais lento era só o caso de esperar um pouco mais para obter os resultados. Esse tipo de processamento é até hoje chamado de processamento *batch*, ou “em lote”.



**Figura 6.4: Whirlwind.<sup>7</sup>**

No entanto, o Whirlwind foi pensado para outro tipo de aplicação. Desejava-se um computador que pudesse controlar os parâmetros de um simulador de voo. Mais do que isso: simuladores de voo baseados em computação analógica já existiam naquela época, mas queriam um simulador que pudesse simular qualquer tipo de aeronave. Apenas um verdadeiro computador digital poderia fazer isso; a partir de dados de configuração predefinidos, ele poderia simular diferentes aeronaves. Porém, este computador teria que ser muito rápido, pois seria o primeiro a operar em tempo real, ou seja, não se poderia esperar pelo resultado de um processamento: cada ação do piloto sobre o simulador teria que surtir efeito imediatamente.

Depois de experimentar com arquiteturas analógicas durante a guerra, o encarregado do projeto, Jay W. Forrester (Estados Unidos, 1918), concluiu que esse tipo de computador era pouquíssimo

preciso e flexível. Assim, em 1945, após a equipe ter tido contato com o ENIAC, chegaram à conclusão de que uma máquina digital seria a solução.

O problema é que as memórias de mercúrio que então estavam em uso eram muito lentas para os propósitos do Whirlwind. Eles acabaram decidindo por usar tubos CRT, ou tubos Williams, que eram uma opção mais rápida. Porém, a equipe considerou que os ciclos de reforço de sinal necessários para manter a memória iriam atrasar os tempos de processamento. Como vimos, o sinal eletrostático na tela do CRT permanecia por apenas um quinto de segundo, e era necessário reforçar todos os pontos da tela antes desse tempo passar para evitar que a memória simplesmente se apagasse, e isso tomava tempo de processamento.

Eles resolveram apostar em uma nova tecnologia de tubos eletrostáticos que estava sendo desenvolvida no MIT e que tornaria desnecessários os ciclos de reforço de sinal. Haveria um segundo cátodo no tubo que enviaria uma enxurrada de elétrons de baixa energia para a tela inteira. A técnica era chamada de “inundação” e, se funcionasse, os pontos carregados positiva ou negativamente na tela ficariam ali permanentemente, sem necessidade de ciclos de reforço.

Infelizmente, a nova tecnologia não funcionou. Desesperado, Forrester procurou por alternativas até se deparar com a propaganda de uma companhia que estava desenvolvendo núcleos magnetizáveis, que ele percebeu imediatamente que seriam uma alterativa para a construção da memória para o Whirlwind.

Nasceu assim a memória de núcleo magnético, que foi o padrão da indústria de computadores até 1970. A [Figura 6.5](#) mostra esquematicamente uma memória de núcleo magnético de 16 bits, organizados em quatro linhas de quatro colunas. Cada núcleo é um anel de material magnetizável que, ao ser exposto a uma corrente elétrica passando por dentro do anel, pode ser magnetizado em uma direção ou outra. Desta forma, pode-se considerar que um anel

magnetizado em sentido horário armazena um 0 e um anel magnetizado em sentido anti-horário armazena um 1.

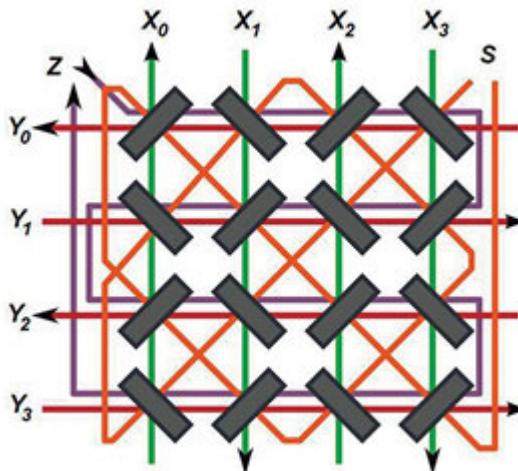


Figura 6.5: Esquema de uma memória de núcleo magnético.<sup>8</sup>

Os anéis depois de magnetizados conseguem manter este estado indefinidamente, ou seja, trata-se de uma memória não volátil, pois mesmo que a corrente seja totalmente desligada, as informações ali armazenadas são mantidas. Ocorre que a cerâmica da qual são feitos os anéis só muda seu sentido de magnetismo se for exposta a certo nível de corrente. Enquanto ela estiver abaixo deste nível não há mudança. Assim, um engenhoso mecanismo matricial permite magnetizar qualquer um dos anéis individualmente.

Observe na [Figura 6.5](#) que há quatro fios X<sub>0</sub> ... X<sub>3</sub> e mais quatro fios Y<sub>0</sub> ... Y<sub>3</sub>. Se você quiser armazenar, por exemplo, 1 no núcleo na posição (2, 2), o que tem a fazer é passar metade da corrente necessária para magnetizar um núcleo por X<sub>2</sub> e metade da corrente necessária em Y<sub>2</sub>. Assim, apenas o núcleo (2, 2) será exposto a uma corrente suficientemente alta para mudar seu estado, enquanto que os demais permanecem no estado que estavam antes.

As linhas S e Z, mostradas na [Figura 6.5](#), atravessam todos os núcleos e são respectivamente o “sensor” e o “inibidor” da memória.

Versões posteriores passaram a usar apenas uma linha SZ, visto que, embora tenham funções diferentes, elas são sempre usadas em momentos diversos.

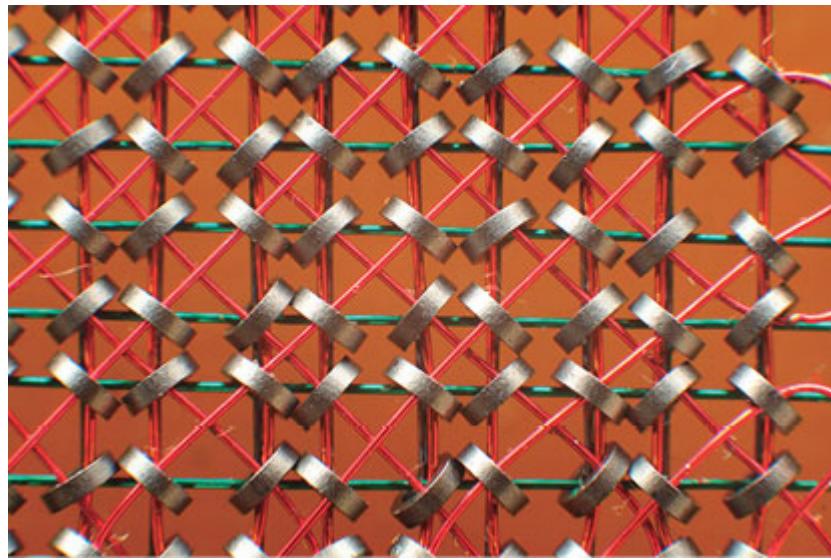
A linha S serve, portanto, como sensor – ou seja, é ela que permite fazer a leitura de dada posição de memória. A leitura em uma memória de núcleo magnético destrói a informação ali armazenada, isto é, quando a informação é lida ela é destruída. Isso ocorre porque o processo de leitura consiste em tentar induzir o valor 0 em um determinado núcleo, digamos (2, 2). Assim, ela é feita aplicando-se metade da corrente, como vimos antes, nos fios X2 e Y2, tentando mudar o valor do núcleo (2, 2) para 0. Se (2, 2) já está em 0, nada ocorre, mas se (2, 2) está em 1, haverá uma inversão de sua polarização e esta causará um pulso no fio S, o qual poderá ser lido.

Há um tempo de espera para esse pulso. Se aplicarmos a corrente em X2 e Y2 e após o tempo de espera não houver pulso nenhum em S é porque (2, 2) já contém 0. Porém, se houver o pulso é porque (2, 2) continha um 1.

Porém, depois da leitura (2, 2) estará com 0. Então, se seu estado anterior fosse 1 será necessário regravar o 1 na posição (2, 2), aplicando novamente a meia-corrente (agora inversa) em X2 e Y2.

Para a escrita, parte-se do princípio de que se a posição de memória foi lida, então ela está com um “0”. Para escrever “1”, aplica-se meia-corrente em X e Y, como vimos. Para escrever um “0”, ou melhor, para evitar que o núcleo deixe de ser zero, aplica-se meia-corrente inversa na linha do inibidor (Z). Assim, a corrente total que passará pelo núcleo será metade da corrente necessária para invertê-lo e, dessa maneira, ele permanecerá com um “0” armazenado ali.

A [Figura 6.6](#) mostra uma foto de uma memória de núcleo magnético real. Os espaços entre os núcleos são de cerca de 1 mm.



**Figura 6.6: Memória de núcleo magnético.<sup>9</sup>**

## **6.8 OXO – 1952**

O primeiro jogo gráfico de computador foi implementado em 1952 por Alexander S. Douglas (Reino Unido, 1921-2010) durante seu doutorado em Cambridge. Lá, ele teve contato com o EDSAC, que tinha memória baseada em tubos de mercúrio, mas cujo conteúdo podia ser exibido em um tubo CRT.

Assim, Douglas implementou o OXO, ou “jogo da velha”, desenhando a armação do jogo na tela e implementando um algoritmo de decisão (uma forma primitiva de inteligência artificial) para fazer uma partida perfeita contra um humano. Assim, ou a máquina ganhava ou empatava.

O humano podia entrar com sua jogada através de um disco de telefone, no qual selecionava uma posição entre 1 e 9. Um pequeno filme com uma simulação do jogo pode ser visto em <https://www.youtube.com/watch?v=SRnGlzn8DA0>.

## **6.9 Compilador A-0 – 1952**

Grace M. Hopper (Estados Unidos, 1906-1992) foi uma destacada programadora do Harvard Mark I. Já nos referimos a ela por ter sido a primeira pessoa a realmente encontrar um “bug” em um computador. Em 1949, Eckert e Mauchly contrataram Hopper para a equipe de desenvolvimento de software para o UNIVAC.



Figura 6.7: Grace Hopper operando o UNIVAC.<sup>10</sup>

Em 1952 ela finalizou o projeto do Sistema A-0, considerado o primeiro compilador efetivo, isto é, um programa que traduz instruções mais abstratas em comandos que a máquina pode executar. Teria sido ela, inclusive, a pessoa a cunhar o termo “compilador”.

Seu compilador era bem simples. Na verdade, era mais parecido com um carregador de rotinas. Ela selecionou um conjunto de programas que realizavam cálculos específicos, por exemplo, seno, cosseno etc., e colocou todos esses programas em uma fita. Ela anotou a posição inicial de cada um deles, de forma que pudesse

fazer a máquina avançar para a posição correta sempre que precisasse de um.

Assim, um “programa” escrito em A-0 nada mais era do que uma lista de chamadas para essas rotinas. Cada chamada era representada por um número que correspondia a uma posição da fita e após esse número eram passados os valores dos parâmetros.

O A-0, assim, não era um compilador na acepção moderna do termo. Porém, esse projeto motivou Hopper a continuar trabalhando em seu aperfeiçoamento. A área de pesquisa foi chamada de “programação automática”; hoje nós a chamamos de “programação em alto nível”, para diferenciar da programação em código de máquina. Programação automática seria a possibilidade de programar o computador com uma linguagem semelhante à linguagem natural, como inglês ou português.

Em um artigo publicado em 1952, Richard Ridgway apresenta um comparativo de tempo dispendido por programadores usando o A-0 e usando o método de programação convencional. Com a técnica usual foram necessários 740 minutos de programação, contra 20 minutos usando o compilador.

O A-0 foi sucedido pelo A-1, A-2 e A-3. Conta-se que a Remington Rand disponibilizou o código fonte (programa) do A-2 gratuitamente para os usuários de seus computadores UNIVAC e que também os encorajava a enviarem sugestões e feedback sobre o sistema. Possivelmente, essa foi então a primeira iniciativa de software livre e de código aberto a existir no mundo. A partir do A-2 a programação já era feita usando uma notação que chamaram de pseudocódigo, ou seja, um código que a máquina não conseguia processar normalmente. Construíram, no entanto, um sistema que lia as instruções em pseudocódigo da fita magnética e as convertia em instruções que a máquina podia executar.

As versões A-0 até A-3 eram chamadas de ARITH-MATIC, uma versão posterior, AT-3, foi chamada de MATH-MATIC e a versão final, B-0 de FLOW-MATIC. A FLOW-MATIC, uma linguagem já

muito parecida com inglês estruturado, foi a base para a criação da linguagem COBOL.

Comenta-se que Hopper teria enfrentado grande resistência por parte daqueles que programavam com código de máquina. Ela chegou a desabafar: "*I had a running compiler, and nobody would touch it because, they carefully told me, computers could only do arithmetic; they could not do programs.*"<sup>11</sup>

## 6.10 IBM 701 – 1952

A IBM não ficou restrita apenas ao mercado de computadores comerciais. Seu primeiro grande projeto de computador científico para produção em escala, inicialmente chamado de Defense Calculator, foi lançado em 1952 com o nome de IBM 701 ([Figura 6.8](#)).

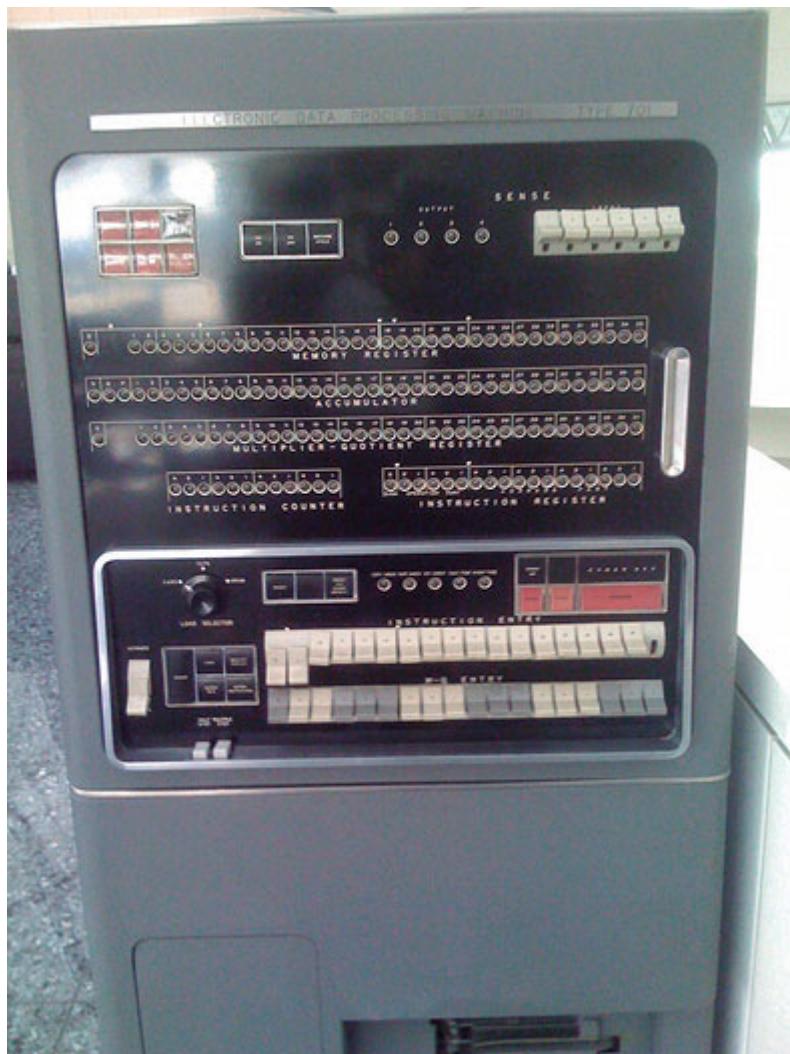


Figura 6.8: Console do IBM 701.<sup>12</sup>

Tratava-se de um computador a válvulas com memória baseada em 72 tubos Williams com capacidade de 1.024 bits cada. Assim, a máquina tinha uma memória total de 2.048 (2K) palavras de 36 bits. Dezenove computadores modelo 701 foram fabricados. O aluguel girava em torno de 8 mil dólares por mês.

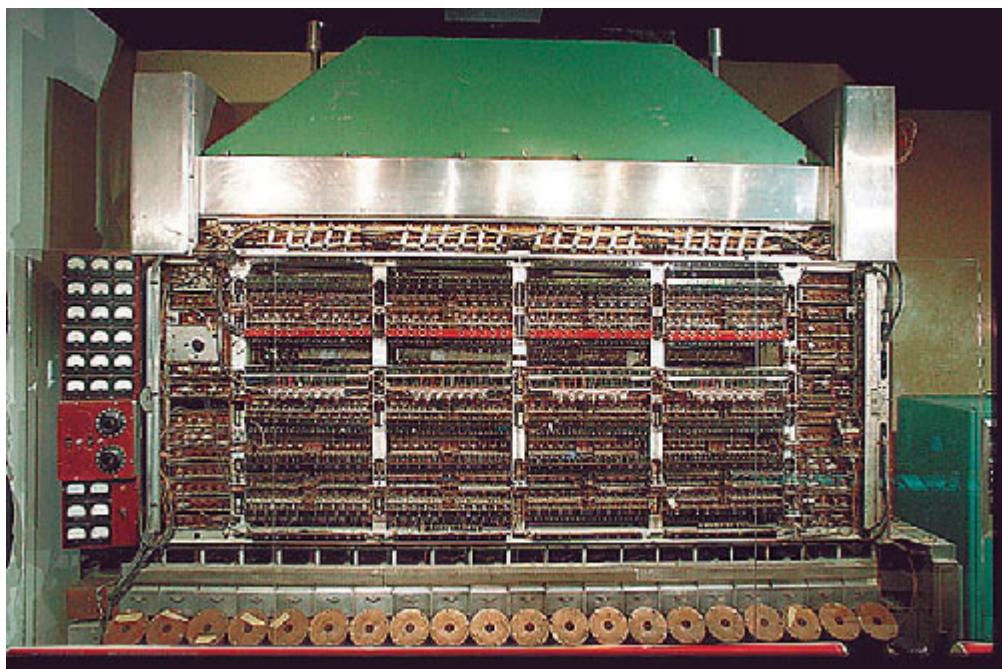
O folclore atribui a Watson, presidente da IBM, a frase “eu acho que no mundo há mercado para no máximo cinco computadores”. Porém, essa lenda provavelmente deve-se à afirmação de Watson de que, antes da produção do 701, ele saiu para visitar 20

potenciais clientes na esperança de vender para pelo menos cinco deles e voltou com 18 pedidos.

Essa foi a primeira máquina totalmente eletrônica e com programa armazenado fabricada pela IBM. Uma das unidades do 701 substituiu o SSEC na Avenida Madison em Nova York. Ela era 25 vezes mais rápida e ocupava menos de 25% do espaço ocupado pelo SSEC, que era construído com relês. O 701 foi o marco para a mudança do foco da IBM do negócio com tabuladoras para o negócio dos computadores eletrônicos.

## 6.11 IAS – 1952

John von Neumann, professor de Princeton e consultor de Eckert e Mauchly durante algum tempo, também projetou e construiu em 1952 um computador, o IAS ([Figura 6.9](#)), nome dado em homenagem ao instituto onde foi construído, o Institute of Advanced Studies. O IAS é também conhecido como “Computador de von Neumann”.



**Figura 6.9: IAS.**<sup>13</sup>

Ele não foi o primeiro computador a implementar a arquitetura de programa armazenado, pois o Manchester Baby já fazia isso anos antes. Porém, von Neumann inovou mais uma vez ao identificar a possibilidade de realizar comandos de repetição em memória, ou seja, com o programa armazenado em memória não seria mais necessário colar as duas pontas de uma fita de programa para realizar instruções em ciclos repetitivos; isso poderia ser feito por comandos em memória.

## 6.12 Trackball – 1952

Quem foi inventado primeiro? O *mouse* ou a *trackball*? Bem, foi a trackball, e foi em 1952.

O primeiro protótipo de trackball, chamado “rollerball”, foi criado por Ralph Benjamin em 1941 para substituir um sistema de joystick que seria usado para entrar com coordenadas para calcular trajetórias de projéteis. Ele achava o joystick desajeitado e propôs usar uma bola metálica encostada em duas rodas. Girando a bola, as rodas “leriam” as posições X e Y. Ele patenteou a ideia em 1947, mas ela nunca foi colocada em prática.

Assim, a primeira trackball efetivamente construída e usada foi fabricada no Canadá, com uma bola de boliche canadense, que é menor do que a bola americana e mais parecida com a nossa do jogo de bolão.

Este equipamento foi usado como dispositivo de entrada em um projeto da Ferranti Canada, subsidiária da Ferranti britânica, responsável pela construção do Ferranti Mark I. O projeto em questão chamava-se DATAR, ou Digital Automated Tracking and Resolving, que seria um sistema de comunicação e processamento de dados eletrônicos navais para a Marinha canadense. O objetivo do projeto era ser um sistema integrado que permitisse consultar e controlar a posição de navios e aviões em tempo real.

A trackball permite ao usuário mover a esfera para registrar uma dupla entrada de dados no computador. Com movimentos da mão sobre a esfera o usuário muda os eixos X e Y, ou coordenadas de entrada. Há – além dos suportes para a bola – dois discos, colocados em posição ortogonal ( $90^\circ$  um do outro). Ao girar a bola em qualquer sentido, giros correspondentes são capturados por esses discos e transmitidos ao computador.

O projeto DATAR infelizmente não foi concluído da forma como se esperava, pois a máquina que faria o processamento dos dados já estava com mais de 30 mil válvulas, e os erros devidos à queima desses componentes estavam acima do aceitável. Anos depois, versões transistorizadas do sistema foram construídas.

Entretanto, o DATAR deixou um interessante legado para o Canadá, pois o país não tinha ainda uma cultura de construção de computadores. Com o projeto, a Ferranti Canadá passou a investir na educação de seus jovens engenheiros, que mais tarde se tornariam pessoas-chave na construção da cultura eletrônica e computacional do país.

## 6.13 IBM 650 e 704 – 1953

O primeiro computador produzido em massa foi o IBM modelo 650, ou IBM Magnetic Drum Data-Processing Machine. Cerca de 2 mil cópias dele foram produzidas desde seu anúncio em 1953 até 1962. Os números no 650 eram representados de forma semelhante ao ábaco, em “bi-quinário”, com 2 bits (o “bi”) para representar se o dígito era de 0 a 4 ou de 5 a 9, e mais 5 bits para representar o número em si (o “quinário”). Assim, os dígitos de 0 a 9 eram representados da seguinte forma:

- 0: 10-10000
- 1: 10-01000
- 2: 10-00100
- 3: 10-00010
- 4: 10-00001

- 5: 01-10000
- 6: 01-01000
- 7: 01-00100
- 8: 01-00010
- 9: 01-00001

A [Figura 6.10](#) mostra um detalhe do painel de controle do 650 onde claramente se veem os indicadores numéricos bi-quinários.



**Figura 6.10:** Painel do IBM 650.<sup>14</sup>

O 650 era vendido como uma máquina para aplicações científicas e de engenharia. Como era relativamente barato e versátil, chegou a ser usado não apenas comercialmente, mas também como máquina para ensinar Ciência da Computação nas universidades. O 650 tinha uma memória de tambor magnético que podia conter 1, 2 ou 4 mil palavras (1, 2 ou 4 kB).

Donald Knuth (Estados Unidos, 1938), autor da influente série de livros *The Art of Computer Programming* de 1968, dedicou-a ao IBM 650: “*This series of books is affectionately dedicated to the Type 650*

*computer once installed at Case Institute of Technology, with whom I have spent many pleasant evenings.”<sup>15</sup>*

Em 1954, a IBM lança outro marco da história da computação: o modelo 704, que seria considerado o sucessor do 701, um computador eminentemente científico, capaz de operar com números de ponto flutuante. Uma das principais melhorias foi o uso de memória de núcleo magnético no lugar dos tubos Williams. Outra foi a adição de registradores de índice, que permitiram realizar operações sobre vetores de forma bem mais eficiente.

## 6.14 Primeiros Sistemas Operacionais – 1954

Os primeiros computadores eram simplesmente máquinas que rodavam programas previamente construídos, um de cada vez. O usuário chegava, introduzia o programa na máquina (às vezes arduamente mudando plugues e chaves, mas depois usando fitas e cartões perfurados), introduzia os dados, iniciava o programa e aguardava o final do processamento, com as saídas sendo impressas ou perfuradas no meio adequado.

Os computadores mais antigos não tinham nenhuma facilidade para, por exemplo, rodar vários programas, um após o outro, de forma automática. Havia sempre a necessidade de um operador humano fazer a retirada do programa anterior e a colocação do novo. Nessa época, a “fila de trabalho” dos computadores era literalmente uma fila de pessoas na porta da sala onde estava instalado, aguardando para rodarem seus programas. Comenta-se que na Universidade de Cambridge, na Inglaterra, os programas em fitas perfuradas eram colocados em varais de roupa e a cor do prendedor usado indicava a sua prioridade na fila.

Porém, em meados dos anos 1950 os computadores começaram a ficar extremamente rápidos em comparação com os anteriores e logo passou a se gastar mais tempo fazendo a colocação dos programas na máquina do que efetivamente rodando-os. Como os computadores eram absurdamente caros, o tempo ocioso – em que

não estavam rodando qualquer programa – precisava ser minimizado.

Por que então não construir um programa capaz de pegar vários programas em fila e automaticamente introduzi-los na máquina? Isso foi feito, e assim nasceram os primeiros sistemas operacionais. Na época, eram chamados de “monitores” ou “programas monitores”, e o tipo de trabalho que faziam é equivalente ao que o kernel dos modernos sistemas operacionais realiza.

O nome “sistema operacional” pareceu mais adequado porque ele realmente substitui o operador em muitas de suas tarefas. Com o método manual, um operador humano teria que ficar 24 horas por dia com a máquina para que ela pudesse operar, trocando os programas e dados. Entretanto, com o sistema operacional a máquina podia trabalhar sozinha. Bastava que os diversos programas fossem enfileirados na leitora de fita ou leitora de cartões e com as devidas diretivas (cartões de “job”) a máquina conseguiria sequenciar esses programas automaticamente, sem a necessidade do operador humano.

Outra coisa que esses sistemas operacionais logo começaram a incorporar foram as bibliotecas de rotinas, que forneciam um acesso mais fácil para as diferentes partes do hardware ou periféricos da máquina. No começo da computação, um programador teria que saber exatamente quais comandos havia e como funcionava todo o hardware e seus periféricos. Se houvesse, por exemplo, três modelos de leitora de fita, cada um deles com comandos diferentes, o programador teria que conhecer todos e saber como programá-los. Já com o sistema operacional, o programador usaria uma rotina mais abstrata, a qual faria o serviço na leitora de fita independentemente de quem fosse seu fabricante ou qual seu conjunto de instruções. Esse tipo de rotina hoje é chamada de “driver” e é o que nos permite usar as mais diversas impressoras, scanners etc., sem precisar saber programá-los.

De acordo com a literatura, os três primeiros sistemas operacionais foram desenvolvidos em 1954, 1955 e 1956. Em 1954, o MIT desenvolveu uma espécie de sistema operacional chamado Tape Director para o seu UNIVAC. Em 1955, a General Motors desenvolveu um sistema operacional para o seu IBM 701 e, em 1956, um outro chamado GM-NAA I/O para o seu IBM 704.

Nessa época os sistemas operacionais eram quase sempre desenvolvidos pelos clientes e não pelos fabricantes das máquinas, pois eram os clientes quem reconheciam de fato a sua necessidade.

Outra coisa que os sistemas operacionais proporcionaram desde muito cedo foi a possibilidade de multitarefas. Por exemplo, nos primeiros computadores, enquanto a impressora estivesse imprimindo, o computador parava seus demais processos e aguardava o final da impressão. Com a introdução dos sistemas operacionais, permitiu-se que a máquina começasse a rodar o programa seguinte enquanto o anterior ainda estivesse imprimindo sua saída. Essa técnica foi chamada de *spooling* e significa: “*Simutaneous peripheral operation on line*.<sup>16</sup> Mais tarde, até mesmo tornou-se possível rodar vários programas ao mesmo tempo na mesma máquina.

## 6.15 Harwell Cadet, o Primeiro Computador Totalmente Transistorizado – 1955

Existe alguma disputa sobre qual teria sido o primeiro computador transistorizado. O fato é que em meados da década de 1950, mesmo com válvulas mais confiáveis do que as da década de 1940, elas ainda eram um incômodo: custavam caro, queimavam com frequência, ocupavam muito espaço e consumiam uma quantidade absurda de energia.

O caminho seria investir em semicondutores sólidos e o grande candidato era o transistor, inventado em 1947. O transistor em si vinha evoluindo também, de versões de laboratório para versões mais aceitáveis para uso industrial.

Os computadores mecânicos, eletromecânicos e a válvula são considerados parte da Primeira Geração de computadores. Com o advento dos computadores transistorizados, inicia-se a Segunda Geração.

A primeira experiência de que se tem registro ocorreu na Universidade de Manchester, Reino Unido, com Tom Kilburn (Reino Unido, 1921-2001) onde um pequeno protótipo de computador transistorizado, o “University of Manchester’s Experimental Transistor Computer”, foi construído em 1953 e depois em tamanho real em 1955. Porém, essa máquina ainda usava algumas válvulas para gerar pulsos de clock<sup>17</sup> mais rápidos e, logo, ela não é considerada totalmente transistorizada.

Em 1954, a Bell Labs nos Estados Unidos criou uma máquina semelhante, o TRADIC, ou TRAnsistor Dlgital Computer. Ele também usava válvulas para gerar pulsos de clock, visto que nenhum transistor da época conseguia produzir toda a energia necessária (30 Watts) com a mesma velocidade.

A IBM anunciou em 1955 uma calculadora totalmente transistorizada – embora só a tenha lançado de fato em 1957. Tratava-se da IBM 608, que continha mais de 3 mil transistores de germânio.

A máquina que pode ser considerada, com propriedade, o primeiro computador de propósito geral totalmente transistorizado é o Harwell Cadet ([Figura 6.11](#)). Em 1953 a divisão de eletrônica das instalações de pesquisa em energia atômica de Harwell, Reino Unido, resolveram investir na tecnologia de transistores para substituir uma máquina mais antiga. Cadet é a sigla para Transistor Electronic Digital Automatic Computer de trás para frente. Ele rodou seu primeiro programa em fevereiro de 1955.

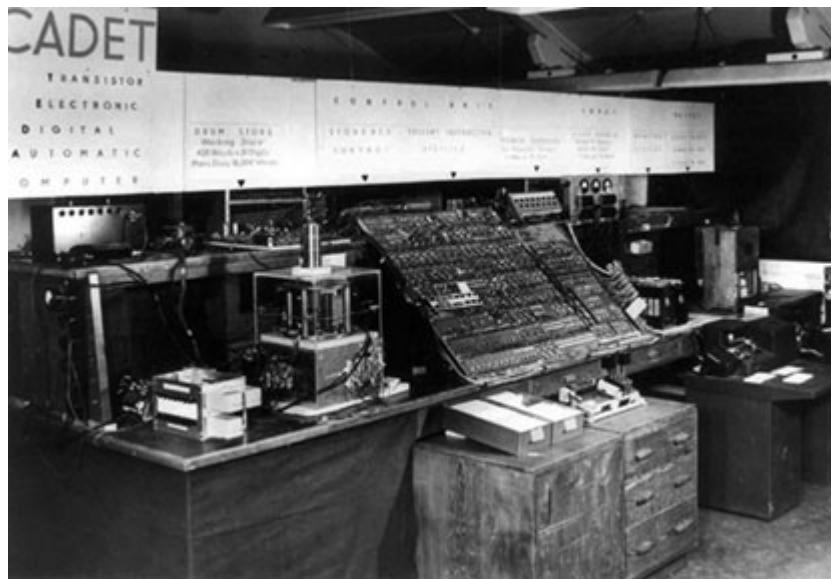


Figura 6.11: Harwell Cadet.<sup>18</sup>

Um dos preços que os construtores tiveram que pagar para ter uma máquina totalmente transistorizada foi um clock baixo, de apenas 58 kHz, quando computadores a válvula já tinham clock até 20 vezes mais rápido.

Em 1956, o Laboratório Lincoln do MIT também construiu um computador totalmente transistorizado, o TX-0 (apelidado de “ticso”). Ele utilizava um novo tipo de transistor de alta velocidade desenvolvido pela Philco em 1953. Com o sucesso da máquina, logo passaram a um projeto mais sofisticado: o TX-1. Como ele demonstrou ser complexo demais, partiram para um conceito simplificado: o TX-2, que acabou sendo construído com partes do TX-0, visto que componentes de computadores, especialmente as memórias, eram demasiadamente caros na época.

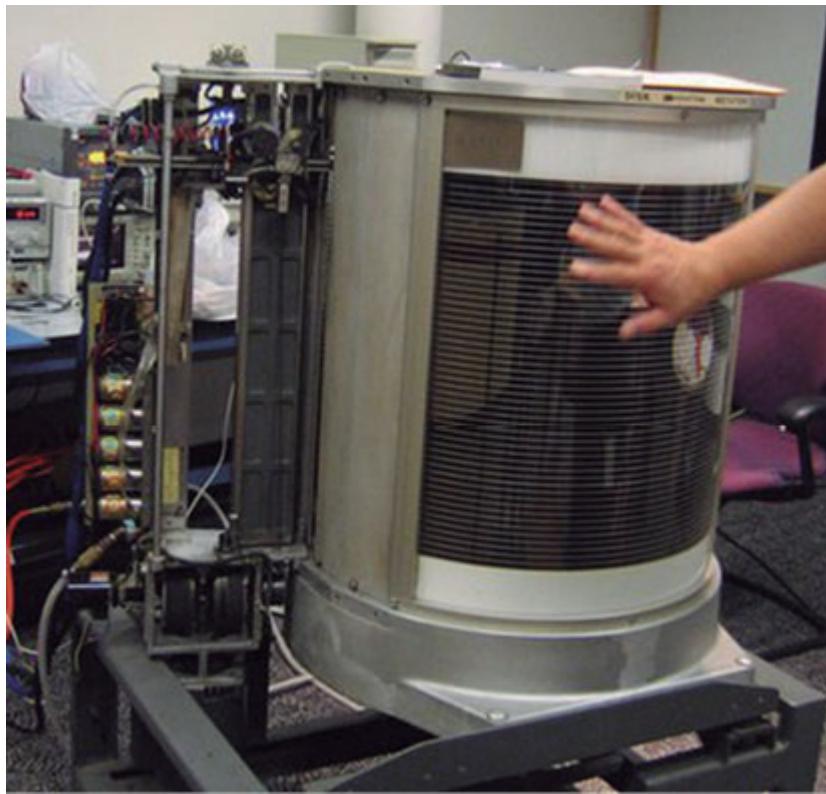
As partes que sobraram do TX-0 acabaram sendo doadas para o Laboratório de Pesquisa Eletrônica do MIT, onde se tornariam peças-chave para a criação do futuro Laboratório de Inteligência Artificial do MIT e da cultura hacker.

## 6.16 Disco Magnético – 1956

Até 1956 os dispositivos de armazenamento secundário dos computadores eram basicamente a fita perfurada ou magnética, os cartões perfurados e os tambores magnéticos. No caso das fitas e cartões, existia um problema muito sério porque estes meios de armazenamento eram memórias lineares. Isto é, para se atingir um determinado ponto da fita, você precisaria avançar ou rebobinar até ele. Um programa que acessasse dados em posições variadas numa fita ficaria mais tempo avançando e rebobinando do que fazendo qualquer outra coisa, visto que essas operações eram demoradas.

No caso dos tambores magnéticos, não havia esse problema: qualquer ponto do tambor podia ser acessado por um cabeçote em um tempo muito curto. Porém, eles ocupavam muito espaço físico. Em 1956, a IBM inovou a indústria de armazenamento secundário ao lançar o primeiro sistema de armazenamento baseado em discos, o disco rígido ou hard drive como hoje é conhecido. A principal vantagem em relação às fitas e aos cartões era que qualquer dado armazenado podia ser acessado rapidamente, pois cabeçotes de leitura corriam sobre as 100 trilhas concêntricas de cada disco, que girava a 1.200 rotações por minuto (bem mais que um LP, que usualmente girava a 33 RPM), e permitiam rapidamente alcançar uma grande quantidade de dados.

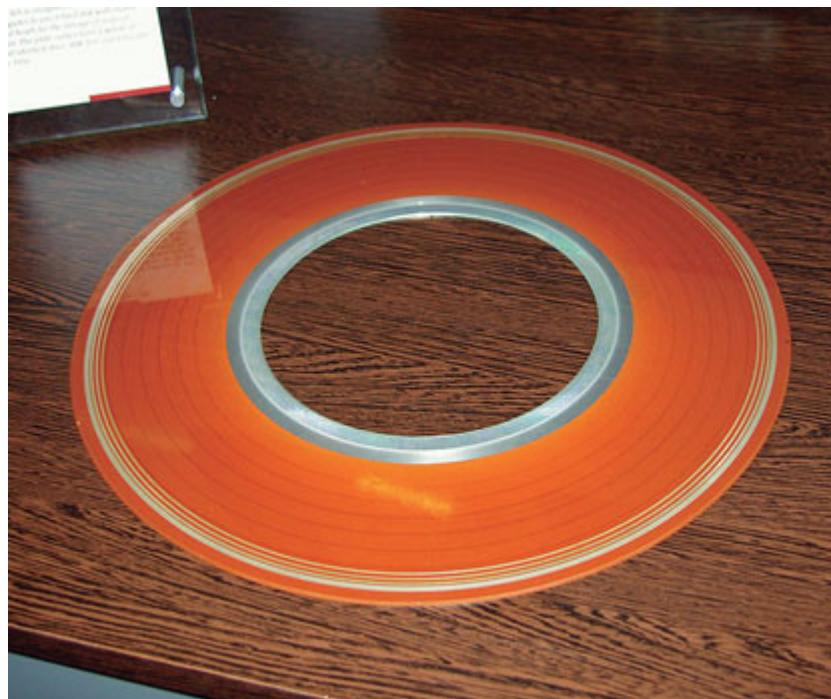
Em relação aos tambores, a vantagem dos discos era a economia de espaço, pois não se usaria apenas o cilindro externo para gravar dados, mas praticamente todo o interior do cilindro no qual vários discos eram empilhados, como mostrado na [Figura 6.12](#).



**Figura 6.12: IBM 350 RAMAC.<sup>19</sup>**

Os primeiros computadores a terem esse sistema foram o IBM 305 RAMAC e o IBM 650 RAMAC, em que RAMAC significa “Random Access Method of Accounting and Control”.<sup>20</sup> O sistema de discos em si era chamado de IBM 350 e IBM 355, respectivamente para o 305 ou 650.

O conjunto de 50 discos tinha uma capacidade de armazenamento de 5 milhões de palavras de 6 bits, ou seja, aproximadamente 3,75 megabytes (mB, ou seja “milhões de bytes”). Um dos discos individuais é mostrado na [Figura 6.13](#).



**Figura 6.13: Um dos discos do 350.<sup>21</sup>**

Um IBM 305 com uma unidade de disco 350 era alugado em 1957 por cerca de 3.200 dólares mensais. O preço de aquisição seria algo na ordem de 160 mil dólares, ou seja, 42.600 dólares por megabyte. No momento em que este texto foi escrito (novembro de 2015), um HD externo com 1 terabyte (um milhão de megabytes) custava no Brasil 250 reais, ou seja, um valor de 0,00025 reais por megabyte. Mais de mil sistemas 350 foram construídos, até finalmente serem considerados obsoletos em 1962.

## **6.17 Logic Theorist – 1956**

No início dos anos 1950, Herbert A. Simon (Estados Unidos, 1916-2001), futuro ganhador do prêmio Nobel de Economia por sua teoria da racionalidade limitada em tomada de decisões, estava prestando consultoria na Rand Corporation (sem relação com a Remington Rand). Durante uma das visitas, ele viu uma impressora que estava imprimindo um mapa com caracteres normais e sinais de

pontuação. Ao observar isso, percebeu o quanto computadores poderiam ser usados para manipular símbolos e não apenas números.

O programa que imprimia tal mapa foi escrito por Allen Newell (Estados Unidos, 1927-1992) da Rand, o qual apaixonou-se pela ideia de usar computadores para manipular símbolos após assistir a uma palestra sobre correspondência de padrões.

Newell e Simon passaram a trabalhar juntos de 1955 a 1956, examinando a possibilidade de que computadores pudessem pensar. Eles resolveram trabalhar em um programa que fosse capaz de provar os teoremas do livro *Principia Mathematica* de Bertrand Russell. Eles obtiveram o apoio de John C. Shaw (1922-1991), programador de computadores da Rand.

Conseguiram idealizar e escrever um programa para realizar a tarefa de provar os teoremas. Esse programa ficou conhecido como Logic Theorist.<sup>22</sup> A primeira versão do programa foi escrita em um pseudocódigo e era executada de forma simulada por seres humanos. Eles juntaram familiares e alguns estudantes de graduação e cada um recebeu um cartão que descrevia uma das regras do programa. Dessa forma, simulavam a sua execução de forma mecânica como se fossem um computador. Mais tarde, Shaw conseguiu implementar o programa em um dos computadores da Rand.

O Logic Theorist provou 38 dos 52 problemas apresentados nos primeiros capítulos do *Principia*; inclusive para um deles o programa produziu uma prova mais elegante do que a de Russell. Eles tentaram publicar essa prova no *The Journal of Symbolic Logic*, mas o artigo foi rejeitado sob a alegação de que publicar uma nova prova para um teorema conhecido não tinha originalidade suficiente. Provavelmente os avaliadores não se deram conta de que a prova tinha sido a primeira criação totalmente original produzida por um computador.

Em 1956, John McCarthy, Marvin Minsky, Claude Shanon e Nathan Rochester organizaram a primeira conferência sobre inteligência artificial, termo que até então não existia e que foi cunhado por McCarthy especialmente para a reunião. Newell e Simon compareceram com o objetivo de apresentar o Theorist, mas foram recebidos sem qualquer entusiasmo. Aparentemente, os pais da inteligência artificial não foram capazes de perceber que naquele momento Newell e Simon já tinham conseguido realizar o que eles próprios ainda buscavam.

O Theorist era um programa que usava uma técnica que até hoje é padrão em sistemas de inteligência artificial: a “busca”. Um programa baseado em busca parte normalmente de um estado inicial, que no caso poderia ser um dos axiomas da lógica, e a cada instante aplica uma regra de transformação tomada de um conjunto de regras válidas. Por exemplo, o axioma inicial pode ser transformado a partir de alguma regra de transformação lógica como “modus ponens”: Se  $P \rightarrow Q$  é verdadeiro e  $P$  é verdadeiro, então  $Q$  é verdadeiro. Assim, uma expressão como “ $P$  e  $P \rightarrow Q$ ” poderia ser transformada em  $Q$ , mantendo-se verdadeira.

O problema que eles perceberam imediatamente era que o número de regras que se poderia aplicar era relativamente grande. Ora, partindo de um estado inicial, digamos que apenas 10 regras existam, então as transformações possíveis em um passo são 10, em dois passos são 100 e em três passos, 1.000. Ou seja, o número de transformações possíveis é de  $10^{\text{passos}}$ . Esse crescimento exponencial logo torna o processo de busca inviável, mesmo para o mais rápido dos computadores. Seria necessário decuplicar a velocidade da máquina apenas para poder avançar um único passo na busca no mesmo tempo.

Para solucionar esta questão, eles estabeleceram uma técnica que desde então é usada para que se possa tratar de problemas do tipo: a heurística. Uma heurística nada mais é do que um critério para eliminar determinados caminhos na busca. Determinadas regras

podem até ser aplicáveis, mas é possível ignorá-las ou pelo menos não as priorizar em função de outras mais promissoras dentro de dado critério.

A linguagem que Newell, Simon e Shaw criaram para escrever o Logic Theorist foi chamada de IPL, Information Processing Language. Era uma linguagem de manipulação de símbolos (não apenas números) baseada em listas e funções, um paradigma de programação que acabou usado depois por John McCarthy para a definição da linguagem LISP.

## 6.18 MUSIC – 1957

Embora o CSIRAC tenha sido o primeiro computador a gerar música e o Ferranti Mark I o primeiro a produzir música que foi gravada, o primeiro programa especificamente projetado para ajudar usuários de computador a produzirem sons musicais foi o MUSIC, criado pelo engenheiro Max Mathews (Estados Unidos, 1926-2011) da Bell Labs em 1957.

O MUSIC foi um primeiro protótipo ou prova de conceito para uma série de outros programas com grande repercussão e uso. Ele foi inicialmente rodado em um computador IBM 704 na Bell.

Todos os programas da série eram baseados em uma biblioteca de rotinas para processamento de sinais e síntese de som, as quais eram invocadas através de códigos de operação (opcodes). Esses códigos podiam ser programados pelo usuário em um arquivo de texto de modo a produzir a textura de um instrumento, o que definiria o tipo de som a ser tocado. Podia ser, por exemplo mais parecido com um violino, flauta ou piano, ou ainda completamente original. Um segundo arquivo definia as notas que seriam tocadas: duração e tonalidade, além de outros parâmetros. Assim, a partir de 1957 já era possível que um usuário escrevesse música para ser reproduzida no computador usando uma linguagem de propósito específico que não era a linguagem de máquina.

A primeira peça tocada pelo IBM 704 foi composta por um colega de Max, Newman Guttman, e chamava-se “The Silver Scale”. Ela pode ser ouvida a partir de <http://opinionator.blogs.nytimes.com/2011/06/08/the-first-computer-musician/>.

Consta que em 1961 Max realizou outra façanha ao fazer uma máquina sintetizar voz humana cantando a música “Bicycle Build for Two”, que também pode ser ouvida no site mencionado anteriormente. Dizem que Stanley Kubrick ficou tão impressionado ao ouvir essa voz sintetizada que resolveu usar parte da melodia, ficando conhecida como “Daisy, Daisy” na sequência em que a memória do computador HAL 9000 começa a perder a consciência ao ser lentamente desligado no filme *2001: Uma odisseia no espaço*. HAL 9000 estaria cantando, assim, uma música que ele teria aprendido na “infância”. Outra curiosidade bem conhecida sobre este computador ficcional é que “HAL” é uma codificação criptográfica ao estilo de Al-Kindi para “IBM”, na qual se substitui cada letra pela imediatamente anterior no alfabeto.

## 6.19 Scanner – 1957

Mais ou menos na mesma época em que os computadores aprendiam a cantar, eles também aprenderam a ver. A invenção do scanner é reputada a Russell Kirsch (Estados Unidos, 1929), que trabalhava em 1957 com o SEAC, ou Standards Electronic Automatic Computer, o primeiro computador programável usado pelo governo federal norte-americano para fins civis.

O scanner de computador pode ser considerado um descendente das máquinas fotográficas e aparelhos de fac-símile (fax), bem como das telefotos; tudo tecnologia desenvolvida a partir do século XIX. Porém, a novidade do scanner construído por Kirsch e sua equipe estava no fato de que pela primeira vez um computador guardou na memória a representação binária de uma imagem, ou seja, uma foto digital.

A primeira imagem digitalizada da história é mostrada na [Figura 6.14](#), e trata-se da digitalização de uma foto de um dos filhos de Kirsch, na época com 3 meses. A foto foi digitalizada em 176 por 176 bits e, como cada bit pode representar apenas 0 ou 1 – ou seja, preto ou branco –, isso impossibilitava tons de cinza.



Figura 6.14: A primeira imagem digital.<sup>23</sup>

Porém, a ideia de digitalizar imagens usando apenas bits com 0 ou 1 não foi uma boa escolha. Quando os engenheiros reproduziram a foto digitalizada em um tubo de raios catódicos, perceberam esse erro. Depois disso, passaram a realizar várias passagens do scanner sobre a imagem com diferentes níveis de sensibilidade à luz. Dessa forma puderam construir uma figura muito mais realista usando tons de cinza em vez de preto e branco.

## 6.20 FORTRAN – 1957

Apesar de Zuse ter concebido a primeira linguagem de programação de alto nível em 1946, ela nunca foi implementada até a década de 1970. A primeira linguagem de programação científica de alto nível implementada foi, de fato, FORTRAN, um acrônimo para FORMula TRANslating.

FORTRAN foi concebida por John Backus (Estados Unidos, 1924-2007), segundo ele, por preguiça de ter que escrever tantos comandos para realizar as operações mais simples no computador. De fato, com FORTRAN a quantidade de comandos que precisava ser programada para atingir o mesmo objetivo era cerca de 20 vezes menor.

Backus começou a programar em 1950 no IBM SSEC. No final de 1953, ele escreveu um memorando ao seu chefe com a proposta de desenvolver uma linguagem de programação para o novo IBM 701, que facilitaria o trabalho de programação.

A proposta foi aprovada e ele ganhou inclusive uma equipe de quatro pessoas para ajudá-lo. Em 1954, com a disponibilidade do IBM 704, Backus passou a focar nesta máquina.

Ainda em 1954, ele e sua equipe lançaram um documento onde especificavam a linguagem cujo compilador (na época chamado de “tradutor”) deveria ser lançado em cerca de seis meses. Porém, o trabalho demorou um pouco mais que o esperado, e em 1957 o primeiro compilador de FORTRAN, ainda cheio de bugs, era disponibilizado ao mundo.

O primeiro compilador FORTRAN consistia de 25 mil linhas de código (de máquina). Toda máquina IBM 704 vinha com uma fita magnética contendo o FORTRAN e um manual de 51 páginas ([Figura 6.15](#)).

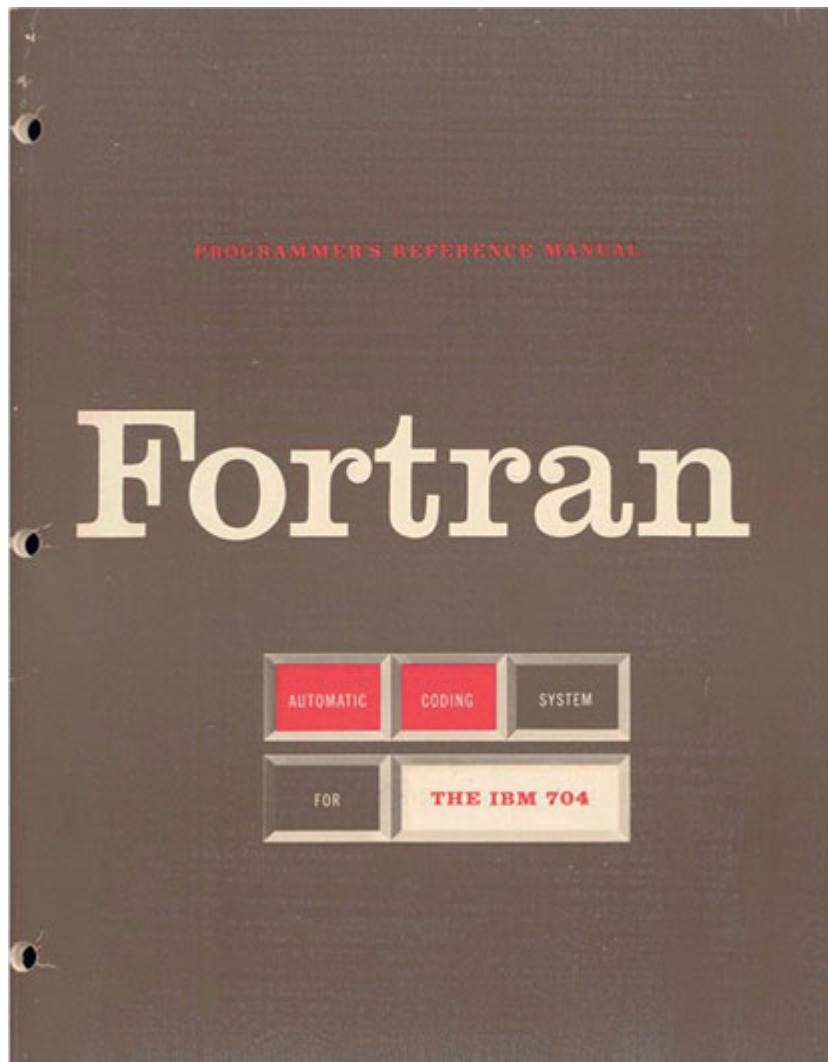


Figura 6.15: Primeiro manual de Fortran.<sup>24</sup>

Na proposta de 1954, Backus já previa que a linguagem tornaria o processo de programação muito mais rápido. O maior risco e fator de rejeição a um possível compilador de linguagem de alto nível vinham do fato de que muitos programadores duvidavam que um computador pudesse gerar código de máquina através do processo de compilação que fosse mais eficiente do que um código de máquina produzido por um humano.

A realidade, entretanto, mostrava que os humanos tinham grande dificuldade de produzir código de máquina de boa qualidade e,

assim, de fato a máquina se saia muito melhor, gerando programas em código de máquina mais eficientes do que os humanos seriam capazes.

Algumas características da proposta inicial do FORTRAN:

- Números inteiros, como 56, +900 e -167, seriam armazenados em variáveis nomeadas com um ou dois caracteres alfanuméricos, iniciando pelas letras i, j, k, l, m ou n. Por isso até hoje muitos programadores fazem suas variáveis de índice (inteiros) serem i, j, k etc.
- Números de ponto flutuante, como .02, 12.45 e -9.4, seriam armazenados em variáveis nomeadas com um ou dois caracteres alfanuméricos iniciados por qualquer letra exceto i, j, k, l, m ou n.
- Operadores binários: +, -, ×, / e  $\times\times$  (exponenciação).
- Funções predefinidas e definíveis pelo usuário, como  $\sin(x)$  - seno,  $\sqrt{x}$  – raiz quadrada,  $\text{factl}(x)$  – factorial,  $\max(a, b, c, d, e)$  - o maior dentre a, b, c, d e e.
- Expressões combinando constantes, variáveis e operadores, desde que tivessem no máximo 750 caracteres.
- Uso de parênteses aninhados em expressões complexas. Por exemplo:  $((a+b)^*c+d)/e$ .
- Variáveis subscritas, hoje chamadas de vetores ou arrays. Por exemplo  $A(n)$  significava  $A_n$ .
- Declaração de arrays ou matrizes pelo comando DIMENSION. Por exemplo, DIMENSION v[2,3] definia uma matriz v com duas linhas por três colunas, as quais podiam ser referenciadas por variáveis subscritas como  $v(1,1)$ ,  $v(1,2)$  etc.
- Números de fórmulas, hoje chamados de rótulos ou números de linha. Um comando em FORTRAN poderia opcionalmente ter um número antes dele para ser referenciado por comandos de repetição ou decisão. Por exemplo, em 45  $x=a+b$ , aplica-se o

rótulo 45 ao comando  $x=a+b$  (atribuir o valor de  $a+b$  à variável  $x$ ). Assim, esse comando passa a ser referenciado como o comando 45.

- Estruturas de repetição, ou DO (para-faça). O formato originalmente proposto seria algo do tipo DO 10, 14, 50  $i=4, 20, 2$  significando o seguinte: repita os comandos desde o número 10 até o número 14 um certo número de vezes. Depois disso vá para o comando 50. O número de vezes é governado pela variável de loop  $i$ , que vai valer 4 da primeira vez e depois ter seu valor somado a 2, cada vez que os comandos forem repetidos. Quando o valor de  $i$  se tornar superior a 20 deve-se parar a repetição. Assim,  $i$  vai assumir os valores 4, 6, 8, 10, 12, 14, 16, 18, 20 e a repetição, no caso, será feita 9 vezes.
- Estrutura de decisão (IF). Embora as versões mais antigas de FORTRAN usassem IF numérico (decidir para onde ir dependendo do sinal de uma variável), a proposta de 1954 já previa o IF lógico, com instruções como: IF ( $a \geq b$ ) 23, 56, ou seja, se  $a$  for maior ou igual a  $b$  vá para o comando 23. Senão vá para o comando 56.
- GOTO. Um comando hoje em desuso em função dos princípios da programação estruturada, mas que fazia o programa ir diretamente para a instrução cujo número era passado logo depois. Por exemplo, GOTO 56 ia diretamente para o comando 56.
- STOP, que fazia o computador parar.
- Instruções de entrada e saída: *read*, *punch*, *print*, *read tape*, *write tape*, *end file*, *rewind*, *backspace*, *read drum* e *write drum*.

FORTRAN foi durante mais de 30 anos a principal linguagem de programação científica e ainda hoje é usada em muitas áreas onde se necessita de alta precisão e eficiência.

Uma característica de FORTRAN que Backus já visualizava em 1954 era o fato de que as linguagens de máquina eram diferentes

de uma máquina para outra, mas um programa escrito em FORTRAN poderia rodar em qualquer máquina desde que compiladores específicos fossem construídos. Assim, quando um usuário trocasse um modelo de computador por outro, não precisaria mais reescrever todos os seus programas. Uma vantagem enorme, hoje conhecida como “portabilidade”.

O problema é que muitos compiladores FORTRAN levemente diferentes foram feitos, os quais eram, portanto, incompatíveis. Isso levou a American National Standards Institute (ANSI, que em português pode ser traduzido como Associação Americana de Padrões) a estabelecer um padrão para a linguagem FORTRAN em 1966, o qual ficou conhecido como FORTRAN IV. Ele foi revisado em 1977 com o lançamento do FORTRAN 77 e, por último em 1990, com o FORTRAN 90.

## 6.21 ALGOL – 1958

Nos anos de 1951 a 1954 algumas tentativas de desenvolver linguagens de alto nível, também chamadas de linguagens algorítmicas, foram feitas na Europa embora sem grandes avanços de início. Porém, essas tentativas frustradas prepararam os espíritos para a chegada de FORTRAN que, embora ainda não implementado, já tinha seu primeiro relatório amplamente divulgado desde 1954.

Ao mesmo tempo em que a indústria, através da IBM e da Bell trabalhava para criar uma linguagem de alto nível para aplicações científicas, a academia não estava parada. De fato, já em 1947 tinha sido fundada nos Estados Unidos a Association for Computing Machinery – ACM –, a qual organizava regularmente encontros e simpósios.

Na Europa, um simpósio internacional sobre computação automática foi planejado em 1955 na Alemanha, em Darmstadt. “Computação automática” ou “programação automática” era o nome que se dava à programação em alto nível naquela época.

Vários participantes dessa conferência concordaram que era necessário definir uma linguagem algorítmica, independente de máquina, universal e única para ser usada por todos, ao invés de existirem várias linguagens diferentes.

Assim, na mesma conferência foi criado o comitê GAMM (uma sigla em alemão que significa “Sociedade para Matemática e Mecânica Aplicadas”) para estudar esse assunto e apresentar propostas.

Porém, em 1957 quando quase completavam seu trabalho na especificação de uma nova linguagem algorítmica, os membros do GAMM se deram conta de que ela seria apenas mais uma em meio a outras. Assim, eles teriam é que se esforçar para unificar as diferentes vertentes. Buscando um maior diálogo com os Estados Unidos, convidaram a ACM para realizar uma conferência conjunta com a finalidade de discutir a proposta de uma linguagem universal.

Ao compararem a proposta do GAMM com as elaboradas pela ACM, encontraram muitos pontos em comum e, assim, a conferência para unificação das propostas foi realizada em Zurique em meados de 1958. Estavam presentes quatro membros da ACM e quatro do GAMM. Os princípios para a elaboração da linguagem universal seriam os seguintes:

- Ela seria a mais parecida possível com a linguagem matemática, de forma que pudesse ser lida por matemáticos com o mínimo esforço de interpretação.
- Deveria ser possível usar a linguagem para descrever processos numéricos em publicações.
- A nova linguagem teria que ser imediatamente traduzível para código de máquina pela própria máquina.

A conferência de Zurique produziu, dessa maneira, o relatório do ALGOL 58. Como havia pouquíssima padronização em relação a comandos de entrada e saída de dados, decidiram deixar de lado este aspecto da computação e se concentrar apenas nas questões mais abstratas, o que depois revelou ser um erro, pois foi um dos fatores que fizeram com que ALGOL, a Linguagem Algorítmica

(ALGOrithmic Language) não se tornasse tão popular quanto FORTRAN ou COBOL na ciência e no comércio.

O primeiro compilador ALGOL foi criado ainda em 1958 por Friedrich L. Bauer (Alemanha, 1924-2015) para rodar no computador Z22, um computador a válvulas, o sétimo modelo construído por Konrad Zuse.

Várias propostas de melhorias foram publicadas na revista *Communications of the ACM* em 1959 (volume 2, número 7). Essa edição ficou conhecida como “Boletim ALGOL”.

Em 1959, o GAMM reuniu 50 membros em Paris para selecionar sete para o comitê final de ALGOL. Nos EUA, a ACM também realizou um encontro para selecionar sete outros delegados. Finalmente, em janeiro de 1960 ocorre em Paris a conferência que produziu o relatório ALGOL 60. Entre outras pessoas, estiveram na conferência John Backus (criador do FORTRAN), Peter Naur e John McCarthy.

Em 1960 ocorreu ainda a fundação da International Federation for Information Processing (IFIP), uma federação de sociedades científicas de vários países do mundo, da qual a ACM tornou-se membro. Em 1962 a manutenção e desenvolvimento do ALGOL foi assumido pela IFIP, em especial, pelo grupo de trabalho WG 2.1 (Linguagens Algorítmicas e Cálculo).

Uma das diferenças de ALGOL em relação a FORTRAN está no fato de que em ALGOL as variáveis devem ter seu tipo declarado. Em FORTRAN é a primeira letra do nome da variável que define o tipo dela. Assim, por exemplo, uma variável iniciada com *i* é sempre inteira e uma variável iniciada por *x* é sempre real. Em ALGOL seria necessário declarar antes de usar a variável se era inteira ou real, e sua letra inicial poderia ser qualquer uma. Isso parece desvantajoso à primeira vista, mas tem seu valor porque com ALGOL uma variável que representasse um contador poderia ser declarada como “*contador*”, enquanto que em FORTRAN seu nome teria que iniciar

com uma letra entre *i* e *m* para ser inteira, como por exemplo “*icontador*”, o que soa estranho.

Além disso, o fato de se poder declarar o tipo da variável também abriu caminho para que as linguagens permitissem que o programador criasse seus próprios tipos personalizados. Assim, em vez de ter apenas inteiros e reais, você poderia contar com booleanos, textos ou mesmo temperaturas, CPF, datas, ISBN etc.

Segue abaixo um exemplo de programa em ALGOL:

```
proc abs max = ([,]real a, ref real y int i, k)real:  
comment O maior elemento absoluto da matriz a, de tamanho  $\Box a$   
por  $2\Box a$  é armazenado em y e os índices deste elemento em i e k;  
comment  
begin  
real y:=0; i:= $\Box a$ ; k:= $2\Box a$ ;  
for p from  $\Box a$  to  $\Box a$  do  
for q from  $2\Box a$  to  $2\Box a$  do  
if abs a[p,q] > y then  
y:=abs a[p,q];  
i:=p; k:=q  
fi  
od  
od;  
y  
end # abs max #
```

Para descrever a estrutura da linguagem ALGOL, John Backus criou uma notação que depois foi aperfeiçoada por Peter Naur e ficou conhecida como Notação Backus-Naur (BNF, de “Backus-Naur Form”). Até hoje ela é a notação padrão para especificação de linguagens de programação.

ALGOL praticamente não é mais usada, mas deixou influências em muitas outras linguagens de programação correntes como C, Pascal, Java e Python.

## 6.22 *Tênis para Dois* – 1958

O primeiro jogo eletrônico para computador com animação em tela foi criado em 1958 por William Higinbotham (Estados Unidos, 1910-1994). Tratava-se de uma simulação de jogo de tênis, que ficou conhecida como *Tennis for Two*, ou *Tênis para dois*.

Consta que Higinbotham resolveu criar o jogo para reduzir a monotonia que sentiam os visitantes do Laboratório Nacional Brookhaven, onde ele trabalhava.

Ele usou um computador analógico Donner Model 30 ligado a um osciloscópio, que é um instrumento com uma tela de vídeo CRT. Com este osciloscópio, ele desenhava na tela um jogo de tênis visto de perfil ([Figura 6.16](#)) com uma rede, o chão e uma “bolinha” representada por um ponto que ficava pulando de um lado para o outro da tela, deixando um pequeno rastro atrás de si.

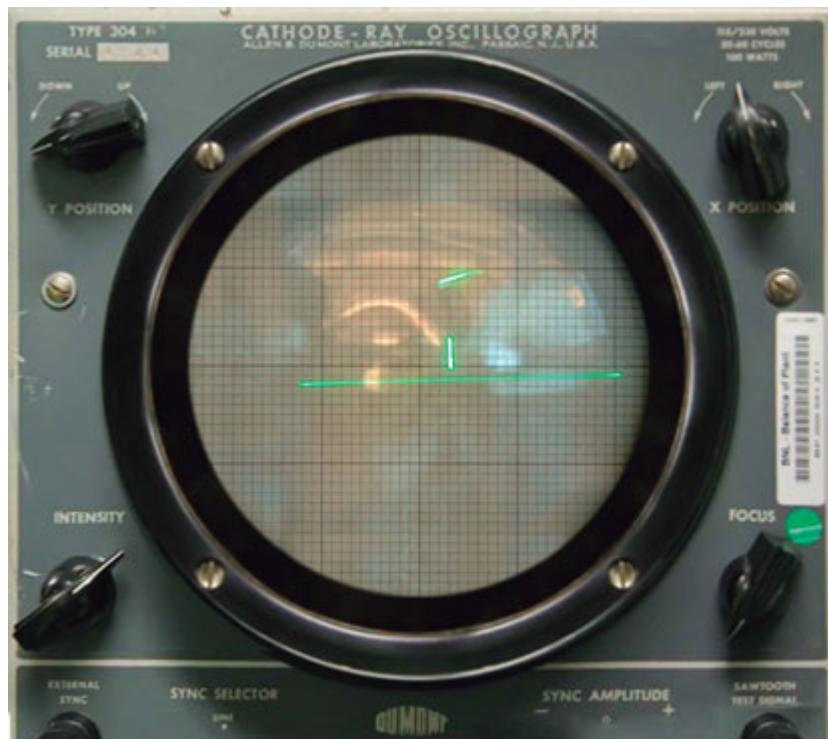


Figura 6.16: *Tênis para dois.*<sup>25</sup>

Higinbotham conta que teve a ideia quando soube que o computador era capaz de calcular a trajetória de projéteis. Daí ele usou os circuitos que faziam estes cálculos para mover a “bolinha” na tela do osciloscópio. O circuito podia sentir quando a bolinha tocava o chão, para inverter seu movimento, e também se ela tocasse a rede. Os jogadores usavam um botão de girar (potenciômetro) para definir o ângulo da raquete (que não aparecia na tela) e um botão de pressionar para indicar o momento da raquetada. Quando a bolinha era atingida, a máquina emitia um bip. Consta que o projeto levou apenas quatro horas para ser desenhado e cerca de duas semanas para ser construído.

Como as demais exibições do laboratório eram basicamente estáticas, o *Tênis para Dois* rapidamente tornou-se a vedete das exposições, e centenas de pessoas faziam fila para jogar.

Porém, após algumas poucas exibições ele foi desmantelado e ficou esquecido por mais de 20 anos até que seu autor foi convocado a depor como testemunha em um processo judicial entre duas produtoras de videogames: a Atari e a Odyssey. A partir dessa redescoberta, Higinbotham foi reconhecido como o avô dos videogames.

## 6.23 LISP – 1958

FORTRAN é a mais antiga linguagem de programação ainda em uso. A segunda mais antiga ainda em uso é uma linguagem que foi criada especificamente para manipulação de símbolos e não de números. Essa linguagem chama-se LISP (LISt Processing) e foi criada por John McCarthy, personagem já citado algumas vezes nas seções anteriores.

Lembram-se de quando falamos na Tese Church-Turing? Turing criou uma definição de computabilidade baseada em sua Máquina A, e Church também criou uma baseada em um princípio bem diferente, chamado Cálculo  $\lambda$  (Lambda). O Cálculo Lambda é uma linguagem baseada em substituições de símbolos. Em vez de simplesmente operar numericamente com variáveis como usualmente fazemos na aritmética, com o Cálculo Lambda trabalha-se diretamente com os símbolos. Por exemplo, pode-se escrever uma expressão que define que  $a(x+y)$  é equivalente a  $ax+ay$ . Fazendo substituições de símbolos, assim, a partir de suas definições, o Cálculo Lambda serve como um mecanismo de simplificação, que vai reduzindo as expressões até um formato considerado final: a saída do programa.

Pois bem, inspirado na notação de Church e também na linguagem IPL usada por Newell, Simon e Shaw para implementar o Logic Theorist, McCarthy, nessa época trabalhando no MIT, publicou o artigo “Recursive Functions of Symbolic Expressions and their Computation by Machine, Part I”,<sup>26</sup> escrito em 1958 e publicado na revista *Communications of the ACM* em 1960. Quanto à parte II, se

é que foi escrita, nunca foi publicada. McCarthy mostrou que era possível construir uma linguagem baseada em transformações sobre listas e funções e que essa linguagem simbólica seria Turing-completa.

O compilador LISP foi implementado pela primeira vez em um IBM 704 por Stephen Russell (Estados Unidos, 1957), o qual também ficou famoso anos depois pela implementação do videogame *Spacewar!*. Porém, essa primeira implementação do LISP era apenas uma prova de conceito.

A primeira execução completa de LISP foi feita em 1962 por Tim Hart e Mike Levin no MIT. Desde sua criação, LISP foi uma linguagem fortemente utilizada pela comunidade de Inteligência Artificial. Vários sistemas de Inteligência Artificial (IA) foram implementados nela.

LISP foi a primeira linguagem a permitir o uso de “recursão”, um conceito que não era permitido pelas primeiras implementações de FORTRAN. Uma função é recursiva se ela for definida em termos dela própria. Por exemplo, a função “fatorial” pode ser definida de forma iterativa como  $n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$ . Assim,  $5! = 5 \times 4 \times 3 \times 2 \times 1$ . Entretanto, a função fatorial também pode ser definida de forma recursiva como  $n! = n \times (n-1)!$ . Assim,  $5! = 5 \times 4!$ . Porém, para que a recursão não continue indefinidamente com a função sempre chamando a si própria é necessário estabelecer que a expressão acima só se aplica se  $n$  for maior do que 0. Define-se, adicionalmente, então, que o fatorial de 0 é 1:  $0! = 1$ .

FORTRAN não permitia que o nome de uma função fosse usado em sua definição; assim, funções definidas recursivamente eram impossíveis. LISP permitia, contudo. O programa abaixo, escrito em uma versão moderna do LISP, é uma definição recursiva da função fatorial:

```
DEFINE ((  
    (FATORIAL (LAMBDA (N)
```

```
(COND ((EQUAL N 0) 1)
      (T (TIMES N (FACTORIAL (SUB1 N))))))
    ))
```

A quantidade de parênteses usada em programas LISP frequentemente fazia os programadores referirem-se aos seus programas como “selvas de parênteses”. O uso intensivo desse caractere é devido ao fato de que todas as expressões da linguagem serem, a princípio, expressões baseadas em listas. Se você quiser somar  $x+y+z$  em LISP teria que escrever (SUM X Y Z). Porém, versões mais modernas da linguagem acabaram permitindo que a notação usual  $x+y+z$  fosse usada em muitas situações.

## 6.24 Até Aqui...

Vimos que a década de 1950 testemunhou o início da computação comercial acessível às empresas e demais organizações. Nessa década surgiram os sistemas operacionais, o disco magnético e os primeiros computadores transistorizados.

Três importantes linguagens de programação nasceram nos anos 1950: FORTRAN, ALGOL e LISP. Duas delas ainda são bastante usadas e a outra (ALGOL) deixou vários descendentes que hoje são dominantes na indústria. Na área comercial, a década viu também a evolução do “compilador” A-0, que gerou o FLOW-MATIC, o qual acabou sendo a principal influência para a definição da linguagem padrão do comércio, o COBOL, logo no início dos anos 1960.

Durante os anos 1950 viu-se também que era possível usar computadores para produzir música e reconhecer imagens. A inteligência artificial, que começou a ser imaginada como possibilidade nos anos 1940, vê surgir seu primeiro triunfo: o Logic Theorist.

Em termos de equipamentos, os principais fornecedores dessa década foram a IBM, com seus modelos 701, 650 e 704, e a Remington Rand, com o UNIVAC. Em termos de computador

pessoal, merece menção o Simon que, com um processador de apenas 2 bits, foi o primeiro equipamento de computação acessível ao cidadão comum.

Toda essa evolução – especialmente a do transistor – das memórias de núcleo magnético, dos sistemas operacionais e da computação de tempo real preparou o caminho para desenvolvimentos ainda maiores na década seguinte.

---

<sup>1</sup> Tradução: *O uso humano dos seres humanos*.

<sup>2</sup> Tradução: “Nós modificamos nosso ambiente tão radicalmente que precisamos agora modificar a nós mesmos para existir nesse novo ambiente.”

<sup>3</sup> Tradução: *Cérebros gigantes, ou máquinas que pensam*.

<sup>4</sup> “Univac I at CHM.agr” by ArnoldReinhold - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Univac\\_I\\_at\\_CHM.agr.jpg#/media/File:Univac\\_I\\_at\\_CHM.agr.jpg](https://commons.wikimedia.org/wiki/File:Univac_I_at_CHM.agr.jpg#/media/File:Univac_I_at_CHM.agr.jpg)

<sup>5</sup> “Mercury memory”. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Mercury\\_memory.jpg#/media/File:Mercury\\_memory.jpg](https://commons.wikimedia.org/wiki/File:Mercury_memory.jpg#/media/File:Mercury_memory.jpg)

<sup>6</sup> “CSIRAC-Pano,-Melb.-Museum,-12.8.2008” by jjron - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:CSIRAC-Pano,-Melb.-Museum,-12.8.2008.jpg#/media/File:CSIRAC-Pano,-Melb.-Museum,-12.8.2008.jpg>

<sup>7</sup> “Museum of Science, Boston, MA - IMG 3168” by Daderot - Own work. Licensed under Public Domain via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Museum\\_of\\_Science,\\_Boston,\\_MA\\_-\\_IMG\\_3168.JPG#/media/File:Museum\\_of\\_Science,\\_Boston,\\_MA\\_-\\_IMG\\_3168.JPG](https://commons.wikimedia.org/wiki/File:Museum_of_Science,_Boston,_MA_-_IMG_3168.JPG#/media/File:Museum_of_Science,_Boston,_MA_-_IMG_3168.JPG)

<sup>8</sup> “Coincident-current magnetic core” by Tetromino - Own work / собственная работа. Licensed under CC BY 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Coincident-current\\_magnetic\\_core.svg#/media/File:Coincident-current\\_magnetic\\_core.svg](https://commons.wikimedia.org/wiki/File:Coincident-current_magnetic_core.svg#/media/File:Coincident-current_magnetic_core.svg)

<sup>9</sup> “KL Kernspeicher Makro 1” by Konstantin Lanzet - received per EMailCamera: Canon EOS 400D. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:KL\\_Kernspeicher\\_Makro\\_1.jpg#/media/File:KL\\_Kernspeicher\\_Makro\\_1.jpg](https://commons.wikimedia.org/wiki/File:KL_Kernspeicher_Makro_1.jpg#/media/File:KL_Kernspeicher_Makro_1.jpg)

<sup>10</sup> By Unknown (Smithsonian Institution) - Flickr: Grace Hopper and UNIVAC, CC BY 2.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=19763543>

<sup>11</sup> Tradução: “Eu tinha um compilador que funcionava e ninguém queria tocar nele porque, e isso eles me falavam com muito cuidado, computadores poderiam lidar apenas com aritmética, não com programas.”

<sup>12</sup> “IBM 701console” by Dan - Flickr: IBM 701. Licensed under CC BY 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:IBM\\_701console.jpg#/media/File:IBM\\_701console.jpg](https://commons.wikimedia.org/wiki/File:IBM_701console.jpg#/media/File:IBM_701console.jpg)

<sup>13</sup> “Princeton IAS computer” by National Museum of American History - Disponível em: [http://americanhistory.si.edu/collections/search/object/nmah\\_334741](http://americanhistory.si.edu/collections/search/object/nmah_334741). Licensed under Public Domain via Commons. [https://commons.wikimedia.org/wiki/File:Princeton\\_IAS\\_computer.jpg#/media/File:Princeton\\_IAS\\_computer.jpg](https://commons.wikimedia.org/wiki/File:Princeton_IAS_computer.jpg#/media/File:Princeton_IAS_computer.jpg)

<sup>14</sup> “IBM 650 panel close-up of bi-quinary indicators” by Mfc - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:IBM\\_650\\_panel\\_close-up\\_of\\_bi-quinary\\_indicators.jpg#/media/File:IBM\\_650\\_panel\\_close-up\\_of\\_bi-quinary\\_indicators.jpg](https://commons.wikimedia.org/wiki/File:IBM_650_panel_close-up_of_bi-quinary_indicators.jpg#/media/File:IBM_650_panel_close-up_of_bi-quinary_indicators.jpg)

<sup>15</sup> Tradução: “Esta série de livros é afetuosaamente dedicada ao Computador Modelo 650 que foi instalado no Instituto de Tecnologia Case, com o qual eu passei muitas noites agradáveis.”

<sup>16</sup> Tradução: Operação simultânea de periféricos em linha.

<sup>17</sup> São os sinais que permitem que o computador sincronize a execução de suas instruções. Em princípio, um ciclo de clock mais rápido implica que o computador vai executar suas instruções mais rapidamente também.

<sup>18</sup> “HarwellCadetComputer” by MichaelWilson78 - Own work. Licensed under Public Domain via Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:HarwellCadetComputer.jpg#/media/File:HarwellCadetComputer.jpg>

<sup>19</sup> “IBM 350 RAMAC” by Transferred from en.wikipedia to Commons by DingirXul.. Licensed under CC BY-SA 2.5 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:IBM\\_350\\_RAMAC.jpg#/media/File:IBM\\_350\\_RAMAC.jpg](https://commons.wikimedia.org/wiki/File:IBM_350_RAMAC.jpg#/media/File:IBM_350_RAMAC.jpg)

<sup>20</sup> Tradução: Método de acesso aleatório para contabilidade e controle.

<sup>21</sup> “RAMAC 305 disk” by I, *Deep silence. Licensed under CC BY 2.5 via Commons.* Disponível em:

[https://commons.wikimedia.org/wiki/File:RAMAC\\_305\\_disk.JPG#/media/File:RAMAC\\_305\\_disk.JPG](https://commons.wikimedia.org/wiki/File:RAMAC_305_disk.JPG#/media/File:RAMAC_305_disk.JPG)

<sup>22</sup> Tradução: Teórico lógico.

<sup>23</sup> “NBSFirstScanImage”. *Licensed under Public Domain via Commons.* Disponível em:

<https://commons.wikimedia.org/wiki/File:NBSFirstScanImage.jpg#/media/File:NBSFirstScanImage.jpg>

<sup>24</sup> “Fortran acs cover” by original uploader was en:User:Muhandis - en:File:Fortran acs cover.jpeg. *Licensed under Public Domain via Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Fortran\\_acs\\_cover.jpeg#/media/File:Fortran\\_acs\\_cover.jpeg](https://commons.wikimedia.org/wiki/File:Fortran_acs_cover.jpeg#/media/File:Fortran_acs_cover.jpeg)

<sup>25</sup> “Tennis For Two on a DuMont Lab Oscilloscope Type 304-A” by Brookhaven National Laboratory - Screenshot. *Licenced under Public Domain via Wikimedia Commons.* Disponível em:

[https://commons.wikimedia.org/wiki/File:Tennis\\_For\\_Two\\_on\\_a\\_DuMont\\_Lab\\_Oscilloscope\\_Type\\_304-A.jpg#/media/File:Tennis\\_For\\_Two\\_on\\_a\\_DuMont\\_Lab\\_Oscilloscope\\_Type\\_304-A.jpg](https://commons.wikimedia.org/wiki/File:Tennis_For_Two_on_a_DuMont_Lab_Oscilloscope_Type_304-A.jpg#/media/File:Tennis_For_Two_on_a_DuMont_Lab_Oscilloscope_Type_304-A.jpg)

<sup>26</sup> Tradução: “Funções recursivas de expressões simbólicas e sua computação por máquina, parte I.”

## **PARTE VII**

### **Do Circuito Integrado à Lua**

A década de 1960 viu o surgimento do circuito integrado, embora computadores a válvula e transistor ainda fossem construídos. Ela foi marcada pela invenção de várias linguagens de programação importantes, como COBOL e BASIC.

Entre 1960 e 1961, o Brasil construiu seus primeiros computadores. Também nesta década a IBM lança a primeira família de computadores totalmente compatíveis do primeiro ao último modelo, iniciando a era da portabilidade e do upgrade indolor. A década viu a alvorada das redes de computadores, em especial a invenção da comutação de pacotes e a construção da ARPANET, ancestral da Internet.

Conceitos de orientação a objetos começam a surgir, inicialmente com o Sketchpad e depois com o Simula 67 e o projeto Dynabook. Nessa década a programação e a engenharia de software tornam-se ciências mais sistemáticas, marcadas especialmente pelo tratado de Knuth sobre programação e as criações de Margareth Hamilton na área de engenharia de software, as quais permitiram que o ser humano fosse à Lua e voltasse com segurança.

#### **7.1 Circuito Integrado – 1960**

Ao longo da década de 1950 o uso do transistor sólido no lugar da válvula incandescente foi um grande avanço para a indústria dos computadores. Porém, à medida que as demandas por máquinas

cada vez mais potentes cresciam, os construtores se deparavam com dificuldades imensas para montá-las, visto que um único ponto de solda malfeito poderia inviabilizar totalmente o funcionamento do computador – e eram centenas de milhares de soldas necessárias em cada máquina. Além disso, à medida que os computadores ficavam maiores havia o problema do tempo que os sinais elétricos levavam para atravessar os fios cada vez mais compridos. Essa limitação também inviabilizava a construção de computadores acima de determinado tamanho.

Porém, a virada para 1960 viu nascer também um novo conceito: o circuito integrado, ou seja, a possibilidade de construir todo um conjunto de transistores a partir de um bloco único de material sólido, sem a necessidade de soldas e fios.

O primeiro idealizador desse conceito, pelo que se sabe, foi George W. A. Dummer (Reino Unido, 1909-2002). Ele trabalhava com engenharia de confiabilidade em um laboratório de pesquisas em telecomunicações. Seu trabalho era fazer com que circuitos baseados em transistores fossem cada vez mais confiáveis, o que era importante visto que naquela época eles definitivamente não eram. Ao verificar que isso com frequência era alcançado reduzindo o tamanho dos componentes, ele chegou à conclusão de que o próximo passo lógico seria produzir todo um conjunto de componentes a partir de um único bloco de cristal, como, por exemplo, silício. Em 1952, ele publicou essa ideia em uma conferência em Washington, sendo considerada a primeira publicação a mencionar circuitos integrados da história.

A ideia de Dummer era produzir várias camadas de elementos condutores, isolantes, retificadores e amplificadores que seriam coladas juntas. O circuito em si seria construído a partir de recortes feitos nessas camadas, de forma adequada.

Apenas em 1957 ele conseguiu produzir uma primeira prova de conceito, ao construir um flip-flop a partir de um bloco sólido de semicondutor preparado para funcionar como quatro transistores.

Porém, suas ideias não tiveram muita aplicação prática de início porque os britânicos não perceberam seu potencial ou não quiseram assumir o risco do projeto.

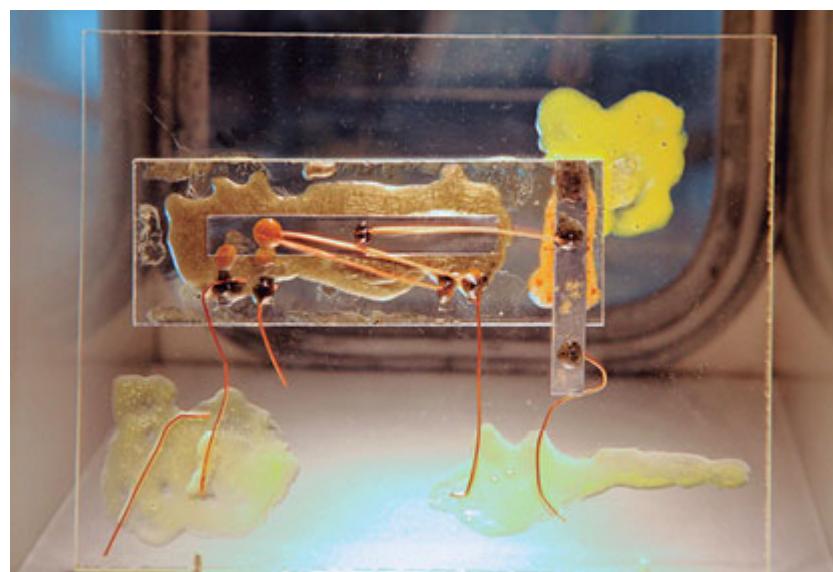
Entretanto, nos Estados Unidos, em 1958, havia pessoas dispostas a assumir esse risco. Um engenheiro chamado Jack Kilby (Estados Unidos, 1923-2005) estava trabalhando sozinho em seu laboratório na Texas Instruments, pois era recém-contratado pela empresa e não tinha direito a férias na época em que a grande maioria de seus colegas estavam fora. A Texas tinha desenvolvido em 1954 o primeiro rádio transistorizado de bolso, o qual foi um tremendo sucesso, e a empresa acreditava que as aplicações de circuitos desse tipo eram ilimitadas. Foi a Texas também que criou neste mesmo ano o transistor de silício, muito mais confiável do que o antigo transistor de germânio. Na mesma pesquisa, descobriram que outros componentes como diodos, resistores e capacitores também podiam ser feitos de silício, o que foi fundamental para a concepção do circuito integrado, visto que esse também precisaria desses demais componentes, não só transistores.

Como Kilby estava lá solitário, tinha muito tempo para pensar e, assim, ele chegou à conclusão de que poderia construir circuitos completos a partir de um único bloco. Essa ideia passou para a história como a “ideia monolítica”.

De início, o conceito enfrentou várias objeções. Considerando que o número de transistores normais que eram construídos e não passavam pelo controle de qualidade era absurdamente alto, construir uma peça única com vários transistores era, segundo as leis da probabilidade, uma certeza de falha.

Em 1960, no entanto, ele conseguiu produzir o primeiro circuito integrado, medindo 11 x 1,5 mm, e a Texas solicitou a patente do projeto neste mesmo ano. Uma réplica desse circuito é mostrada na [Figura 7.1](#). A placa transparente inteira tem o tamanho aproximado de uma unha humana.

Curiosamente em 1959, na Califórnia, outro inventor teve a mesma ideia de forma totalmente independente. Robert Noyce (Estados Unidos, 1927-1990) trabalhava na Fairchild Semiconductor, empresa que ele e mais sete colegas fundaram em 1957. De fato, o conceito de Noyce era até superior ao de Kilby, pois tornava bem mais fácil construir os circuitos, chamados por ele de “círculo unitário” – ao passo em que Kilby os nomeara “círculo sólido”. A Fairchild entrou com um pedido de patente um pouco depois da Texas, o que levou Noyce a ficar sabendo da outra iniciativa. Ele procurou detalhar bastante a sua proposta, deixando claro que não infringia nenhuma das ideias de Kilby.



**Figura 7.1: Réplica do primeiro circuito integrado a ser construído.<sup>1</sup>**

A patente de Noyce, embora solicitada depois, foi concedida antes da de Kilby, em 1961; Kilby só receberia o reconhecimento em 1964. Assim, a Texas e a Fairchild entraram em uma disputa judicial, que só foi resolvida por um acordo em 1966. Ambas as empresas reconheceram que de forma independente tinham inventado o circuito integrado. Elas poderiam produzir circuitos sem pagar

direitos uma à outra, mas outras empresas que quisessem produzi-los teriam que pagar *royalties* a ambas.

Hoje, tanto Kilby quanto Noyce são reconhecidos como os criadores do circuito integrado, mas o Prêmio Nobel de 2000 chegou tarde para Noyce, que já havia falecido 10 anos antes, e foi recebido por Kilby apenas.

A Texas começou a usar circuitos integrados em seus produtos ainda em 1960, e em 1963 o SN510 e SN514 foram os primeiros circuitos integrados a serem usados em órbita da Terra. A Fairchild também lançou seu primeiro circuito integrado comercial em 1960: um flip-flop. A partir de 1962, outras empresas começaram também a produzi-los, entre elas a Motorola e a Signetics.

## 7.2 COBOL – 1960

Em abril de 1959, um grupo de pessoas da academia, fabricantes e usuários de computador reuniu-se no centro de computação da Universidade da Pensilvânia. Entre outras coisas, eles chegaram à conclusão de que era necessário rapidamente desenvolver uma linguagem de programação independente de hardware que fosse adequada para problemas da área empresarial. Para aplicações científicas já havia FORTRAN, e ALGOL estava também sendo definida. Porém, a área comercial e empresarial precisava de sua linguagem. Assim nasceu a ideia para o desenvolvimento de COBOL.

Eles chegaram à conclusão de que necessitariam de um forte patrocínio para esse projeto e que este não poderia vir dos fabricantes de computadores, embora eles fossem convidados a participar da especificação da nova linguagem. Para obter um financiamento neutro, procuraram o Departamento de Defesa dos Estados Unidos, o DoD, que por sua vez ficou bastante interessado no projeto.

Em maio de 1959 foi realizada a primeira reunião especificamente convocada pelo DoD para a especificação do projeto da nova

linguagem. Essa reunião aconteceu no Pentágono, contando com aproximadamente 40 pessoas.

Inicialmente a nova linguagem foi chamada de CBL (Common Business Language, ou Linguagem Comum de Negócios). A maioria dos participantes do evento concordou que deveria ser uma linguagem a mais próxima possível da língua inglesa, abrindo mão dos formalismos matemáticos. O objetivo era que fosse compreensível por mais pessoas, além de matemáticos e engenheiros.

Porém, bem no início, o grupo sequer tinha certeza de que precisariam de fato criar uma nova linguagem. A ideia era, em curto prazo, analisar três outras linguagens que seriam candidatas a se tornar a CBL e talvez propor uma combinação destas. Tratava-se da FLOW-MATIC, desenvolvida por Grace Hopper da Sperry Rand, que, como vimos, era uma sucessora do A-0; a COMTRAN, ou Commercial Translator, definida pela IBM em 1959, mas nunca implementada; e a AIMACO, ou Air Material Command, desenvolvida pela Força Aérea americana também em 1959. Outras linguagens da época acabaram tendo uma menor influência, como Autocoder III, SURGE, FORTRAN, RCA 501 Assembler, Report Generator (GE Hanford) e APG-1 (Dupont). Os membros do comitê sabiam do desenvolvimento de ALGOL, mas consta que ela não teve qualquer influência no trabalho.

As empresas RCA e Sylvania, que participavam do grupo, também estavam desenvolvendo suas linguagens nessa época. Assim, um dos desafios iniciais do grupo era estancar esses projetos, de forma que todos se concentrasssem na linguagem comum.

Foi então estabelecido um comitê executivo, o CODASYL (Committee on Data Systems Languages), o qual supervisionaria o trabalho de outros três comitês: o de curto prazo, o de médio prazo e o de longo prazo. O comitê de longo prazo acabou nunca sendo efetivamente criado e o de curto prazo fez um trabalho em tempo recorde, acreditando que este seria apenas uma obra provisória a

ser totalmente revisada pelo comitê de médio prazo. Isso não aconteceu e, assim, o resultado do trabalho do comitê de curto prazo acabou se consolidando.

O grupo incluía representantes de várias empresas que eram competidoras entre si. Comenta-se que as questões políticas sempre afloravam nas reuniões. Cada empresa concordava em implementar a linguagem, porém sempre havia um “mas” que tornava difícil obter consenso. Além disso, foram criados dois comitês técnicos, um para projetar a linguagem de definição de dados e outro para projetar a linguagem de comandos. Essas duas divisões na linguagem, como se fossem programas separados, são uma das características específicas de COBOL que não é compartilhada por quase nenhuma outra linguagem de programação: na maioria delas, dados e comandos são definidos conjuntamente, mas em COBOL são documentos completamente separados.

Em agosto de 1959 os comitês decidiram que os programas escritos nessa linguagem teriam três grandes divisões: dados, procedimentos e ambiente. A primeira definiria as estruturas dos dados, a segunda definiria as operações a serem realizadas e a terceira, a especificação do hardware no qual o programa seria compilado e executado.

Numa reunião em setembro vários nomes foram propostos para a nova linguagem. Entre eles:

- BUSY: Business System.
- BUSYL: Business System Language.
- INFOSYL: Information System Language.
- DATASYL: Data System Language.
- COSY: Common Systems Language.
- COCOSYL: Common Computer Systems Language.

Felizmente, o nome escolhido acabou sendo COBOL, sigla para Common Business Oriented Language.

Em dezembro de 1959, após muitos dias e noites de trabalho intenso dos comitês, o Relatório COBOL ficou pronto. Foi publicado no início do ano seguinte ([Figura 7.2](#)), sendo o marco para o início da implementação da linguagem. Em 1960 os principais fabricantes de computadores, então, entraram na corrida para definir quem desenvolveria primeiro o compilador COBOL. Por indefinições internas, a IBM ficou de fora dessa corrida por algum tempo. Enquanto isso, a RCA e a Remington-Rand-Univac terminaram seus compiladores. Em dezembro de 1960 ocorreu um marco histórico: dois computadores de fabricantes distintos rodaram o mesmo programa em suas máquinas. A única diferença entre os programas teria sido a divisão de ambiente, que especificava a máquina onde o programa iria funcionar e que, portanto, precisava ser dessa forma. A era da portabilidade de sistemas se iniciava.

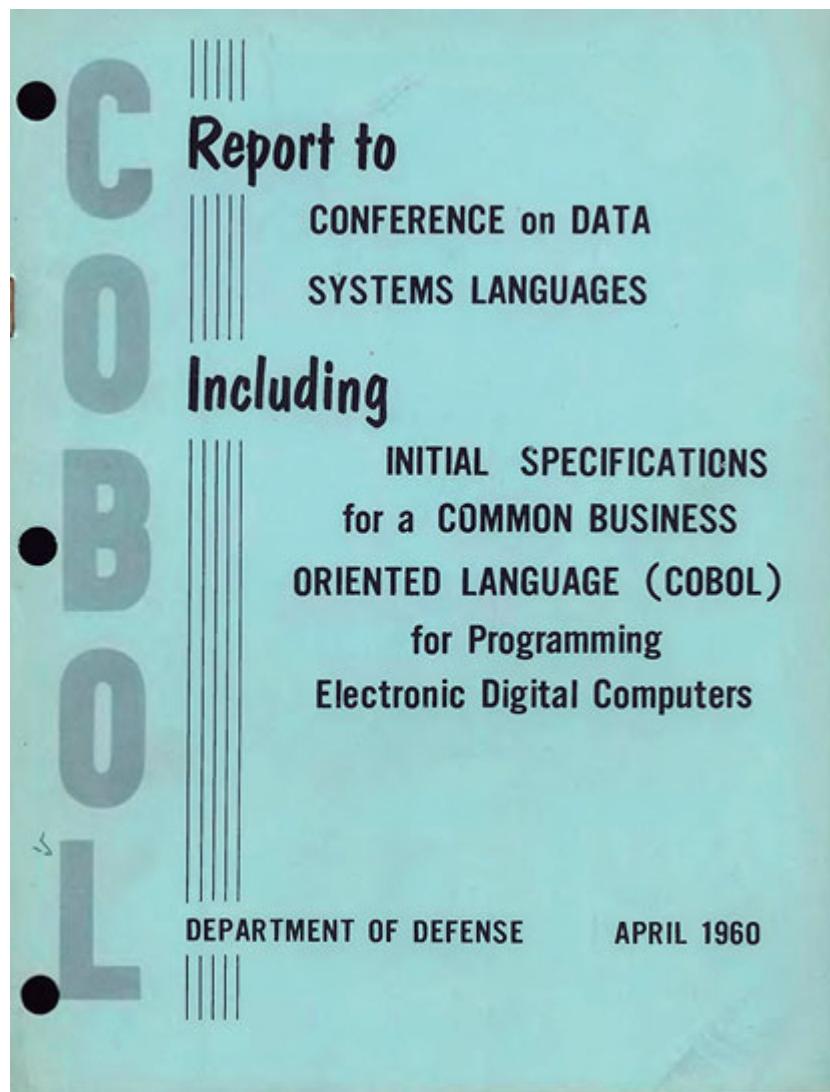


Figura 7.2: Capa do Relatório COBOL-60.<sup>2</sup>

Pode-se dizer que o trabalho de Hopper com FLOW-MATIC teve grande influência em COBOL. Jean E. Sammet (Estados Unidos, 1928), outra mulher que participou do comitê de criação de COBOL, relembra as principais influências do FLOW-MATIC na nova linguagem:

- FLOW-MATIC funcionava! Ela já estava em uso em várias empresas e organizações militares.
- Ela permitia usar nomes longos. Mesmo ainda estando limitada a 12 caracteres, que era o tamanho de uma palavra no UNIVAC,

era melhor escrever SOCIAL-SECUR do que SOSEC, como aconteceria em FORTRAN.

- Ela usava palavras em inglês em vez de símbolos matemáticos. Ler expressões com palavras como ADD ou COMPARE era bem mais fácil do que ler símbolos matemáticos para quem não estava acostumado com eles.
- FORTRAN considerava que todas as palavras de dados em memória possuíam o mesmo tamanho. Isso podia funcionar com aplicações científicas, mas não era adequado para aplicações comerciais. Às vezes, precisava-se de um único caractere para indicar M ou F, representando o gênero masculino ou feminino. Em outros casos, era necessário definir dados longos, maiores do que a palavra que o computador usava. Assim, tanto FLOW-MATIC quanto COBOL permitiam a definição de dados de tamanhos variados, independentemente da máquina que fosse processar estes dados.

Em alguns casos, ela confessa, as decisões sobre nomes de comandos foram tomadas especificamente para serem diferentes do FORTRAN da IBM. Por exemplo, em vez de usar a palavra “DO” para indicar um comando de repetição, eles preferiram “PERFORM”, simplesmente porque FORTRAN utilizava “DO”.

COBOL permitia que qualquer instrução fosse finalizada com a sequência “AND GO TO”, seguido de um endereço, o que possibilitava que o fluxo de controle do programa fosse redirecionado para qualquer outro ponto rotulado pelo endereço. Com o passar do tempo, GOTO tornou-se um comando da maioria das linguagens de programação. Porém, com a adoção das técnicas de programação estruturada, a partir da década de 1970, cada vez mais o uso do GOTO virou uma heresia em programação e símbolo do chamado “código macarrônico”, pois são tantos desvios para lá e para cá no código que a organização dele mais se parece com um prato de macarrão.

COBOL exigia que cada comando iniciasse com um verbo. Isso obrigava, por exemplo, a escrever “COMPUTE” antes de uma fórmula que deveria ser calculada. Havia exceções, entretanto, porque “IF” não é um verbo, mas pode iniciar um comando.

COBOL também permitia que o programador usasse o verbo **DEFINE** para definir o significado de um novo verbo como sequência de comandos. Assim, essa foi possivelmente a primeira vez na história que uma linguagem de programação implementava o conceito de “macro”, ou seja, a possibilidade de dar nome a um fragmento de código, sendo que esse nome, cada vez que fosse usado no código, seria substituído pelos comandos usados para defini-lo.

A linguagem não tinha funções pois, para o comitê, isso só seria do interesse de aplicações científicas, visto que na época o conceito de função era associado apenas com seno, cosseno etc.

Outra característica única que visava a legibilidade em COBOL era a possibilidade de escrever palavras que não tinham nenhum significado para o compilador, mas que tornavam os comandos mais legíveis. Por exemplo, o programador poderia escrever, usando apenas as palavras essenciais:

```
READ file-name INTO data-name; END any-imperative-statement.
```

Ou poderia escrever de forma mais legível usando *noise words*, isso é, palavras que não eram traduzidas em comandos:

```
READ file-name RECORD INTO data-name; AT END any-imperative-statement.
```

Outra característica interessante que não se vê na maioria das linguagens é a possibilidade de escrever expressões contextualizadas. Se você fizer dois testes ou mais, na maioria das linguagens vai escrever algo como:

```
IF X IS EQUALS TO Y comando1;
```

```
IF X IS GREATER THAN Y comando2;
```

IF X IS LESS THAN Y comando3.

Em COBOL você poderia escrever:

IF X IS EQUALS TO Y comando1;

IF GREATER comando2;

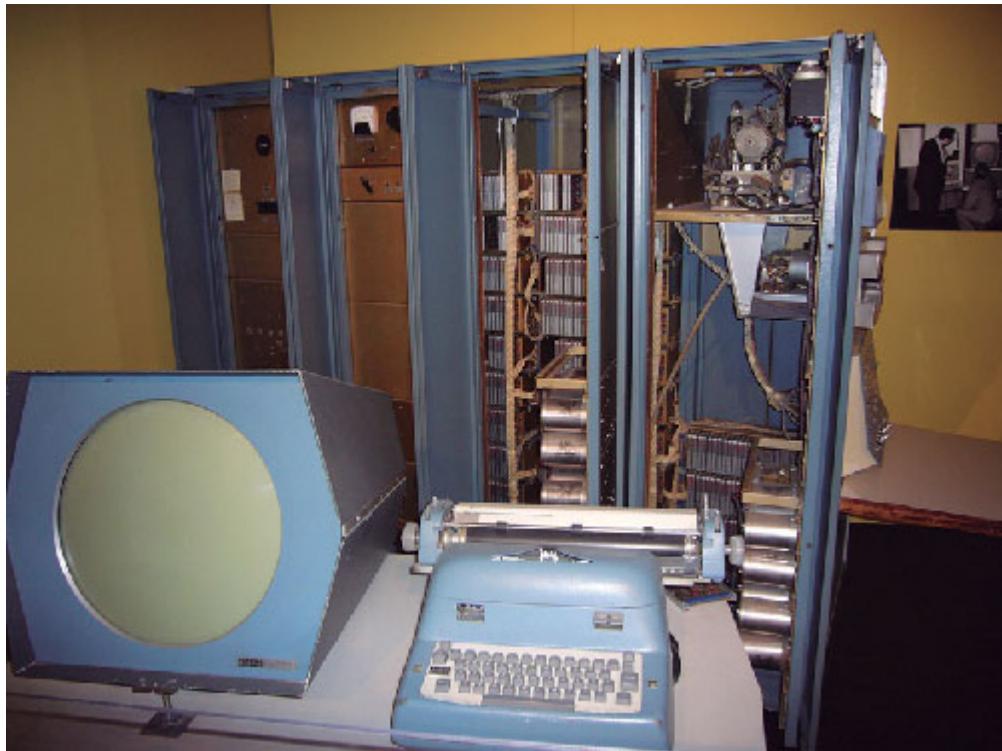
OTHERWISE comando3.

Hoje, COBOL é considerada uma linguagem antiquada pela maioria dos programadores, sendo difícil achar jovens programadores que se habilitem a estudá-la. Apesar disso, foi estimado na virada do século XX para XXI que haveria no mundo cerca de 200 bilhões de linhas de programas escritas em COBOL e que elas seriam responsáveis por cerca de 80% dos sistemas comerciais no mundo. Isso, aliado ao fato de que muitos programadores nos anos 1960 e 1970 optavam por representar o ano nas datas com apenas dois caracteres é que levou ao famoso “bug do milênio”, como veremos. Como o ano 1999 estava representado na maioria dos sistemas simplesmente como “99”, quando fosse meia-noite de 31 de dezembro desse ano, muitos sistemas passariam a acreditar que estavam no ano “00”, mas isso não significava 2000 para os computadores e, sim, 1900. Desta forma, no final do século XX aconteceu no mundo uma intensa busca por programadores COBOL para corrigir os antigos sistemas ainda em uso e prepará-los para usar quatro dígitos em vez de dois.

## 7.3 DEC PDP-1, o Primeiro Minicomputador – 1960

Vimos que o Laboratório Lincoln do MIT desenvolveu o TX-0, um dos primeiros computadores totalmente transistorizados. Com a evolução do projeto para o TX-1 e depois o TX-2, acabaram acontecendo algumas dificuldades que fizeram com que vários membros da equipe saíssem do MIT para fundar suas próprias empresas, entre eles Ken Olsen (Estados Unidos, 1926-2011) e Harlan Anderson (Estados Unidos, 1929).

Foi assim fundada a Digital Equipment Corporation (DEC) que passou a produzir um computador cujo design se baseou em uma versão mais “limpa” do TX-2. Esse novo computador começou a ser entregue aos clientes em 1960. A máquina chamava-se Programmed Data Processor-1 ou PDP-1 ([Figura 7.3](#)). O PDP-1 é também considerado o primeiro minicomputador a entrar em operação comercialmente.



**Figura 7.3: PDP-1.<sup>3</sup>**

A DEC procurou valorizar em seus computadores um fenômeno observado no MIT: eles perceberam que, apesar de haver uma máquina IBM muito potente disponível, os estudantes faziam fila para usar o TX-0, mesmo quando já estava praticamente depenado a esta altura. Ocorre que o TX-0 era um computador que permitia interatividade com o usuário, enquanto que as máquinas IBM eram meros processadores de dados.

Em 1962 a DEC doou uma unidade PDP-1 para o MIT, a qual foi colocada na mesma sala de seu ancestral TX-0. O PDP-1 logo se tornou a máquina preferida da crescente comunidade *hacker* do MIT. Esse computador foi o palco de alguns momentos importantes da história da computação, como a criação do jogo *Spacewar!*, do primeiro editor de texto, do primeiro *debugger* interativo, de um jogo de xadrez completo e de vários sistemas relacionados à música computacional.

O PDP-1 era vendido a 120 mil dólares (equivalente a aproximadamente 950 mil dólares em 2015). Cinquenta e três máquinas foram produzidas até 1969. Três delas ainda existem e estão expostas no Museu de História da Computação de Mountainview, Califórnia. Uma ainda é totalmente operacional e é ligada duas vezes por mês para que os visitantes possam vê-la funcionar.

O PDP-1 tinha um monitor CRT com resolução de 1024x1024 pontos. Ele tinha comandos para desenhar pontos, retas e mesmo curvas, o que fez com que se tornasse uma máquina fundamental para o desenvolvimento inicial da computação gráfica.

Havia também uma espécie de caneta que podia ser colocada na tela e que era capaz de identificar qual ponto da matriz de 1024x1024 estava tocando. Assim, esse instrumento podia ser usado para desenhar na tela e obter representações digitais desses desenhos.

Ele foi também bastante usado para a produção de música computacional. Com alguns periféricos especialmente projetados, era capaz de tocar canções com harmonia de quatro vozes, contando inclusive com um sofisticado programa de codificação de música, o Harmony Compiler, que foi usado para executar muitas composições de Bach e de Mozart.

## 7.4 IFIP – International Federation for Information Processing – 1960

Em junho de 1959, a Organização das Nações Unidas para a Educação, a Ciência e a Cultura (UNESCO) organizou, em Paris, a Primeira Conferência Internacional em Processamento de Informação.

No ano seguinte a UNESCO constituiu a Federação Internacional de Sociedades Científicas da Área de Computação. Originalmente, ela foi batizada IFIPS – International Federal for Information Processing Societies –, mas em 1961 o termo “Societies” foi eliminado e a organização passou a se chamar simplesmente IFIP. Assim, a conferência de Junho de 1959 – hoje rebatizada como IFIP World Computer Congress – ocorre a cada 4 ou 5 anos e é considerada como o primeiro congresso da IFIP.

A IFIP funciona como uma federação de sociedades científicas. O Brasil já foi representado pela SUCESU (Associação dos Usuários de Informática e Telecomunicações), mas atualmente seu representante é a SBC (Sociedade Brasileira de Computação). A ACM é membro da IFIP também desde seu início.

A sociedade organiza seus trabalhos em torno de 13 comitês técnicos, cada qual dedicado a um tópico diferente, como por exemplo, educação (TC3), sistemas de informação (TC8), inteligência artificial (TC12) etc. Cada comitê técnico tem um representante de cada uma das sociedades científicas e um representante de seus grupos de trabalho (WG), usualmente o coordenador ou subcoordenador do grupo.

Já os WG se organizam em pequenos grupos de profissionais interessados em diferentes subtópicos do TC. Por exemplo, o TC12 (inteligência artificial) tem nove grupos de trabalho, como, por exemplo, agentes inteligentes (WG12.3), bioinformática inteligente e sistemas biomédicos (WG12.8).

As regras para pertencer a um grupo de trabalho variam, mas usualmente exige-se que o interessado já tenha participado de pelo menos dois eventos científicos (simpósios, congressos ou workshops) organizados pelo WG, quando então ele pode participar

de uma reunião do grupo de trabalho e pode ser indicado como membro correspondente (sem direito a voto). Quando o membro correspondente participar de sua segunda reunião do grupo de trabalho ele pode ser indicado como membro permanente.

A IFIP tem atuado ativamente no desenvolvimento da computação desde seu início. Como maiores destaques pode-se mencionar o estabelecimento da linguagem ALGOL que, embora não seja mais muito utilizada atualmente, produziu as bases teóricas para a grande maioria das linguagens imperativas de hoje, e o framework de currículos de referência na área de informática, um documento que procura orientar universidades, especialmente aquelas em regiões com maior dificuldade de captar recursos, na elaboração de currículos de informática objetivos e factíveis. Além disso, a federação organiza regularmente eventos e publicações que são acessíveis gratuitamente através de sua biblioteca digital.

## 7.5 Compartilhamento de Tempo – 1961

Até 1960 os computadores eram usados por um usuário de cada vez, mas como ainda eram muito caros, qualquer minuto ou mesmo segundo que ficassem inativos seria um desperdício de recursos. A invenção dos sistemas operacionais diminuiu significativamente o tempo de ociosidade entre um usuário e outro. Porém, enquanto um usuário estivesse usando a máquina poderia haver momentos de ócio até que fosse liberada para o usuário seguinte.

A forma de aproveitar ao máximo a máquina era permitir que vários usuários a usassem ao mesmo tempo. Dessa maneira, com vários programas rodando em simultâneo, quando um deles ficasse ocioso ou deixasse de precisar do processador momentaneamente, o tempo poderia ser ocupado por outro programa. Na prática, os sistemas operacionais tiveram que ser adaptados para permitir que vários programas fossem acionados ao mesmo tempo. Assim, o tempo de processamento passou a ser dividido em “fatias”. Cada programa poderia usar o processador apenas durante a sua fatia.

Depois o computador passaria a executar a fatia do programa seguinte e, assim por diante, até que o primeiro tivesse novamente direito à sua fatia.

A primeira publicação deste conceito ocorreu em 1957 em um artigo de Bob Bemer (Estados Unidos, 1920-2004). Porém, o primeiro projeto a colocar essa ideia em prática foi iniciado por John McCarthy (o criador do LISP) no final de 1957. Assim, embora McCarthy tenha saído do projeto depois, o Compatible Time-Sharing System (CTSS), o primeiro sistema operacional com compartilhamento de tempo foi demonstrado em 1961 no MIT.

Ele teve a ideia quando foi trabalhar no MIT em 1957 e entrou em contato com um IBM 704. Porém, para colocá-la em prática, precisaria criar um sistema de interrupção para essa máquina. Uma “interrupção” era a possibilidade de fazer o computador parar o que estava processando para fazer outra coisa e depois voltar para o programa anterior.

Porém, antes que ele instalasse o sistema que havia concebido soube que a IBM tinha desenvolvido um “pacote para tempo real” a pedido da Boeing, que usava um 704 para captar informações produzidas por um túnel de vento. O pacote foi requisitado e, embora tenha demorado mais de um ano, foi finalmente entregue. A ideia era operar o 704 a partir de três terminais simultaneamente.

Além disso, a IBM concordou em fazer modificações no novo computador 7090 que seria instalado no MIT. Elas incluiriam mais memória para o sistema de compartilhamento de tempo e proteção de memória, que era uma forma de evitar que um programa interferisse com outro quando estivessem rodando ao mesmo tempo.

O PDP-1, sobre o qual já falamos, foi o primeiro computador a disponibilizar compartilhamento de tempo como uma característica de fábrica.

McCarthy defendeu com afinco a ideia desse tipo de sistema. Ele propunha que tempo de computador deveria ser tratado como

serviço público, como água e luz, que você tem à disposição e paga conforme o uso. A ideia foi marcante para a indústria até a década de 1980, quando o surgimento dos microcomputadores diminuiu a necessidade de sistemas de compartilhamento de tempo, porque então cada um poderia ter seu próprio computador a preços relativamente baixos. Na década de 2010, porém, a ideia volta novamente com muita força graças à chamada “computação em nuvem”.

## 7.6 Videogame *Spacewar!* – 1961

*Spacewar!* não foi o videogame que inaugurou essa mídia, como já vimos, mas certamente foi o primeiro a ser amplamente utilizado. Ele foi desenvolvido para o computador interativo, o PDP-1 ([Figura 7.4](#)). O jogo era para dois jogadores, cada qual com uma nave que disparava mísseis contra a outra. No centro da tela havia uma estrela, cuja gravidadeatraia as naves, mas não os mísseis, por falta de poder de processamento.



**Figura 7.4: Spacewar! no PDP-1.<sup>4</sup>**

Em uma situação de emergência, um dos jogadores poderia ir para o hiperespaço. Isso fazia sua nave desaparecer e reaparecer em um lugar aleatório na tela. Porém, cada vez que o recurso do hiperespaço era usado a probabilidade de a nave explodir aumentava.

Um dos principais autores do *Spacewar!* foi Steve Russell (Estados Unidos, 1937), o mesmo que implementou o LISP para o IBM 704. O jogo foi implementado em 1961 no PDP-1 do MIT.

Russell conta que os estudantes do MIT estavam motivados a fazer programas interessantes que pudessem demonstrar as capacidades do novo computador PDP-1, que substituía o antigo TX-2. Com a nova tela de CRT, pensaram em fazer algo que fosse visualmente manobrável. A escolha óbvia, segundo ele, seriam naves espaciais.

Russell foi apelidado de “lesma” pelos colegas, devido à sua demora em executar as tarefas de programação. Em relação ao *Spacewar!*, conta-se que ele procrastinava muito, alegando, dentre outras coisas, que não tinha as rotinas para calcular senos e cossenos necessárias para implementar o jogo. Seu colega Alan Kotok (Estados Unidos, 1941-2006) entrou em contato com a DEC, que lhe informou que possuía as rotinas. Kotok dirigiu até lá, pegou uma fita onde estavam gravadas, jogou-a na frente de Russell e perguntou se ainda tinha uma desculpa.

Russell, então, codificou o jogo, iniciando em dezembro de 1961 e finalizando em 1962. *Spacewar!* ainda pode ser jogado no PDP-1 em operação no Museu de História da Computação na Califórnia.

## **7.7 Zezinho e Lourinha – 1961**

A partir de 1958, o Brasil inicia a construção de computadores. Inicialmente, um projeto secreto do exército brasileiro, realizado no IME, produziu em 1960 um computador digital chamado “Lourinha” e outro analógico. A construção do digital ficou a cargo de José A. M. de Mendonça e Jorge M. Barreto. Após a demonstração, a máquina

foi desmontada para que suas placas pudessem ser usadas em estudos.

Posteriormente, em 1961, o Instituto Tecnológico da Aeronáutica (ITA) passou a construir um computador, o Zezinho, que foi o primeiro a ter divulgação pública.

Tercio Pacitti (Brasil, 1928-2014) introduziu o primeiro computador no Brasil, um IBM 1620. Pacitti ficou muito conhecido também por ter sido autor do livro *FORTRAN Monitor*, que vendeu cerca de 250 mil cópias. Houve muita burocracia para trazer o computador, doado pela Fundação Ford, para o Brasil. No final, ele entrou sob a rubrica de “material para segurança nacional” e foi usado unicamente para fins didáticos.

Esse fato elevou o moral dos professores americanos do instituto. Porém, os professores europeus instigavam o diretor para que criasse o projeto de um computador brasileiro, “mais potente do que o norte-americano”.

Iniciou então o projeto do Zezinho ([Figura 7.5](#)), realizado por quatro alunos de graduação da turma de 1961: Alfred Volkmer, András György Vásárhelyi, Fernando Vieira de Souza e José Ellis Ripper Filho. Eles foram orientados pelo professor Tien Wei Chu.

A máquina foi construída também com finalidade didática, não havendo interesse na época em iniciar uma indústria nacional de computação, o que só veio a acontecer mais de uma década depois.

O Zezinho tinha uma memória de apenas 100 bits, e uma velocidade muito baixa. Em 1963, ele foi aperfeiçoado pelo formando Valdemar W. Setzer (Brasil, 1940), que projetou uma nova fonte de alimentação e concedeu-lhe, então, sua versão final.

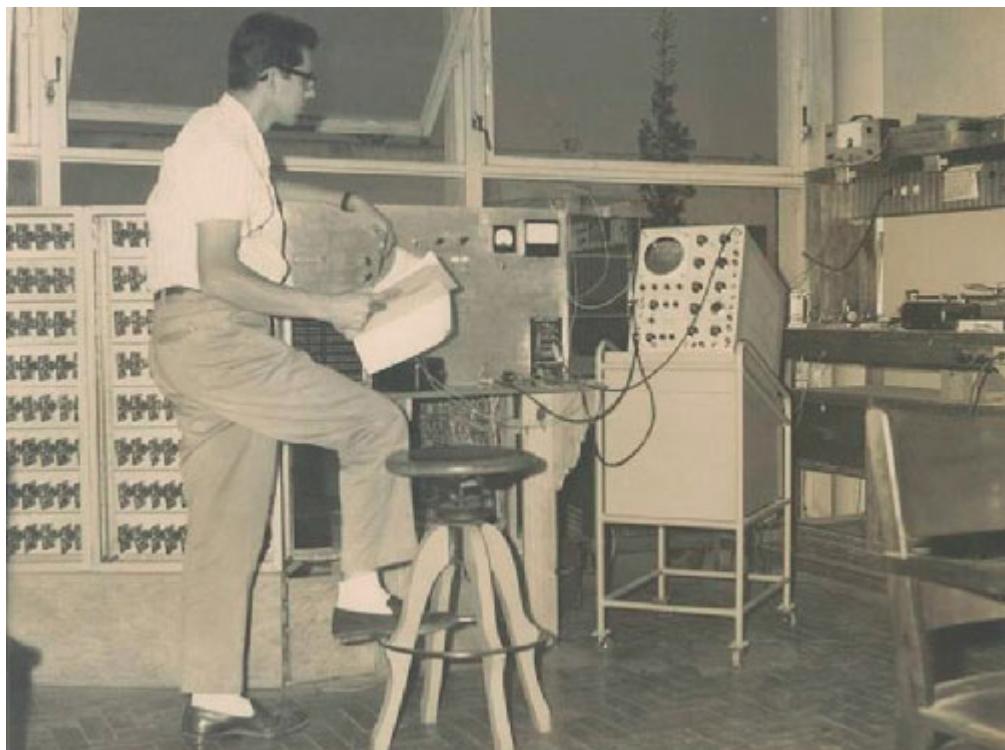


Figura 7.5: Valdemar W. Setzer trabalhando no Zezinho.<sup>5</sup>

## 7.8 ReserVec, Sistema de Reserva de Passagens Aéreas – 1962

A década de 1950 viu o *boom* da aviação comercial. Porém, naqueles tempos, conseguir fazer uma reserva de passagem aérea era um verdadeiro sacrifício, que chegava a deixar uma pessoa pendurada no telefone por até 90 minutos. Cada empresa tinha uma central de reservas que registrava os voos e seus passageiros em fichas de papel que ficavam adequadamente estocadas em gavetas. O problema é que ter muitos funcionários procurando fichas em gavetas acabava causado dificuldades de acesso aos armários, e um tempo demasiadamente longo de espera. Além disso, quando o voo estava cheio, o funcionário precisava voltar ao telefone, pedir outra opção de data ou voo e retornar às gavetas.

Já em 1946 a American Airlines criou um dispositivo eletromecânico para minimizar esse tempo, o Reservisor: uma máquina que permitia ao funcionário selecionar em uma matriz o número do voo e o dia (inicialmente, permitia fazer reservas para os próximos 10 dias) e marcar que um passageiro estava solicitando reserva neste voo. A partir desse registro, os funcionários conseguiam saber o status de um voo sem sair de sua mesa.

Com isso, a empresa conseguiu aumentar bastante o número de atendimentos. Infelizmente, o sistema não funcionava muito bem. O maior problema eram os relês, que ficavam sujos, evitando o contato correto, e precisavam ser limpos periodicamente. Muitos contratemplos aconteceram neste período.

Porém, não foi a American Airlines a primeira empresa a lançar um verdadeiro sistema de reserva de passagens aéreas. Essa honra coube à TCA (Trans-Canada Airlines), hoje Air Canada, que começou a usar em 1962 o ReserVec, ou Reservations Electronically Controlled. A TCA conhecia o Reservisor, mas como ele falhava muito, não acharam o projeto atraente. Um de seus engenheiros passou a estudar o problema em 1952 e chegou à conclusão de que a única solução seria usar computadores.

A TCA procurou ajuda da Universidade de Toronto, que possuía um computador Manchester Mark I chamado FERUT. Um sistema de reservas foi implementado no FERUT e avaliado em 1953. Funcionava bem, mas a entrada e saída de dados era ainda um gargalo. Outro problema era o fato de o FERUT ser construído com válvulas que queimavam de vez em quando, limitando assim sua confiabilidade, o que fazia retornar ao problema inicial.

A TCA procurou então ajuda à recém-formada empresa Ferranti-Canadá. A solução proposta pela Ferranti foi o uso de cartões de papel que podiam ser marcados a lápis pelos operadores. Depois seriam inseridos em uma máquina que lia as marcas e perfurava no mesmo cartão as posições equivalentes para que ele pudesse ser lido por uma leitora comum. Os dados podiam então ser

transmitidos por linha telefônica a um concentrador que faria o processamento de dados vindos de vários operadores ou mesmo de vários escritórios espalhados pelo país.

Além disso, foi proposta a construção de um computador transistorizado, o que reduziria o problema da confiabilidade baixa causada pela queima de válvulas.

Em 1957, um protótipo do sistema programado no FERUT e ligado a seis terminais, que eles chamavam na época de “transactors”, funcionou perfeitamente.

Em 1959, a TCA executou um projeto de 2 milhões de dólares para construir 350 terminais e um computador chamado Gemini, com duas unidades de processamento: Castor e Pullux. O computador seria construído pela Ferranti-Canada e continha palavras de 25 bits e 4.096 (4 kB) de memória principal, depois expandida para 8 kB. A memória secundária era formada por cinco tambores magnéticos, sendo um de reserva, e cada qual podia conter 32 kB com palavras de 25 bits e seis unidades de fita.

Quando o sistema na TCA foi instalado e colocado em uso percebeu-se que o tempo médio de atendimento de um pedido de reserva caiu de 90 minutos para um segundo! No pior dos casos, quando o sistema estava muito ocupado, levava dois segundos. O sistema ReserVec como um todo conseguia atender 10 requisições por segundo.

A American Airlines (AA), porém, não ficou parada esse tempo todo. Tendo já experiência com o Reservisor, eles também iniciaram o projeto de um sistema de reservas automatizado em 1953. Seu sistema, o SABRE, só entrou em operação em 1964, dois anos depois do ReserVec. Porém, o sistema da AA era mais sofisticado, pois registrava os nomes dos passageiros, enquanto o ReserVec só registrava um código de três letras para cada passageiro.

A Ferranti esperava vender seu sistema para outras empresas aéreas, mas nunca conseguiu. As empresas americanas estavam saturadas com máquinas IBM ou UNIVAC, no Canadá não havia

outras companhias de porte e na Europa a Ferranti britânica, em vez de passar a vender o sistema canadense, anunciou que desenvolveria o seu próprio, coisa que a Ferranti-Canadá tomou como insulto. No final, o sistema britânico nunca foi desenvolvido e a Ferranti-Canadá, agora Ferranti-Packard, acabou vendendo a divisão de computadores para se concentrar em outros produtos.

## 7.9 Supercomputadores Atlas e Titan – 1962

O computador Atlas, quando foi construído em 1962 pela Ferranti, Plessey e Universidade de Manchester, foi considerado um supercomputador (possivelmente o primeiro da história). De fato, ele era de longe o mais poderoso computador de sua época. Dizia-se que quando era desligado, a Inglaterra perdia metade de seu poder computacional.

Apenas três Atlas foram construídos. Uma versão chamada Atlas II ou Titan, com capacidade para compartilhamento de tempo, também teve três unidades construídas. O Atlas foi um projeto de pesquisa realizado pela Universidade de Manchester que visava construir o primeiro processador capaz de processar uma instrução em um milionésimo de segundo, ou um microsegundo.

Ele tinha palavras de 48 bits, 16 kB de memória principal (eletrônica) e 96 kB de memória secundária em tambor magnético. Embora possuísse quatro tambores, um sistema de memória virtual permitia que os programadores usassem essa memória e a memória principal como uma única unidade.

Atlas e Titan suportavam os compiladores ALGOL, FORTRAN e COBOL, e a maioria de suas unidades foram instaladas em universidades britânicas, como Manchester, Londres e Cambridge.

A [Figura 7.6](#) é uma foto do Titan de Cambridge.

O conceito de memória virtual do Atlas, ainda hoje usado em praticamente todos os computadores, foi inventado por Tom Kilburn (Reino Unido, 1921-2001). É possível que ele tenha se baseado na

tese de doutorado de Fritz-Rudolf Güntsch (Alemanha, 1925-2012), de 1956.

Até esse conceito ser inventado, os programadores tinham que dizer explicitamente se estavam acessando dados na memória principal (eletrônica) ou secundária (magnética). Acontece que os computadores só conseguem efetivamente operar os dados que estão na memória principal; um dado que esteja na memória secundária precisa ser carregado para a memória principal e só então pode ser operado. Se você tivesse uma memória principal de, digamos, 16 kB e uma secundária de 96 kB, você só poderia trabalhar com no máximo 16 kB de cada vez, e precisaria ficar escrevendo e lendo dados na memória secundária. Com a criação da memória virtual, seria como se a memória principal tivesse 96 kB. Os programas poderiam ser escritos considerando uma memória de 96 kB, sem se preocupar em ler e escrever na memória secundária. Um sistema operacional usaria uma técnica chamada de “paginação” para gravar na memória secundária os dados menos usados e trazê-los de volta para a memória principal se fossem necessários novamente.



Figura 7.6: Titan de Cambridge.<sup>6</sup>

## 7.10 Mouse – 1962

Dez anos depois da construção da primeira trackball, e possivelmente de forma totalmente independente, Douglas Engelbart (Estados Unidos, 1925-2013) criou o primeiro mecanismo de entrada de dados baseado em uma caixinha que podia ser movida com a mão sobre uma mesa. Como a caixinha tinha um fio que ficava preso na parte de trás e a ligava ao computador, ele batizou o invento como “mouse”, por sua vaga semelhança com o roedor. Em Portugal, o dispositivo é efetivamente chamado de “rato”.

O mouse foi inventado por Engelbart no Stanford Research Institute (SRI International), que ficou com a patente. Porém, ela venceu antes que o invento se tornasse amplamente usado nos computadores pessoais.

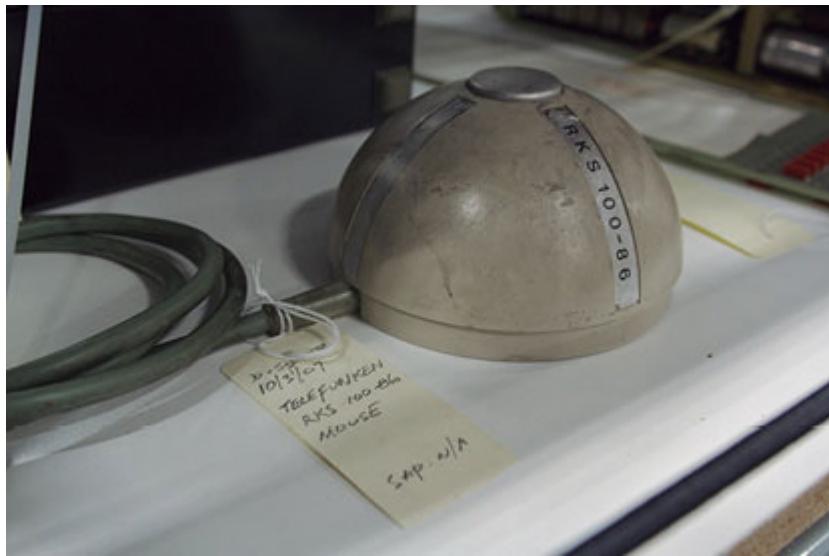
Esse primeiro mouse sequer tinha a *rollerball* em seu interior, mas apenas dois discos ortogonais que giravam em função da fricção com a superfície da mesa. O dispositivo é mostrado na [Figura 7.7](#).



**Figura 7.7: O primeiro mouse.<sup>7</sup>**

Ele não surgiu do nada. O SRI, com apoio da NASA, investigou vários instrumentos que poderiam ser usados para realizar determinadas atividades no computador, como, por exemplo, apontar para um objeto que surgisse numa posição aleatória da tela. Objetos como a caneta de luz não eram muito ergonômicos, pois era necessário pegá-los da mesa e erguê-los até a tela, num movimento de braço amplo, que dificultava seu uso. Com o mouse sobre a mesa isso era muito mais fácil.

Os primeiros mouses com *rollerball* interna foram construídos pela Telefunken a partir de 1968 e eram chamados de “rollkugel”. Esse dispositivo é mostrado na [Figura 7.8](#).



**Figura 7.8: Primeiro mouse com *rollerball* interna.<sup>8</sup>**

Atualmente os mouses utilizam luz para “sentir” o movimento e não possuem mais a *rollerball* em seu interior. Além disso, a criação dos mouses sem fio eliminou uma de suas principais características de roedor: a cauda.

## **7.11 SAGE e os Maiores Computadores de Todos os Tempos – 1963**

O SAGE, ou Semi-Automatic Ground Environment, foi um gigantesco projeto de computação e redes norte-americano, iniciado nos anos 1950 e que ficou operacional a partir de 1961. Ele seria uma rede com os maiores computadores construídos em toda a história da humanidade (em volume, não em capacidade de processamento, visto que hoje o sistema sucessor do SAGE funciona em microcomputadores).

O SAGE seria responsável por coletar, transmitir e organizar dados recebidos de vários radares espalhados pelos Estados Unidos e Canadá. Esses dados seriam usados para direcionar o NORAD (North American Aerospace Defense Command) no caso de ataque aéreo. O sistema foi construído principalmente em função da Guerra

Fria, quando se esperava possíveis ataques atômicos soviéticos vindos pelo Ártico ou a partir de submarinos.

Os computadores gigantescos que formavam a base do SAGE eram modelos IBM AN/F SQ-7<sup>9</sup> Combat Direction Central, mais conhecidos como “Q7”. Vinte e quatro dessas máquinas foram instaladas nos diversos centros de controle do SAGE. Eram redundantes, ou seja, cada máquina era formada por dois computadores independentes que processavam os mesmos programas ao mesmo tempo. Se houvesse alguma falha em um deles, eles produziriam resultados diversos e, assim, os técnicos poderiam verificar rapidamente que os resultados não eram confiáveis, evitando conflitos acidentais.

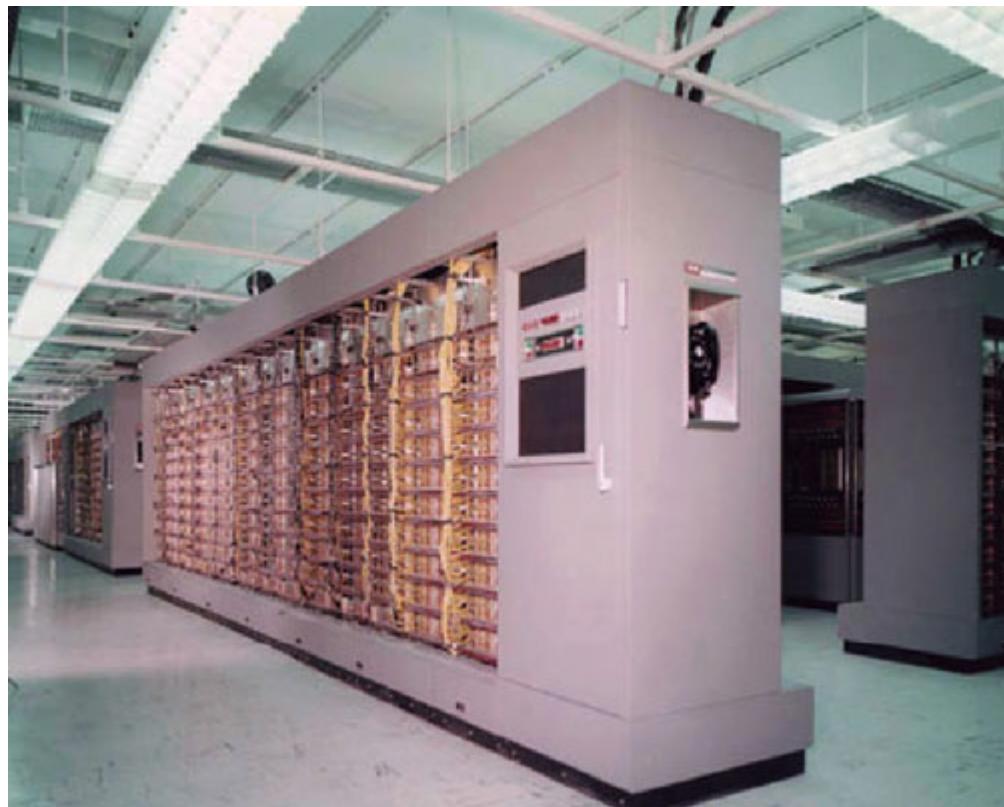
As máquinas foram projetadas com base no IBM 701 e usavam memórias de núcleo magnético que seriam versões melhoradas da memória do Whirlwind I. Cada uma pesava 250 toneladas e era construída com mais de 60 mil válvulas, consumindo cerca de 3 milhões de watts. Como se pode observar na [Figura 7.9](#), o console do computador incluía até uma cabine telefônica. A máquina precisava ser muito rápida, pois deveria receber dados de um número grande de estações de radar e teria que processar estes dados em tempo real.

No caso de uma invasão de espaço aéreo, o software do SAGE era capaz de predizer o melhor ponto de interceptação e automaticamente direcionaria mísseis ou aeronaves para ele.

Os computadores tinham monitores gráficos de CRT e podiam ser operados através de canetas de luz. O software que rodava no SAGE tinha cerca de 500 mil linhas de código, de longe os maiores programas de computador da época.

Um projeto para computadorizar a segurança dos céus norte-americanos já existia desde 1945. Em 1949 foi sugerido por um alto comitê a criação de uma “rede de computadores”. Em 1951, um teste foi feito com a transmissão de dados produzidos por radar através de uma linha telefônica para um computador, possivelmente

a primeira realização deste tipo de transmissão da história. O teste foi conduzido por C. Robert Wieser, do Laboratório Lincoln do MIT, e comprovou a viabilidade do sistema.



**Figura 7.9: IBM Q7.<sup>10</sup>**

O primeiro relatório com os parâmetros iniciais do SAGE foi produzido em 1953 em um projeto conjunto da IBM, MIT e Força Aérea Americana (USAF).

O primeiro protótipo do Q7 foi entregue em 1956 e o primeiro posto de batalha regional do SAGE se tornou operacional em 1958 em Syracuse, Nova York. O sistema completo ficou ativo em 1963. Porém, quando isso aconteceu, os computadores a válvula já eram considerados obsoletos, visto que na época a maioria dos novos modelos eram transistorizados ou mesmo implementados com circuitos integrados. Ainda assim, esse parque computacional ficou

ativo durante praticamente toda a Guerra Fria, sendo desativado apenas ao longo da década de 1980.

## 7.12 Sketchpad, o Início da Computação Gráfica – 1963

Em 1963, Ivan Sutherland (Estados Unidos, 1938) defendeu uma tese de doutorado no MIT na qual, orientado por Claude Shannon e apoiado por Marvin Minsky, ele apresentou o Sketchpad: um sistema gráfico interativo que permitia ao usuário, através de uma caneta de luz, desenhar pontos em uma tela. Mas não só isso: os pontos podiam ser automaticamente ligados por linhas; polígonos podiam ser definidos por desenho livre e seriam corrigidos automaticamente pelo sistema. Ele podia desenhar um círculo a mão livre e o programa corrigiria as imperfeições, deixando o círculo o mais perfeito possível de acordo com a resolução da tela.



**Figura 7.10: Ivan Sutherland operando o Sketchpad.**<sup>11</sup>

Ainda tinha mais: uma vez que as figuras eram desenhadas, podiam ser manipuladas como objetos de desenho, ou seja, você podia “pegar” um polígono com a caneta de luz e “arrastá-lo” de um canto da tela para outro. Isso foi algo totalmente inovador em termos de computação. Esse marco, essa tese, é considerada a fundação da área de computação gráfica, e o Sketchpad, um sistema ainda em produção, é tido como o primeiro programa de computação gráfica com interface gráfica.

O programa teria sido também influenciado pelas ideias de Vannevar Bush, especialmente o Memex. A primeira versão rodava no MIT no computador TX-2, sucessor do TX-0. Um dos fatores que permitiu a Sutherland implementar um sistema interativo e gráfico foi justamente o fato de que o TX-2 implementava o conceito de computação interativa, o mesmo explorado depois pela DEC no seu PDP-1. Com computadores no estilo “batch”, ou seja, computadores que recebiam dados, processavam e depois produziam uma saída, um programa como o Sketchpad seria apenas algo como um conjunto de instruções e dados que o usuário poderia inserir na máquina e, depois, caso o resultado não fosse o que ele queria, precisaria corrigir número por número, instrução por instrução, e tentar processar novamente. No entanto, a era da computação interativa estava começando e, assim, os usuários de agora em diante poderiam comunicar-se diretamente com seus programas à vontade, movendo objetos na tela e fazendo outras coisas que a máquina interpretaria e transformaria em dados.

O conceito moderno de programação orientada a objetos também é atribuído a Sutherland. De fato, o termo só passou a ser usado nos anos 1970 após a criação de Smalltalk. Depois, descobriu-se que a linguagem de programação Simula 67 já implementava estes conceitos em 1967. Porém, o Sketchpad em 1963 já trabalhava com a concepção de classe, instância e herança, que são fundamentais para a programação orientada a objetos, sendo, portanto, o pioneiro.

O trabalho de Sutherland com Alan Kay, seu aluno, levou a influência de suas ideias (tornar o computador acessível ao usuário leigo) para o projeto Dynabook, dentro do qual muitos conceitos da computação moderna foram criados ou consolidados.

O primeiro contato de Sutherland com um computador ocorreu com o kit Simon de 1950, para o qual ele escreveu um programa para dividir dois números. Esse programa com oito páginas de fita de papel possivelmente foi um dos maiores, senão o maior, já executado no Simon. Para fazê-lo funcionar, Sutherland precisou implementar um novo circuito de modo a parar a computação dada uma certa condição.

Em 1988, ele recebeu o Prêmio Turing da ACM pela invenção do Sketchpad. Não tanto pelo seu uso, que foi limitado, mas pelas ideias apresentadas na tese. A interface gráfica do Sketchpad influenciou outras que ainda estavam por nascer, como a da estação de trabalho Xerox Star e a interface gráfica do Apple Macintosh e do Microsoft Windows.

## 7.13 BASIC – 1964

Até meados dos anos 1960, como já vimos, a maioria dos computadores não eram interativos, mas máquinas que recebiam dados, executavam processamento e devolviam uma saída. Não eram aparelhos em frente aos quais você pudesse se sentar e, por exemplo, fazer perguntas e obter respostas, ou pesquisar assuntos baseando-se em palavras-chave. As linguagens de programação também tinham essa característica: você escrevia um programa, ia numa perfuradora de cartões ou fita para obter uma versão que a máquina pudesse ler, compilava o programa, produzindo uma versão em código de máquina, e depois disso estava pronto para o ciclo de entrada de dados, processamento e saída.

O problema é que algumas vezes o tempo que se levava entre você entregar o programa em cartões perfurados no birô de processamento até obter o relatório de saída impresso podia ser de

algumas horas até alguns dias. Se o seu programa tivesse algum erro (bug), ele não ia ser executado, mas você só ficaria sabendo disso no dia seguinte.

Essa situação tornava o processo de aprendizagem de programação bastante desgastante para os estudantes. Pensando nisso, John Kemeny (Hungria, 1926-1992) e Thomas E. Kurtz (Estados Unidos, 1928), professores do Dartmouth College, conceberam um sistema de compartilhamento de tempo e uma linguagem de programação simplificada, ambos com propósito didático; ou seja, seria o primeiro ambiente de aprendizagem de programação efetivamente idealizado para este fim.

O sistema de compartilhamento de tempo foi o DTS ou Dartmouth Time-Sharing System, inspirado no sistema do PDP-1. Ele ficou à disposição dos alunos de Dartmouth, funcionando de 1964 até 1999. Originalmente, foi implementado em um computador fabricado pela General Electric, o GE-200. A sugestão de que Kurtz fizesse um sistema desse tipo partiu de John McCarthy, em uma visita daquele ao PDP-1 no MIT.

A linguagem criada por eles foi o BASIC (Beginner's All-purpose Symbolic Instruction Code). Inspirada em FORTRAN e ALGOL, pretendia ser bem mais simples do que as antecessoras. Ela podia ser programada a partir de um terminal do GE-200 executando DTS. Não havia editores de texto como os que conhecemos hoje. Os comandos para o DTS e os programas BASIC eram digitados a partir de um editor de uma única linha. Assim, cada linha de um programa BASIC tinha um número que definia sua posição no programa. Normalmente essas linhas eram numeradas de 10 em 10 para facilitar a eventual inserção de outras linhas no meio do código.

Por exemplo, o programador podia digitar a linha 10 e depois as linhas 20 e 30. Se ele resolvesse inserir uma linha entre as de número 10 e 20, escreveria no editor uma linha iniciada, por exemplo, com 15. Assim, uma listagem do programa mostraria as linhas 10, 15, 20 e 30.

Se o programador quisesse, por outro lado, apagar uma linha do programa, bastaria escrever o número da mesma no editor de linhas. Por exemplo, para apagar a de número 20, escreveria “20” no editor e, ao dar entrada nesta, a linha 20 do programa seria apagada. Se você quisesse alterar uma linha de código, precisaria escrevê-la novamente por completo, iniciando pelo seu número no editor de linhas.

Quando o programador quisesse ver o programa na tela, ele devia executar o comando “LIST” (sem número de linha). Isso fazia com que o programa fosse listado na tela do computador, mas não lhe dava a opção de editá-lo diretamente, o que só aconteceria através do editor de linhas.

O BASIC, desde o início, foi mantido como uma linguagem de domínio público. Devido à sua simplicidade, ele rapidamente se espalhou pelas universidades e empresas. A partir de meados dos anos 1970, com a chegada do microcomputador e a possibilidade de implementar versões do BASIC em máquinas extremamente limitadas, chegou a ser a linguagem de programação mais usada no mundo, só perdendo a liderança com a disseminação das modernas linguagens orientadas a objetos.

O manual do BASIC de 1964 apresenta logo em seu início um exemplo de uso da linguagem. Trata-se de um programa para resolver duas equações lineares simultâneas:

$$A_1X_1 + A_2X_2 = B_1 \quad A_3X_1 + A_4X_2 = B_2$$

Como se trata de apenas duas equações, os valores das incógnitas  $X_1$  e  $X_2$  podem ser calculados por:

$$X_1 = \frac{(B_1A_4 - B_2A_2)}{(A_1A_4 - A_3A_2)}$$

$$X_2 = \frac{(A_1B_2 - A_3B_1)}{(A_1A_4 - A_3A_2)}$$

Porém, por causa da divisão, uma solução única só pode existir se  $(A_1A_4 - A_3A_2)$  for diferente de zero. Assim, o seguinte programa BASIC resolveria essas equações:

```

10 READ A1, A2, A3, A4
15 LET D = A1*A4-A3*A2
20 IF D=0 THEN 65
30 READ B1, B2
37 LET X1=(B1*A4-B2*A2)/D
42 LET X2=(A1*B2-A3*B1)/D
55 PRINT X1, X2
60 GO TO 30
65 PRINT "NO UNIQUE SOLUTION"
70 DATA 1, 2, 4
80 DATA 2, -7, 5
85 DATA 1, 3, 4, -7
90 END

```

A linha 10 lê os valores 1, 2, 4 e 2 nas linhas 70 e 80, que são linhas que contêm os dados. A linha 15 calcula D como a expressão usada no divisor de ambas as fórmulas:  $(A_1A_4 - A_3A_2)$ . A linha 20 verifica se D é zero; se for, pula a execução para a linha 65.

Caso o programa não tenha sido desviado para a linha 65, ele continua na linha 30 que lê os valores -7 e 5 na linha de dados 80. A

linha 37 calcula X1 e a linha 38 calcula X2. A linha 55 imprime os dois valores calculados.

A linha 60 envia o programa de volta à linha 30, onde ele vai ler novos valores para B1 e B2, no caso, os valores 1 e 3 da linha 85. As saídas são novamente calculadas e impressas nas linhas 37, 42 e 55.

Uma nova execução da linha 60 leva novamente e volta à linha 30, onde B1 e B2 serão carregadas com os valores 4 e -7 obtidos no final da linha 85. O cálculo das saídas é novamente feito e pela terceira vez os resultados são impressos.

A última execução da linha 60 leva o programa à linha 30, onde ele vai parar porque não existem mais dados para serem lidos pelo comando READ.

A existência do comando GO TO (implícito na linha 20) e a falta de estruturas de programação como, por exemplo, recursão fizeram com que o BASIC recebesse muitas críticas, especialmente após o advento da programação estruturada, nos anos 1970.

O BASIC, porém, não foi a primeira linguagem criada pelo grupo de Dartmouth. Eles já haviam proposto a DARSIMCO (Dartmouth Simplified Code) em 1956, um conjunto de padrões de programação em Assembly que acabou não sendo muito usado pois FORTRAN foi lançada logo no ano seguinte.

Outra experiência ocorreu em 1962, com a linguagem DOPE (Dartmouth Oversimplified Programming Experiment), já sofrendo maior influência de FORTRAN e ALGOL, mas ainda uma linguagem muito simples, pois, por exemplo, só permitia uma única operação aritmética por linha de código.

O BASIC criado em Dartmouth era compilado, porém tornou-se bastante popular e várias versões interpretadas surgiram. Compilar implica em o computador traduzir o programa em BASIC para um programa na linguagem da máquina, e depois executar este segundo programa. Interpretar implica em o computador ler cada instrução em alto nível e imediatamente executar as instruções de

máquina correspondentes. O processo de interpretação é usualmente bem mais lento do que a execução de programas compilados, porque durante a execução existe toda a sobrecarga de ler e interpretar os comandos em alto nível antes de executar o código propriamente dito.

Ganhava-se em interatividade, entretanto. Por exemplo, sistemas BASIC interpretados podiam fazer a checagem sintática dos comandos no momento em que cada linha era digitada. Programas compilados só fariam uma checagem geral após o programa todo estar finalizado. Além disso, os compiladores daquela época eram bem diferentes dos atuais. Kurtz conta que havia uma história sobre um compilador FORTRAN que levava 45 segundos para compilar um programa com uma única instrução “END”. Isso ocorria porque para compilar essa instrução ele tinha que buscá-la em uma tabela de comandos armazenada em uma fita e ela era seu último elemento.

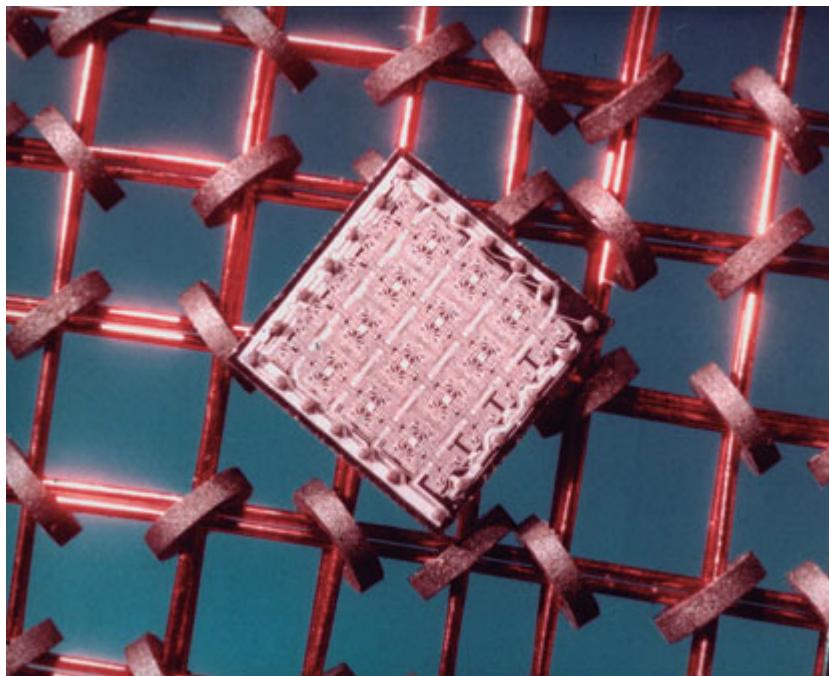
Várias versões do BASIC foram criadas ao longo da história. Só em Dartmouth, até 1971 seis gerações da linguagem passaram. A proliferação de dialetos fez com que a ANSI (American National Standard Institute) iniciasse em 1974 um projeto para uma especificação mínima padrão de BASIC, o que ocorreu em 1978, e uma especificação padrão surgiu em 1987.

Muitos microcomputadores dos anos 1970 e 1980 vinham de fábrica com uma cópia do BASIC, às vezes até gravada em hardware, como no caso dos Sinclair e assemelhados. Atualmente a versão mais utilizada é o Visual Basic. Segundo a tabela TIOBE<sup>12</sup> em fevereiro de 2016 o Visual Basic .NET era a 7ª linguagem de programação mais popular no mundo, ao passo que o Visual Basic (não .NET) aparece em 12º.

## 7.14 IBM 360 – 1964

Pode-se dizer que o início da era da computação moderna ocorreu em 1963 quando John Watson Jr. (Estados Unidos, 1914-1993)

anunciou o modelo IBM 360. O 360 é considerado o marco da terceira geração de computadores (com circuitos integrados), inovando em termos de processador e memória, utilizando circuitos integrados capazes de operar instruções em bilionésimos de segundo. Os circuitos integrados usados no 360 foram criados pelo Solid Logic Technology da IBM e eram muito mais rápidos e confiáveis do que outros que existiam. Só para exemplificar, a [Figura 7.11](#) mostra em primeiro plano o circuito integrado de uma memória sólida de 64 bits sobre uma memória de núcleo magnético, que era o que de melhor se tinha antes disso.



**Figura 7.11:** Memória sólida do 360 sobre uma memória de núcleo magnético.<sup>13</sup>

Porém, a velocidade talvez não tenha sido a principal característica que tornou essa máquina um marco, pois tratava-se apenas de uma evolução natural. A verdadeira revolução que este modelo iniciou estava no fato de que a partir dele passou a existir toda uma família de computadores totalmente compatíveis e atualizáveis, porque teriam o mesmo conjunto de instruções de máquina. Antes disso,

quando uma empresa comprava um computador relativamente pequeno e depois atualizava para um mais potente, ela precisava reescrever boa parte de seus programas, além de comprar novos periféricos como impressoras, unidades de disco e terminais. Com o 360, era possível que o usuário iniciasse com um computador pequeno e fosse atualizando seus sistemas na medida da necessidade, sem maiores preocupações ou esforços. O próprio número 360 vinha do fato de que a IBM queria abranger um círculo de 360° de necessidades de usuários. O computador dava aos usuários a segurança de saber que os upgrades seriam tranquilos.

A preocupação com portabilidade era tão grande que emuladores foram construídos para sistemas desenvolvidos em computadores mais antigos. Emuladores eram programas que simulavam uma máquina, mas rodavam em outra.

Na época, o projeto do 360 poderia ter sido considerado um suicídio, pois a IBM gastou 5 bilhões de dólares nele e seu sucesso implicaria na obsolescência de muitos outros produtos da empresa.

Seis modelos de processador foram anunciados. O mais rápido era 25 vezes mais veloz que o mais lento. Seis anos depois, novos modelos foram lançados e essa relação passou a ser de 200 para 1.

Além disso, toda uma série de periféricos compatíveis foi construída de forma que uma organização pudesse escolher e utilizar aqueles que fossem necessários na quantidade que precisavam.

Outro aspecto que o 360 consolidou foi o fim da distinção entre o computador científico e o computador para fins comerciais. Ele podia servir a qualquer uma dessas áreas.

Um IBM 360/91 ([Figura 7.12](#)) foi especificamente desenvolvido para a recém-criada Agência Espacial Americana, a NASA. Esse computador foi largamente utilizado no projeto Apollo, que levou o ser humano à Lua.



Figura 7.12: Painel do IBM 360-91 na NASA.<sup>14</sup>

## 7.15 Comutação de Pacotes – 1965

Até a invenção da Comutação de Pacotes (Packet Switching) em 1965, a forma de transmitir informação através de redes de telefonia ou assemelhadas consistia em estabelecer uma conexão entre dois pontos, transmitir toda a informação e desligar a conexão. Dessa forma, a banda de comunicação ficava dedicada desde o início até o final a uma mensagem e bloqueada para outras até que terminasse. Dessa forma, estimava-se que até 90% do tempo de banda de transmissão era desperdiçado.

Porém, no início dos anos 1960, em plena Guerra Fria, Paul Baran (Polônia, 1926-2011), trabalhando para a Rand nos Estados Unidos, tentava imaginar uma forma de transportar informação vital entre computadores de forma a resistir a eventuais ataques atômicos. Baran imaginou uma rede de computadores ligada de várias maneiras diferentes, de forma que a destruição de uma linha de comunicação pudesse ser compensada por outras linhas que

permaneceriam ativas, mesmo que entre computadores diferentes. Além disso, no modelo que ele imaginou, as informações seriam divididas em pequenos blocos, transmitidos um de cada vez através da rede, seguindo possivelmente caminhos diferentes para serem reagrupados somente no receptor.

Seria algo como mover uma casa de uma cidade para outra, transportando suas peças, em caminhões diferentes. Os caminhões podem até seguir caminhos distintos, mas no destino a casa poderá ser novamente montada. No caso da comunicação de informações, isso permitia que diversos computadores se comunicassem com outros, enviando várias mensagens ao mesmo tempo através de múltiplos caminhos. Rotas que ficassem muito cheias poderiam ser preteridas caso as mensagens, ou partes delas, pudessem transitar por alternativas, como em um sistema de estradas: quando a rodovia principal fica congestionada, os carros começam a procurar vias diferentes. A [Figura 7.13](#) mostra esquematicamente dois computadores transmitindo três mensagens diversas através de caminhos distintos em uma rede hipotética.

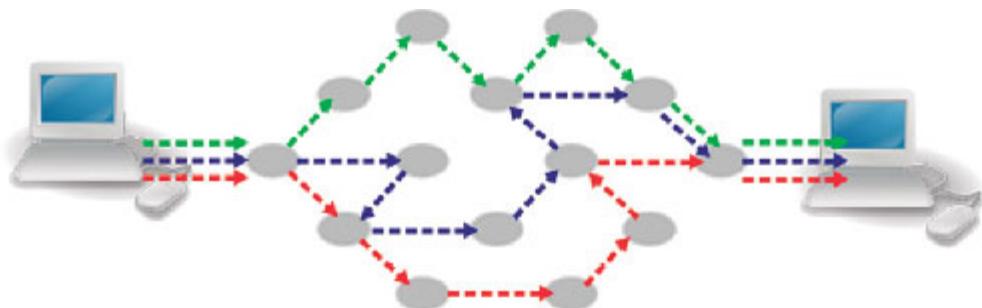


Figura 7.13: Comutação de pacotes.<sup>15</sup>

Em 1965, Baran obteve aprovação da Força Aérea Americana para testar sua teoria de comutação de blocos. Porém, a Defense Communications Agency – DCA (Agência de Comunicação da Defesa), atual Defense Information Systems Agency – DISA tomou o

projeto para si e o complicou tanto que Baran não conseguiu testar a ideia e acabou desistindo.

Nessa mesma época, Donald W. Davies (Reino Unido, 1924-2000) na Inglaterra, concebeu um conceito bem similar de forma totalmente independente, que ele chamou de “comutação de pacotes”. Cada pacote de informação teria o tamanho de 128 bytes. Após distribuir um artigo com suas ideias em 1966, ele ficou sabendo da publicação de Baran no ano anterior.

A primeira publicação de Davies sobre o assunto ocorreu em 1967 em um simpósio da ACM. Havia na época, entre os engenheiros, uma tendência a rejeitar sistemas de comunicação baseados em divisão de mensagens, como a comutação de pacotes, porque essa ideia já era conhecida havia décadas e sempre fora considerada economicamente desvantajosa. Portanto, ninguém pensava seriamente em retomar o assunto.

No entanto, Davies realizou uma rigorosa análise econômica e, considerando que agora se tratava da transmissão de dados entre computadores e não de voz entre seres humanos, mostrou que neste caso a ideia de dividir as mensagens em pacotes era extremamente vantajosa.

Hoje, Davies e Baran são considerados os coinventores desse importante conceito, que viabilizou anos mais tarde as redes de computadores.

## 7.16 CDC 6600 de Seymour Cray – 1965

A CDC, ou Control Data Corporation, foi uma das fabricantes de computadores dos anos 1960 que dividia com outras seis empresas uma pequena fatia de um mercado que era amplamente dominado pela IBM. Por conta disso, a IBM e essas outras sete empresas acabaram sendo conhecidas como a “Branca de Neve e os sete anões”.

A CDC foi criada em 1957 por alguns ex-funcionários da Sperry Rand que não estavam satisfeitos com a visão fechada da empresa.

A nova companhia iniciou atividades com a fabricação de pequenos periféricos, como memórias de tambor magnético.

Em 1958, Seymour Cray (Estados Unidos, 1925-1996) tornou-se engenheiro-chefe da nova empresa e ele tinha grandes ideias. Já em 1959 produziu um pequeno protótipo de computador transistorizado de 6 bits, conhecido como CDC Little Character. Outros modelos se seguiram rapidamente.

Cray não queria pouco: desejava produzir o computador mais rápido do mundo, desbancando a IBM. Ele conseguiu esse feito em 1965 com o lançamento do CDC 6600 ([Figura 7.14](#)). Ainda não era um computador baseado em circuitos integrados, mas em uma tecnologia de transistores chamada *cordwood* (feixe de lenha) que otimizava muito o tempo de processamento. Além disso, ele acrescentou ao processador principal dez outros processadores para lidarem com entrada e saída e outras funções da máquina de forma independente.

O CDC 6600 tinha 400 mil transistores e mais de 160 km de fios. De 1964 a 1969 foi o computador mais rápido do mundo, sendo desbancado apenas pelo seu sucessor, o CDC 7600, outro projeto de Cray.

Quando ficou provado que o CDC 6600 era mais rápido do que qualquer outro computador no mundo, Thomas Watson Jr., presidente da IBM, ficou furioso. Lançou então o que ficou conhecido como o “memorando do faxineiro”, no qual questionava como a gigantesca IBM podia ter perdido sua liderança mundial para uma empresa com “34 funcionários, incluindo o faxineiro”.

A IBM rapidamente anunciou modelos muitos mais velozes do Sistema 360, o que freou as vendas do CDC 6600. Nunca chegou a lançá-los, porém, e a CDC respondeu com uma ação judicial antimonopólio. Quatro anos depois, a gigante concordou em indenizar a CDC com computadores e dinheiro no valor total de 600 milhões de dólares.



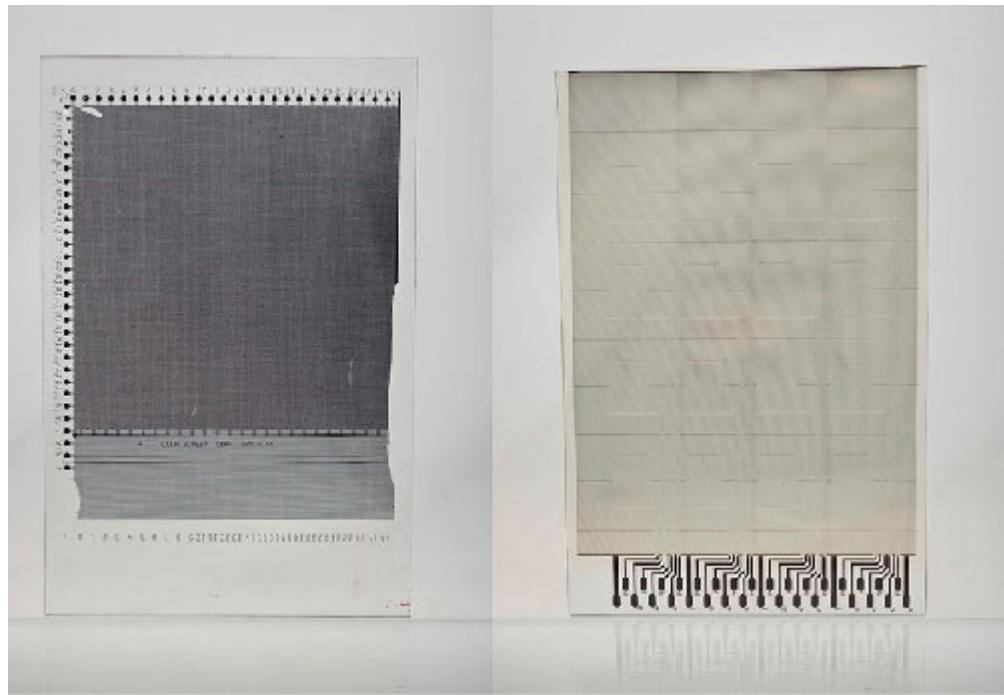
Figura 7.14: CDC 6600.<sup>16</sup>

## 7.17 Touch Screen – 1965

As telas sensíveis ao toque (*touch screen*), que hoje são o padrão de interface para tablets e smartphones, foram inventadas há meio século, mais precisamente na Inglaterra de 1965, por Eric A. Johnson. Ele descreveu a ideia de usar uma tela capaz de “sentir” o toque humano para operar um sistema de controle de tráfego aéreo, a qual foi publicada na revista *Electronic Letters* sob o título “Touch Display—A novel input/output device for computers”.<sup>17</sup> Em 1967 ele publicou um artigo mais extenso, no qual explicava em maiores detalhes seu dispositivo, bem como apresentava a foto de um protótipo. Johnson recebeu uma patente pela invenção em 1969.

Em 1972, o engenheiro Bent Stumpe (Dinamarca) foi solicitado a criar um projeto de interface de computador para o CERN (Genebra) que substituísse uma série de botões e interruptores. Stumpe projetou uma tela sensível ao toque com um número prefixado de

botões, a qual se tornou operacional em 1973 e assim permaneceu até 2008.



**Figura 7.15: Protótipos de tela sensível ao toque desenvolvidos no CERN.<sup>18</sup>**

Diversos modelos de telas sensíveis ao toque foram propostos e construídos nos anos 1970 e 1980. Porém, duas tecnologias se tornaram dominantes nos dias atuais: a resistiva e a capacitiva.

Uma tela resistiva é formada por várias camadas, entre as quais estão duas camadas cada qual com minúsculas linhas condutoras que são separadas por pequenos espaços. As camadas são colocadas uma sobre a outra de forma que uma tenha as linhas na vertical e a outra na horizontal. Elas são separadas entre si por um espaçamento minúsculo feito de tal maneira que se um dedo ou apontador colocar uma pequena pressão sobre a camada superior, as linhas condutoras nela tocarão as linhas da camada inferior, estabelecendo assim o contato. Uma voltagem elétrica é aplicada a uma camada e sentida pela outra, de forma que se possa transmitir para o computador as coordenadas nas quais a tela está sendo

tocada. Essa tecnologia é usada ainda em alguns dispositivos que demandam alta resistência e baixo custo. Sua desvantagem em tablets é o fato de que exige pressão, não apenas o toque do dedo. A outra tecnologia predominante nos dias atuais é a capacitiva, a qual tem muitas diferentes formas de implementação, mas via de regra trata-se de uma superfície condutora sobre uma superfície isolante. Ao colocar o dedo sobre a superfície condutora, como a pele humana conduz eletricidade haveria uma distorção no potencial eletrostático da tela naquele ponto. Essa distorção poderia ser detectada e transformada em coordenadas, de forma que se saiba qual ponto foi tocado. Essa tecnologia é muito mais sensível do que a resistiva, mas é de implementação mais cara e só funciona com o dedo humano ou materiais condutores. Por exemplo, se você usar uma luva de borracha, não conseguirá nada com uma tela capacitiva.

## 7.18 Memória DRAM – 1966

Robert H. Dennard (Estados Unidos, 1932) foi contratado em 1958 pela divisão de pesquisa da IBM, onde ele intencionava passar uns dois anos apenas para aprender as manhas da indústria, mas na qual acabou ficando por mais de 40 anos.

Ele estava a cargo de uma pequena equipe que estudava memórias de computador. Na época, um bit de memória precisava de seis transistores para ser mantido. Em 1966, porém, ele teve a ideia de usar um capacitor para segurar o sinal elétrico do bit. O problema com capacitores é que eles vão perdendo a carga elétrica rapidamente se ela não for reforçada. Dennard percebeu que bastaria usar um transistor para efetuar esse reforço com a velocidade necessária.

Assim, em vez de usar seis transistores para representar um único bit de memória, bastaria um transistor e um capacitor. Uma das consequências disso foi um aumento significativo na capacidade de

memória dos circuitos integrados, de forma que hoje bilhões de bits podem ser representados em apenas um circuito.

Essa nova memória baseada em capacitores foi chamada de DRAM, para Dynamic Random Access Memory. O “dynamic” vem do fato de que o capacitor precisa ser reforçado dinamicamente pelo transistor para não perder a carga.

Em 1970, a recém-criada empresa Intel produziu um chip (circuito integrado) de número 1103, uma memória DRAM de 1 kB. Houve muita demanda das indústrias por este produto e em 1972 ele se tornou o circuito integrado mais vendido no mundo. É conhecido hoje como o circuito que acabou com a era das memórias de núcleo magnético.

## 7.19 Chatterbot ELIZA – 1966

Entre 1964 e 1966 o professor Joseph Weizenbaum (Alemanha, 1923-2008) do MIT trabalhou em um programa de computador chamado ELIZA. Ele seria o primeiro programa a simular a capacidade de conversação de um ser humano de maneira semelhante à proposta no Teste de Turing. Assim, ele foi o primeiro *chatterbot* da história.

ELIZA era um sistema de casamento de padrões que podia ser carregado com diferentes *scripts*, e cada qual corresponderia a um diferente perfil de conversação. O mais famoso destes perfis foi o DOCTOR, que simulava um psicanalista rogeriano que supostamente trabalharia com a técnica de terapia centrada na pessoa.

ELIZA funcionava no sistema de compartilhamento de tempo MAC do MIT. O sistema consistia de um computador central ligado a um conjunto de terminais que eram basicamente máquinas de escrever elétricas. Cada usuário do computador tinha a impressão de que ele sozinho operava o computador, graças ao sistema de compartilhamento de tempo. O usuário podia datilografar comandos na máquina e o computador escrevia respostas na mesma máquina.

Weizenbaum aproveitou essa característica para implementar o programa ELIZA, que “lia” expressões em linguagem natural datilografadas por um usuário e respondia também em linguagem natural, como se fosse um ser humano. Muitas pessoas foram enganadas pelo sistema, pelo menos no início, pois após algum tempo ficava claro que não se tratava de um ser inteligente, mas de uma máquina que simplesmente analisava padrões no texto e produzia respostas baseadas em regras que indicavam como uma determinada pergunta deveria ser convertida em uma determinada resposta.

Abaixo, um exemplo parcial de um dos diálogos de ELIZA. As frases digitadas pelo humano estão em minúsculas e as respostas do computador em maiúsculas. O exemplo foi traduzido diretamente do artigo de Weizenbaum de 1966:

- Os homens são todos iguais.
- DE QUE MANEIRA?
- Eles estão sempre nos incomodando com uma coisa ou outra.
- VOCÊ CONSEGUE PENSAR EM UM EXEMPLO ESPECÍFICO?
- Bem, meu namorado me fez vir aqui.
- SEU NAMORADO FEZ VOCÊ VIR AQUI.
- Ele disse que eu estou deprimida a maior parte do tempo.
- LAMENTO OUVIR QUE VOCÊ ESTÁ DEPRIMIDA.
- ...

O programa funciona de forma muito simples: ele procura identificar palavras chave no texto, como “deprimida”, no exemplo acima. Ele é capaz de identificar categorias de palavras, por exemplo, X poderia ser uma categoria de palavras negativas como “deprimida”, “cansada”, “triste” etc. Depois, uma regra diz que ele deve responder a uma frase com a palavra negativa da categoria X usando a frase “lamento ouvir que você está X”. Assim, o computador não precisa saber, e realmente não sabe, o que

significa a frase. Se depois o humano dissesse “hoje estou cansada”, ele responderia “lamento ouvir que você está cansada”.

ELIZA era sofisticado a ponto de ter prioridade entre as regras. Assim, se o programa estivesse lendo um texto e encontrasse uma palavra-chave, ele não daria a resposta imediatamente, ele continuaria lendo o texto e caso encontrasse outras palavras-chave, ele ao final escolheria a de maior prioridade e aplicaria a regra correspondente a ela somente.

O programa era ainda capaz de prover respostas genéricas caso o humano fugisse do tópico. Por exemplo, se a pessoa perguntasse “qual é seu compositor favorito?”, o computador não teria informações para dar uma resposta aceitável. Assim, responderia algo como: “Por que essa pergunta lhe interessa?”

Weizenbaum implementou o sistema na linguagem SLIP (Symmetric List Processor), sua própria versão de uma linguagem funcional semelhante a LISP.

O nome ELIZA foi escolhido em homenagem à personagem Eliza Doolittle, da peça teatral *Pigmaleão* de George Bernard Shaw (Irlanda, 1856-1950). Ela teria sido uma florista que procurou um professor para ter aulas de eloquência e etiqueta e assim poder participar da alta sociedade londrina. Esta peça também teve uma famosa versão musical na Broadway, *My Fair Lady*.

Em 1972 ELIZA teve um encontro histórico com o programa PARRY, um chatterbot que simulava uma pessoa com esquizofrenia. O diálogo, ficou famoso por ter sido a primeira vez que dois computadores conversaram um com o outro usando linguagem humana. O diálogo completo pode ser consultado em <https://tools.ietf.org/html/rfc439>.

## 7.20 OS/360 – 1967

O Sistema 360 da IBM era mais do que hardware. A empresa lançou em 1964 um ambicioso projeto para desenvolver um sistema operacional, o OS/360. Na verdade, ele seria uma família de

sistemas operacionais totalmente compatíveis e crescentemente mais completos. Para os computadores menores com 8 kB de memória haveria o BOS/360 (Basic Operating System). Para máquinas médias com unidades de fita de 16 kB haveria o TOS (Tape Operating System). Finalmente, para as maiores máquinas com unidades de disco haveria o DOS (Disk Operating System).

A empresa escolheu Frederick J. Brooks Jr. (Estados Unidos, 1931) como líder do projeto. Brooks havia sido um dos mais brilhantes alunos de Robert Aiken, criador do Harvard Mark I 20 anos antes. O orçamento inicial para o projeto era de 25 milhões de dólares.

Inicialmente, o sistema deveria permitir multiprogramação, ou seja, que vários programas rodassem na máquina ao mesmo tempo, compartilhando fatias de tempo do processador. Porém, logo adiaram a ideia quando perceberam a complexidade que isso traria ao projeto.

Ele se iniciou em 1964 com um grupo de 12 designers liderando uma equipe de 60 programadores. Porém, começou a atrasar devido a pequenos problemas. Em 1965 a equipe foi aumentada para 150 pessoas e o cronograma já estava seis meses atrasado. Testes foram feitos e se percebeu que o sistema estava aquém do esperado, precisando de extensiva reprogramação.

Brooks continuava contratando cada vez mais programadores, até chegar a ter uma equipe de mil pessoas. O OS/360 foi finalizado e entregue apenas em 1967, com um ano de atraso e a um custo de meio bilhão de dólares. Posteriormente, Brooks reconheceu que contratar mais programadores para um projeto que já estava atrasado era exatamente a coisa errada a se fazer. Isso ficou conhecido em engenharia de software como a “Lei de Brooks”. Nas palavras dele: *“The bearing of a child takes nine months, no matter how many women are assigned.”*<sup>19</sup> Com isso, ele reconhecia uma importante característica da indústria de software: um projeto de software precisa de tempo para amadurecer e organização para funcionar. Posteriormente, em 1975, Brooks publicou um livro que

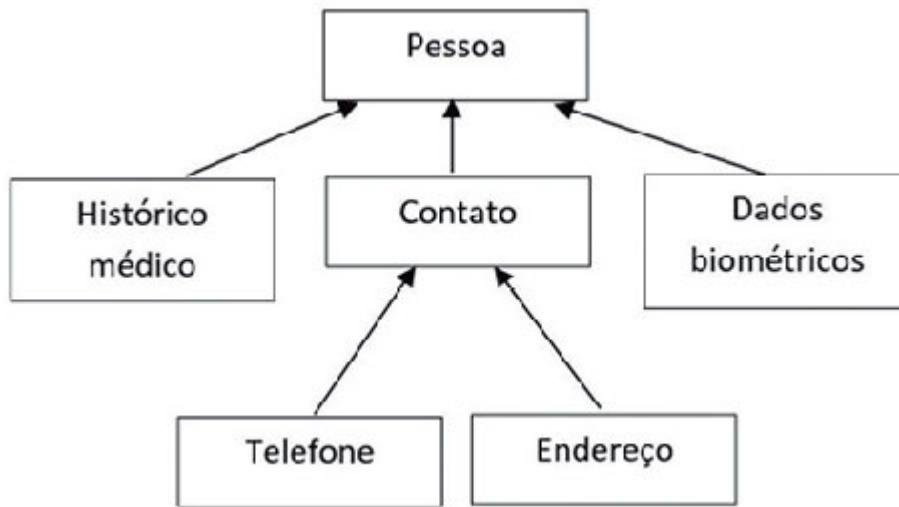
seria um marco para a engenharia de software *The Mythical Man-Month*,<sup>20</sup> no qual apresentava essa e outras ideias.

## 7.21 MUMPS – 1967

Outra linguagem desenvolvida ainda nos anos 1960 foi MUMPS (Massachusetts General Hospital Utility Multi-Programming System). Ela foi projetada por Neil Pappalardo e alguns colegas que trabalhavam no Hospital Geral de Massachusetts em Boston entre 1966 e 1967. A linguagem foi criada especificamente para permitir maior facilidade no tratamento de dados médicos e, especialmente, dados na forma de texto.

Como o hospital possuía um DEC PDP-7, era natural que a linguagem acabasse tendo a interatividade própria das linguagens interpretadas, diferentemente das compiladas como COBOL e FORTRAN.

MUMPS implementava desde o início facilidades para trabalhar com estruturas de dados hierárquicas, ou seja, uma maneira de organizar os dados de forma que uma informação possa ter “dependentes”. Assim, por exemplo, o registro referente a uma pessoa poderia ter como dependentes seu contato, histórico médico e dados biométricos. O contato, por sua vez, poderia ter como dependentes o telefone e o endereço etc. A [Figura 7.16](#) mostra graficamente esse exemplo de estrutura.



**Figura 7.16: Exemplo de estrutura de dados hierárquica.**

Nos anos 1970, a linguagem MUMPS e seu banco de dados integrado tornaram-se muito populares na comunidade médica. Por ser de domínio público, muitas versões foram implementadas, e a partir de 1972 ela passou por um processo de padronização. O padrão ANSI foi aprovado em 1977 e revisado em 1990 e 1995. Muitas empresas e organizações da área de saúde ainda usam MUMPS para desenvolver seus sistemas hoje.

## 7.22 Simula – 1967

O Sketchpad em 1963 já implementava alguns conceitos do que seria posteriormente conhecido como “programação orientada a objetos”. Porém, ele não era uma linguagem de programação, mas um programa para desenhar. A programação orientada a objetos surgiu, embora ainda não tivesse esse nome, com o projeto de uma linguagem de simulação conhecida como Simula 67.

O projeto de Simula foi iniciado por Kristen Nygaard (Noruega, 1926-2002) da Universidade de Oslo, que escrevia programas de simulação desde 1957. Nygaard tentava achar formas mais adequadas para descrever simulações de eventos discretos. Uma simulação baseada em eventos discretos é um sistema onde se

programa o que cada um dos componentes deve fazer em cada instante de tempo. Depois, um motor de simulação vai executar o programa para cada componente repetidas vezes, provocando um comportamento coletivo observável. Por exemplo, se um componente tem seu comportamento definido como mover-se um passo para frente e girar um grau em torno de seu eixo, ao ser colocado sob simulação discreta esse componente vai parecer estar andando em círculos, porque a cada instante de tempo ele anda um passo e gira um grau.

Em 1962, Nygaard percebeu que precisava de alguém com mais conhecimentos em computação e aproximou-se de Ole-Johan Dahl (Noruega, 1931-2002) para lançar ainda este ano a primeira definição de Simula I, uma linguagem de propósito especial assemelhada a ALGOL. Em 1965, a linguagem foi implementada e tornou-se disponível no UNIVAC 1107 e, anos depois, no Burroughs B5500 e no computador russo URAL-16.

Nygaard e Dahl conseguiram deixar a linguagem relativamente popular na Europa a partir de palestras e cursos, e chegou-se a formar um comitê para sua padronização. Este grupo, formalizou no início de 1968 a versão SIMULA 67, a qual incorporava vários conceitos que mais de uma década depois passaram a ser comuns em linguagens de programação, mas na época não eram.

Eles se inspiraram no conceito de classe definido por Charles A. R. Hoare (Sri Lanka, 1934) e estenderam suas ideias, implementando-as na linguagem. Entre outros conceitos que hoje são fundamentais nas linguagens de programação, SIMULA 67 já implementava:

- *Classe e objeto*. A ideia de que uma classe é uma espécie de definição para instâncias ou objetos, que são a realização dessa definição. Por exemplo, “Pessoa” seria uma classe e “João” e “Maria” seriam duas instâncias.
- *Subclasse*. Classes mais abstratas podem ser especializadas em subclasses que compartilham parte da definição. Por exemplo, se a classe “mamífero” é definida como formada por

animais que têm pelos e que mamam ao nascer; e se “cão” e “gato” são subclasses de mamífero, então, “cão” e “gato” podem ter suas especificidades, mas eles compartilham a definição de “mamífero”, ou seja, todos os cães e gatos têm pelos e mamam ao nascer.

- *Objetos ativos*. Em Simula qualquer objeto pode funcionar como uma máquina autônoma, ou seja, cada um deles pode operar como se o processador do computador fosse só seu, mesmo que na realidade eles compartilhem fatias de tempo do processador.

A primeira publicação oficial sobre Simula 67 ocorreu em uma *working conference* sobre linguagens de simulação da IFIP em Oslo no ano de 1967. Simula teve grande influência na definição das modernas linguagens de programação orientadas a objetos. Dahl e Nygaard receberam a Medalha John von Neumann e o Prêmio Turing por suas contribuições à ciência da computação e pela introdução dos conceitos fundamentais de orientação a objetos em linguagens de programação.

## 7.23 Logo – 1967

Seymour Papert (África do Sul, 1928) é um matemático que trabalhou por algum tempo com o epistemologista Jean Piaget (Suíça, 1896-1980), que o influenciou com sua teoria do Construtivismo – uma forma inovadora de compreender a aprendizagem como processo que envolve a ação do sujeito que aprende, e não apenas a “transmissão de conhecimento”.

Quando chegou no MIT em meados dos anos 1960, Papert fundou, juntamente com Marvin Minsky (Estados Unidos, 1927), o Laboratório de Inteligência Artificial. Papert trabalhou com a equipe de Wallace Feurzeig (Estados Unidos, 1927-2013) na criação da linguagem Logo. Participaram da equipe também Daniel G. Bobrow (Estados Unidos, 1935) e Cynthia Solomon. O Logo não foi criado na universidade, porém, mas em uma empresa de pesquisas de

Cambridge chamada BBN Technologies (originalmente Bolt, Beranek and Newman).

A primeira implementação de Logo foi feita em LISP em uma máquina PDP-1. Seu objetivo era desenvolver uma “terra da matemática” na qual as crianças pudessem experimentar com números, palavras, funções e comandos.

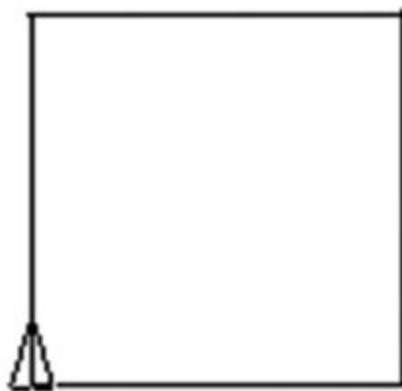
Ao contrário da maioria das demais linguagens da época, cujos nomes derivavam de acrônimos ou siglas, Logo teve seu nome inspirado na palavra grega para “pensamento”. Possivelmente, Logo foi a primeira linguagem de programação destinada a desenvolver a habilidade humana de pensar. Enquanto FORTRAN foi criada para resolver problemas científicos e COBOL para problemas comerciais, Logo foi criada para aprimorar o potencial humano.

Inicialmente, tratava-se de apenas mais um dialeto de LISP, mas com a introdução do conceito de *turtle graphics* seu potencial educacional foi enormemente aumentado. *Turtle graphics* é o nome que se dá genericamente a sistemas de desenho baseados em um cursor e geometria radial. Originalmente, tratava-se de um pequeno robô que lembrava o robô programável de Heron de Alexandria, mas com uma caneta que podia desenhar sobre folhas de papel.

O robô obedecia a comandos como andar para frente ou para trás um certo número de “passos” ou girar um certo ângulo para a direita ou para a esquerda. Por exemplo, considere que o comando para mover a tartaruga para a frente é FD (forward) e que para girar a direita é RT (right), ambos seguidos de um número de passos ou ângulo. O seguinte programa em Logo faria a tartaruga desenhar um quadrado na tela:

REPEAT 4 [FD 60 RT 90]

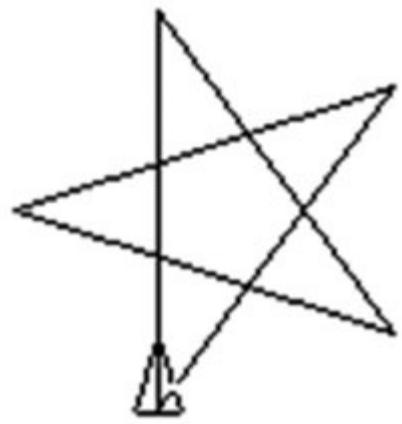
Ou seja, a tartaruga deveria repetir quatro vezes a sequência de andar 60 passos e girar 90°. O resultado é um quadrado, como mostrado na [Figura 7.17](#)



**Figura 7.17:** Um quadrado desenhado com Logo.<sup>21</sup>

O quadrado, então, poderia ser definido como um novo comando para a linguagem, que é extensível, e usado para formar outras figuras.

A partir de pouquíssimos conhecimentos em programação é possível começar a experimentar desafios intelectuais. Por exemplo, sabendo-se como desenhar um quadrado podemos inferir como seria o programa para desenhar uma estrela de cinco pontas, como mostrado na [Figura 7.18](#)?



**Figura 7.18:** Estrela desenhada com Logo.<sup>22</sup>

A resposta seria:

## REPEAT 5 [FD 60 RT 144]

O processo de raciocínio a ser desenvolvido seria justamente descobrir qual o ângulo que a tartaruga deve girar para obter o efeito desejado, o que pode ser descoberto com a aplicação de trigonometria simples ou por tentativa e erro. Entre muitas outras possibilidades, uma seria perceber que, para desenhar o pentagrama, a tartaruga precisa dar duas voltas completas sobre o próprio eixo, ou seja,  $2 \cdot 360^\circ$ . Assim,  $720^\circ$  dividido por 5 resulta em cinco giros de  $144^\circ$ . Por outro lado, girando  $360/5$ , ou  $72^\circ$ , a tartaruga desenharia um pentágono e não uma estrela. A vantagem da máquina é que ela dá feedback automático para as tentativas de raciocínio do estudante.

Com a introdução dos microcomputadores nos anos 1970, que implementavam essa linguagem e *turtle graphics*, seu uso tornou-se muito popular em escolas do mundo todo. No Brasil, uma das maiores incentivadoras dessa tecnologia desde os anos 1980 foi Lea da Cruz Fagundes (Brasil, 1930). Psicóloga construtivista, Lea também é uma das pioneiras da área de Informática na Educação no Brasil.

## 7.24 Forth – 1968

Charles H. Moore (Estados Unidos, 1938) era um programador que não estava satisfeito com as linguagens de programação de alto nível que existiam em 1968. Ele colocou na cabeça que podia fazer mais: queria ser capaz de produzir muito mais programas do que conseguia com FORTRAN. Com este objetivo em mente, criou uma linguagem que viria a ter características únicas. De certa forma, como explicaremos adiante, ela não é nem compilada nem interpretada, mas alinhavada. A linguagem produz um código muito eficiente e compacto, que hoje em dia é bastante adequado para dispositivos eletrônicos embutidos ou autônomos como minirrobôs. Ela é baseada no uso de uma pilha de dados, o que faz com que as operações sejam *pós-fixadas*, o que explicaremos na sequência.

Forth é uma linguagem muito simples; um sistema capaz de rodá-la pode ser implementado com pouquíssimos bytes. Ao contrário de outras, não há um compilador que precisa traduzir expressões de alto nível em linguagem de máquina. Forth é constituída por um conjunto inicial de rotinas, cada qual com um endereço de memória definido. Um pequeno programa, o alinhavador de código, faz um loop no qual lê em uma posição inicial da memória um valor que deve corresponder ao endereço de uma das rotinas. Então ele desvia o controle para esse endereço, fazendo com que a rotina seja executada. Quando ela termina, o controle retorna ao alinhavador que então busca o endereço da rotina seguinte. Assim, um programa Forth consiste em última análise em uma sequência de endereços de memória nos quais estão as rotinas que devem ser executadas.

Porém, um programador Forth não vai usar endereços de memória diretamente em seus programas, ele vai usar “palavras”. Cada palavra em Forth é traduzida em um endereço de memória que o alinhavador vai executar. A associação entre as palavras e endereços de memória é armazenada em uma estrutura chamada dicionário.

Programar em Forth basicamente consiste em criar novas “palavras”, escrevendo sequências de palavras que formam um procedimento.

Uma característica única de Forth é o fato de que toda a manipulação de dados se dá através de uma pilha. Um número inteiro, por exemplo, consiste em um comando, ou palavra, que tem como efeito colocar o valor representado pelo número no topo da pilha. Assim, um programa como:

20 30 50

Colocará estes três números na pilha, ficando 50 no topo. Se em seguida o programador escrever a palavra “+”, isso ativará a rotina que soma os dois valores do topo. Assim, o computador vai somar 30 e 50, colocando o resultado, 80, no topo da pilha, sobre o 20. Se

em seguida o programador escrever a palavra "", o computador vai multiplicar os dois números do topo, ou seja, 2080, o que resulta em 1.600.

Assim, o programa completo para calcular  $20*(30+50)$  seria escrito em Forth como:

20 30 50 + \*

Essa forma de escrita, na qual os operadores são colocados depois dos parâmetros é chamada de "notação pós-fixada" ou "notação polonesa reversa".

Leo Brodie publicou um livro interessante sobre Forth em 1987 que ficou famoso por usar caricaturas para exemplificar o funcionamento da linguagem (Figura 7.19).

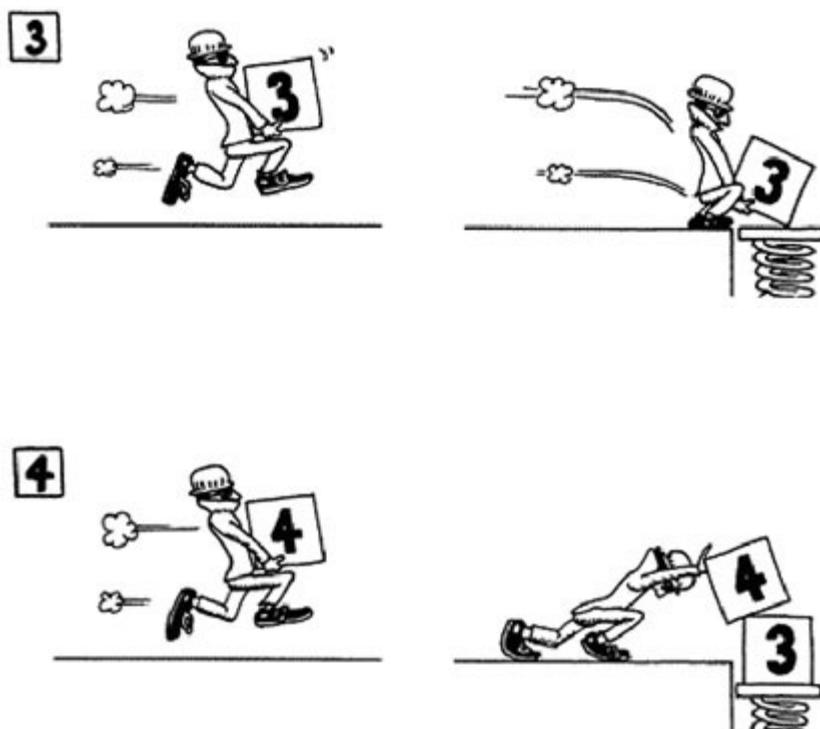




Figura 7.19: Caricaturas sobre Forth de Leo Brodie.<sup>23</sup>

Um dos desenhos que ficou particularmente famoso foi o dragão do *swap* (Figura 7.20), um comando que troca de posição os dois valores mais ao topo da pilha.

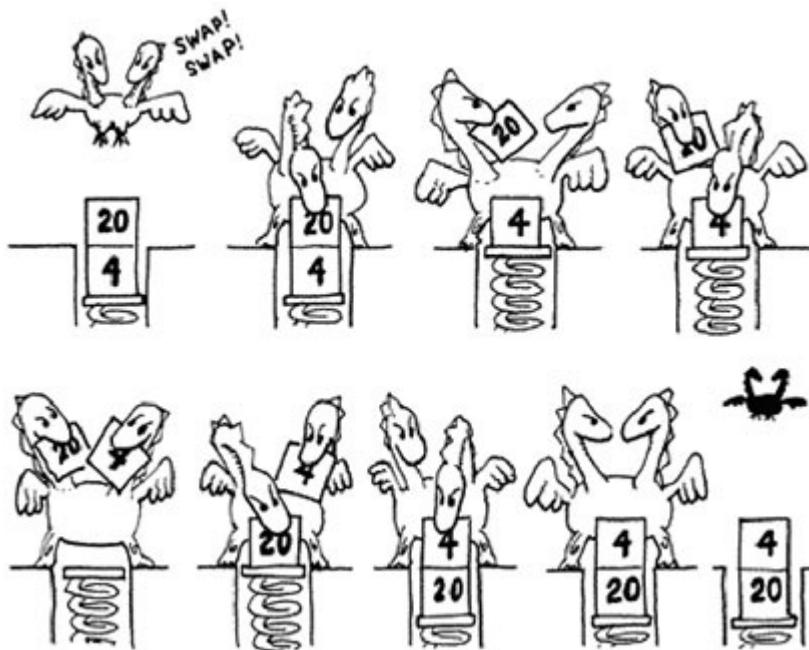


Figura 7.20: Dragão do Swap.<sup>24</sup>

A primeira implementação de Forth foi feita por Moore em um IBM 1130. Ele considerava que essa seria uma linguagem de quarta geração e queria que ela se chamassem “Fourth”, mas o 1130 só permitia identificadores com no máximo cinco caracteres. Assim, ele a chamou da maneira como conhecemos hoje.

## 7.25 BESM 6, o Computador Soviético – 1968

Sim, a Rússia (então parte da União Soviética) também produziu computadores desde cedo. Não ficaram tão conhecidos entre nós, possivelmente devido à Guerra Fria, mas já em 1948 ela começou a construção do MESM (Malaya Elektronno-Schetnaya Mashina<sup>25</sup>), que ficou pronto em 1951, sendo um dos primeiros computadores da Europa.

Pouco depois iniciou-se a construção da série BESM (Bolshaya Elektronno-Schetnaya Mashina<sup>26</sup>). O BESM-1 foi completado em 1952 e possuía 5 mil válvulas, tornando-se então o computador mais rápido da Europa e já usando memória de núcleo magnético.

A partir do BESM-3 os soviéticos passaram a usar transistores. Em 1968, um BESM-4 foi usado para produzir uma das primeiras animações criadas em computador. O modelo matemático de um gato deu origem a um desenho animado chamado *Kitty*, que pode ser assistido em <https://www.youtube.com/watch?v=0O4mm3hXNgA>.

Porém, o mais famoso e importante modelo da série foi o BESM-6 ([Figura 7.21](#)), projetado a partir de 1965 e concluído em 1968. Tratava-se de um computador transistorizado com processador de 48 bits e capaz de processar um milhão de instruções por segundo (o mais rápido do mundo na época era o CDC 6600, com capacidade de 3 milhões de instruções por segundo).



**Figura 7.21: BESM-6.<sup>27</sup>**

Mais de 300 unidades do BESM-6 foram fabricadas até 1987, pouco antes do colapso da União Soviética. Ele foi usado extensivamente no Programa Espacial Soviético. O cálculo dos parâmetros orbitais da Soyus levou cerca de um minuto para ser processado pelo BESM, enquanto os computadores da NASA precisaram de 30 minutos para fazer cálculos similares para o programa Apollo.

## **7.26 Espada de Dâmocles de Sutherland e o Início da Realidade Virtual – 1968**

Ivan Sutherland não sossegou depois de criar o Sketchpad. Em 1968, juntamente com Bob Sproull, ele criou um dispositivo que é considerado o primeiro equipamento (capacete) de realidade virtual e realidade aumentada, o Sword of Damocles<sup>28</sup>.

A animação mostrada no aparelho era muito simples, apenas uma sala modelada com linhas desenhadas. Porém, o equipamento conseguia dar a impressão de profundidade, ou terceira dimensão, por mostrar para cada olho uma imagem ligeiramente diferente.

Para que o usuário pudesse olhar para diferentes pontos da sala, ele devia mover a cabeça, mas como na época ainda não havia dispositivos sem fio capazes de sentir movimentos, como hoje, eles fixaram o capacete em um suporte de forma que os movimentos giratórios pudessem ser lidos por sensores fixos.

A aparelhagem toda era bem grande e dava a impressão de que o usuário estava com uma enorme espada sobre a cabeça. Daí derivou o nome “Espada de Dâmocles”, em função da antiga lenda grega homônima.

O dispositivo era considerado como realidade aumentada porque os visores oculares eram parcialmente transparentes, permitindo ao usuário continuar vendo o mundo real enquanto via também as imagens geradas pelo computador mescladas a ele.

## 7.27 Oki Wiredot, a Primeira Impressora Matricial – 1968

As primeiras impressoras de computadores eram basicamente máquinas de escrever modificadas. Nestas máquinas, cada letra ou símbolo era gravada em alto relevo em um martelo, o qual, batido contra uma folha ou fita de carbono superposta a uma folha de papel, produzia o desenho da letra.

Porém, em 1968, a fabricante japonesa OKI criou um modelo diferente: a impressora matricial OKI Wiredot.

Ainda se tratava de uma impressora de impacto, sendo, portanto, diferente das jato-de-tinta ou laser. Entretanto, em vez de ter um martelo específico para cada letra, ela possuía um conjunto de agulhas que se movia sobre o papel. A cada instante, algumas agulhas batiam no papel e outras não. O movimento combinado com as batidas produzia na folha o desenho de qualquer letra ou símbolo que se quisesse, bem como desenhos complexos. A [Figura 7.22](#) mostra um exemplo de impressão obtida com essa tecnologia.

ystem where a  
ld allow us t  
mercial supplier.

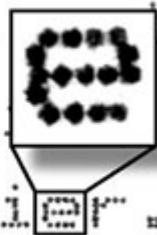


Figura 7.22: Exemplo de impressão matricial.<sup>29</sup>

As matriciais tornaram-se o padrão da indústria de impressoras a partir dos anos 1970 até serem suplantadas pelas jato-de-tinta e laser; mas como estas últimas não são baseadas em impacto, elas não são adequadas para imprimir várias cópias carbono de uma vez. Assim, em aplicações que exigem cópias carbono, ou impressão baseada em impacto por qualquer outro motivo, as matriciais ainda continuam sendo usadas.

Há basicamente dois tipos de impressoras matriciais, as de coluna e as de linha. As impressoras de coluna têm uma fileira vertical de agulhas que corre horizontalmente com um cabeçote, imprimindo assim caractere por caractere em uma linha. Elas são usualmente mais lentas, com capacidade de no máximo 500 caracteres por segundo.

Já as impressoras de linha têm uma ou mais fileiras horizontais de agulhas com largura equivalente à do papel. Assim, elas literalmente imprimem uma linha inteira de cada vez, sendo bem mais rápidas, com velocidades que excedem facilmente 1.000 caracteres por segundo.

## 7.28 HP 9100A, a Primeira Calculadora Programável Pessoal – 1968

Em 1968 a Hewlett-Packard criou a HP 9100A, considerada a primeira calculadora programável de uso pessoal. Na verdade, na época ela foi responsável por cunhar pela primeira vez o termo “computador pessoal”, isso vários anos antes do Altair e Xerox Alto.

Porém, por questões de marketing, a HP não quis que ela fosse chamada de computador, mas de calculadora. Isso porque na época computadores era máquinas enormes e para comprá-las as empresas passavam por processos complexos que envolviam os diretores de mais alto escalão. Qualquer gerente médio podia comprar calculadoras, contudo.



Figura 7.23: HP 9100A.<sup>30</sup>

Mesmo assim, o preço de uma HP não era acessível para a maioria dos mortais: cerca de 5 mil dólares, o que equivaleria a algo como 34 mil dólares em 2015.

O modelo 9100A inaugurou a tradição da HP na construção de calculadoras científicas programáveis e também o uso da notação polonesa reversa em suas linguagens de programação.

## 7.29 Dynabook de Alan Kay – 1968

Alan C. Kay (Estados Unidos, 1940) foi aluno de Ivan Sutherland em meados dos anos 1960, com o qual conheceu o Sketchpad e seus conceitos subjacentes. A partir de 1967, ele começou a trabalhar em um conceito de máquina programável por usuários, inclusive

crianças, que ele chamou de máquina FLEX. Seu encontro com Seymour Papert em 1968 e sua introdução aos conceitos relacionados a Logo o motivou a continuar investindo nessa direção. Outra influência importante foi Vannevar Bush e seu Memex.

Em 1970, a Xerox, ciente de que seu modelo de negócio baseado em cópias de papel estaria com os dias contados na medida em que os computadores fossem cada vez mais usados para armazenar e consultar informação, resolveu investir em pesquisas para tentar desvendar e antecipar esse futuro para si própria. Foi assim fundado o centro de pesquisas Xerox PARC em Palo Alto, Califórnia, que ficou famoso por várias invenções relacionadas à computação nos anos 1970. Kay foi contratado como um dos pesquisadores deste centro e logo aliou-se ao projeto Smalltalk, que visava desenvolver uma linguagem de programação acessível ao público infanto-juvenil. Seus conhecimentos sobre orientação a objetos oriundos do Sketchpad o levaram a trabalhar para que Smalltalk fosse efetivamente uma linguagem orientada a objetos.

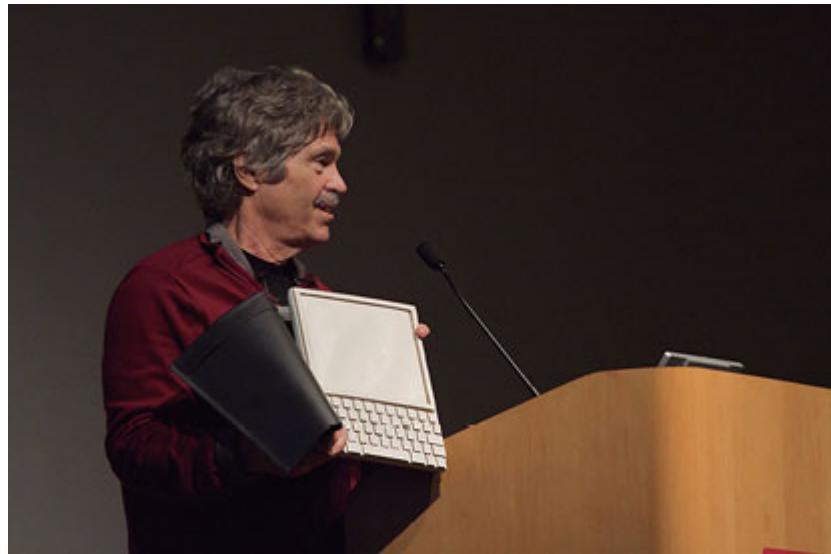
Tanto Kay quanto o PARC foram extremamente prolíficos em gerar novas ideias, mas a que vamos descrever aqui é uma concepção de 1968, o Dynabook, que seria um computador pessoal para crianças de todas as idades. O conceito foi apresentado em um artigo de 1972 no qual Kay enfatizava que uma máquina como o Dynabook seria importante para permitir a aprendizagem construtivista, melhorando o sistema educacional da época.

No início do artigo ele mostra duas crianças sentadas na grama com Dynabooks jogando *Spacewar!* em rede e aprendendo sobre gravidade e movimento uniforme enquanto se divertem.

O Dynabook seria muito semelhante aos nossos modernos notebooks ou tablets. Deveria pesar menos de 1 kg, ter o tamanho de um livro, com uma tela e um teclado portáteis e interface gráfica para o usuário, semelhante ao Sketchpad. A [Figura 7.24](#) mostra Kay e seu protótipo. O Dynabook deveria também poder conversar com

outros computadores através da linha telefônica, antecipando os modems construídos anos mais tarde.

Com o Dynabook seria possível “instanciar” (baixar) livros na máquina, em vez de comprá-los em papel. A tela teria que ser de cristal líquido, ele sugeriu, porque consumiria menos energia, visto que o aparelho seria mantido por baterias. O teclado não deveria ter partes móveis, mas ser sensível ao toque e produzir um feedback sonoro quando tocado. O preço do Dynabook deveria ser equivalente ao de uma TV, cerca de 500 dólares na época, uma previsão que surpreendentemente se realizou atualmente, visto que os preços de televisores e notebooks ou tablets se equiparam mais ou menos neste patamar.



**Figura 7.24: Alan Kay segurando um protótipo do Dynabook.<sup>31</sup>**

Muito à frente de seu tempo, o Dynabook nunca foi construído. No entanto, ele serviu de inspiração para um dos primeiros computadores pessoais, o Xerox Alto, além das interfaces gráficas dos computadores Apple e do sistema operacional Windows.

## **7.30 A Arte de Programação de Computadores de Knuth – 1968**

Possivelmente o maior e mais importante tratado sobre programação de computadores é de autoria de Donald Knuth (Estados Unidos, 1938). O trabalho começou a ser publicado em 1968 e ainda não foi terminado.

Knuth iniciou o projeto em 1962, inicialmente planejado como um único livro de 12 capítulos. Ele se motivou a isso quando começou a escrever sobre programação de compiladores e percebeu que poderia ir bem mais além, produzindo um tratado completo e detalhado sobre técnicas de programação e análise de algoritmos.

Ele finalizou o primeiro manuscrito em 1966, com 3 mil páginas, assumindo que a transcrição para a versão tipográfica seria da ordem de 5 para 1. Porém, na verdade foi da ordem de 1,5 para 1, portanto, o livro finalizado ficaria com cerca de 2 mil páginas. Bastante coisa, especialmente considerando que ele era recém-formado; nenhuma editora aceitaria esse risco. Porém, com apoio e entusiasmo de outros pesquisadores Knuth conseguiu aprovar a produção de um projeto de sete volumes, cada qual contendo um ou dois capítulos do original. A organização dos volumes ficou assim:

- Volume 1 – algoritmos fundamentais
  - o Capítulo 1 – conceitos básicos
  - o Capítulo 2 – estruturas de informação
- Volume 2 – algoritmos seminuméricos
  - o Capítulo 3 – números aleatórios
  - o Capítulo 4 – aritmética
- Volume 3 – ordenação e pesquisa
  - o Capítulo 5 – ordenação
  - o Capítulo 6 – pesquisa
- Volume 4 – algoritmos combinatórios (parcialmente publicado)
  - o Capítulo 7 – pesquisa combinatória

- o Capítulo 8 – recursão
- Volume 5 – algoritmos sintáticos (lançamento estimado para 2020)
  - o Capítulo 9 – escaneamento léxico, pesquisa em texto e compressão de dados.
  - o Capítulo 10 – técnicas de parsing.
- Volume 6 – Teoria das linguagens livres de contexto (planejado)
- Volume 7 – Técnicas de compilação (planejado)

Knuth ofereceu desde o início um dólar hexadecimal (2,56 dólares) para qualquer pessoa que encontrasse um erro em um de seus livros. As revisões de seu trabalho, assim, contribuíram para sua popularidade e alta precisão, sendo hoje um dos mais respeitados e influentes na área de algoritmos. A [Figura 7.25](#) mostra alguns volumes publicados.



**Figura 7.25: *The Art of Computer Programming*.**<sup>32</sup>

No livro há exercícios de vários níveis de dificuldade, desde aquecimentos até problemas de pesquisa até hoje não resolvidos. Bill Gates teria afirmado certa vez que alguém que conseguisse ler os livros de Knuth e entender tudo, poderia lhe mandar o currículo para ser imediatamente contratado.

## 7.31 Crise do Software – 1968

Em 1968, a OTAN organizou a primeira conferência de Engenharia de Software em Garmisch, na Alemanha. Lá, alguns participantes cunharam e passaram a divulgar o termo “crise do software”, que foi por décadas uma das questões mais comentadas dentro da área de engenharia de software. Basicamente, constatou-se já naquele tempo que os computadores tinham ficado muito poderosos e que, pela falta de técnicas e processos adequados, a produção de software não acompanhava esse crescimento.

Basicamente, projetos de software custavam mais do que o previsto e passavam do prazo, os produtos tinham qualidade insatisfatória e eram difíceis de manter; isso quando não eram simplesmente cancelados antes mesmo de serem lançados.

O termo foi bastante divulgado por Edsger Dijkstra (Holanda, 1930-2002), em um artigo publicado em 1972. Ele se intitulava como o primeiro programador da Holanda, pois iniciou essa atividade em 1952 com o EDSAC em Cambridge, e ninguém nunca questionou esse título. Naqueles primeiros anos, ele recorda que invejava seus colegas que trabalhavam com hardware porque sabiam exatamente o que estavam fazendo, enquanto que ele, programador, não era sequer reconhecido como profissional. Quando se casou, quis declarar sua profissão como “programador” nos registros, mas a prefeitura não permitiu e, assim, acabou constando como “físico teórico” em seu documento de casamento.

Em relação à evolução da computação nos anos 1950 e 1960, ele dizia que os mecanismos de interrupção introduzidos nas máquinas mais modernas da época tinham criado comportamentos imprevisíveis. Além disso, comentava que quando os primeiros computadores foram construídos, programação era um problema menor, mas no final dos anos 1960 já havia se tornado extremamente difícil, considerando a capacidade das novas máquinas. A coisa teria piorado muito com o surgimento dos computadores de 3<sup>a</sup> geração, como o IBM 360. Dijkstra lamentava que, com o lançamento de tais máquinas, embora os programadores esperassem que suas especificações fossem claras

e sistemáticas, o que se observou foi o contrário: estas eram complexas e obscuras, o que tornava os programadores as mais infelizes criaturas da face da Terra.

Os anos 1970 ficaram conhecidos para a engenharia de software como os anos da crise. Já a década de 1980 foi o período das “balas de prata”, na qual vários metodologistas apresentaram soluções para o problema de produção de software que alegadamente resolveriam todas as dificuldades. Apenas na virada do século XXI chegou-se à era da maturidade em que se percebeu que diferentes tipos de sistema exigem diferentes tipos de processos, não havendo assim uma “receita de bolo” definitiva, embora muito se tenha evoluído hoje nessa área.

## 7.32 Microprocessador – 1968

Com a invenção dos circuitos integrados em 1960, logo várias empresas começaram a imaginar a possibilidade de integrar em uma única peça todos os circuitos necessários para uma unidade central de processamento (CPU) de um computador. Tal circuito seria o que hoje chamamos de microprocessador: um computador completo em um único chip de silício. Isso foi conquistado por várias dessas empresas de forma independente a partir de 1968.

Oficialmente, a invenção do microprocessador é atribuída a Marcian E. “Ted” Hoff (Estados Unidos, 1937). Logo após terminar seu doutorado, Hoff resolveu trabalhar na indústria. Ele conseguiu uma entrevista de emprego com Robert Noyce, o inventor do chip de silício, que saiu da Fairchild para ajudar a fundar uma *startup* chamada Intel. Noyce perguntou o que o entrevistado imaginava que seria a nova geração de chips de silício e Hoff respondeu “memória”. Como era exatamente o que Noyce esperava ouvir, ele foi contratado e tornou-se o empregado número 12 da Intel.

Hoff foi o primeiro a perceber que com a nova tecnologia de circuitos de silício da Intel e uma arquitetura simplificada seria possível juntar em um único chip os vários circuitos necessários para construir um

microprocessador. O projeto originou-se da demanda de uma fabricante de calculadoras japonesa, a Busicom, por um conjunto de circuitos de alta performance. Hoff propôs o uso de uma memória estática (ROM) com um programa armazenado, ligada a uma memória dinâmica (RAM) para dados, um processador de entrada e saída (I/O) e um microprocessador de 4 bits. Entretanto, na época ele ainda não tinha o conhecimento necessário para produzir este circuito e o projeto permaneceu como mera ideia até o lançamento do Intel 4004 três anos depois.

Em 1990, porém, Gilbert Hyatt, após um processo de 20 anos contra o escritório de patentes, foi reconhecido como inventor do microprocessador também em 1968, alguns meses antes da invenção da Intel. A patente foi novamente invalidada posteriormente, no entanto.

Praticamente na mesma época, a Texas Instruments também estava criando seu microprocessador, o TMS 1000, que ficou pronto em 1971 e passou a ser comercializado em 1974.

Também em 1968, outra empresa, a Garrett AiResearch, produziu um microprocessador para o controle de voo dos novos caças F-14 Tomcat. O projeto, completado em 1970, foi considerado tão avançado que a Marinha se recusou a permitir sua publicação até 1997, de forma que tornou-se pouquíssimo conhecido.

## 7.33 MULTICS – 1969

A experiência do MIT com compartilhamento de tempo não terminou com o CTSS. Três anos após a criação deste, iniciou-se o projeto de um novo sistema operacional comercial com compartilhamento de tempo, o MULTICS (Multiplexed Information and Computing Service).

O projeto foi liderado por Fernando J. Corbató (Estados Unidos, 1926) e iniciado em 1964 em uma cooperação entre o MIT, a General Electric e a Bell Labs. O plano foi apresentado em 1965 em uma série de seis artigos. A partir de 1969, o serviço se tornou

operacional no campus do MIT, servindo a milhares de estudantes e professores.

A Bell saiu do projeto em 1969, mas algumas das pessoas que lá trabalharam acabaram depois ajudando a criar o Unix. Em 1970, a GE vendeu seus negócios relacionados a computação, inclusive o MULTICS, para Honeywell, hoje Bull. Conta-se que a IBM não quis participar do projeto, embora convidada pelo MIT, porque estava para lançar o OS/360 e não via uso para as ideias do novo sistema operacional.

Depois que a Bull parou de dar suporte ao MULTICS, seus usuários gradualmente migraram para sistemas Unix. Até onde se tem notícia, a sua última instalação foi encerrada no ano 2000 no Canadá.

O MULTICS incorporava os requisitos para oferecimento de computação como serviço, ou seja, você poderia comprar poder e tempo de computação como compra água encanada ou eletricidade. Tratava-se de um sistema altamente modular e novos módulos podiam ser adicionados conforme a necessidade.

Entre outras características, MULTICS se destacava pelo uso de memória virtual. Múltiplos processadores poderiam compartilhar uma mesma memória física. Ele implementava várias linguagens de programação: PL/I, BCPL, BASIC, APL, FORTRAN, LISP, SNOBOL, C, COBOL, ALGOL 68 e Pascal. Além disso, programas em uma das linguagens poderiam chamar rotinas escritas em outras.

MULTICS implementou o primeiro sistema comercial de banco de dados relacional, o MULTICS Relational Data Store (MRDS). Além disso, foi o primeiro sistema operacional a receber a certificação de segurança nível B2 pelo National Computer Security Center do governo americano, fato que é celebrado no botão mostrado na [Figura 7.26](#).



**Figura 7.26: Botão celebrando a certificação de segurança B2 obtida pelo MULTICS.<sup>33</sup>**

Além disso, ele foi inteiramente implementado em uma linguagem de alto nível, a PL/I, da IBM. Até então, sistemas operacionais eram implementados em linguagem de máquina; implementá-los em linguagem de alto nível era considerado muito arriscado, pois o código resultante poderia não ser eficiente. A realidade, porém, mostrou que os compiladores eram capazes de gerar código muito mais eficiente e em tempo muito mais curto do que programadores humanos com linguagem de máquina poderiam.

### **7.34 ARPANET – 1969**

A ARPANET pode ser considerada, sem sombra de dúvida, como a precursora mais antiga da atual Internet. A motivação para sua criação com frequência é confundida com a criação da comutação de pacotes por Davies e Baran e, por esse motivo, muitas vezes afirma-se que a ARPANET foi criada como uma rede de computadores para resistir a um ataque nuclear. Porém, segundo seus criadores, isso não é verdade; a motivação para criar uma rede deste tipo estava no fato de que os meios de comunicação na época eram tão precários que as conexões nem precisavam de ataques nucleares para falhar – faziam isso espontaneamente.

Inclusive conta-se que a primeira mensagem transmitida pela ARPANET foi “LO”. Um operador estava tentando transmitir a mensagem “LOGIN”, mas após digitar a segunda letra a rede caiu devido a um erro de programação e só foi restabelecida mais de uma hora depois. Assim, a história registrou “LO” como a primeira mensagem transmitida pela ARPANET e, consequentemente, pela Internet.

ARPANET ou Advanced Research Projects Agency Network foi uma rede de comutação de pacotes criada pelo Departamento de Defesa norte-americano. Seu projeto foi inspirado por Joseph C. R. Licklider (Estados Unidos, 1915-1990). Curiosamente, em um memorando de 1963 ele batizou a rede como “Intergalactic Computer Network”:<sup>34</sup> ela seria o principal e essencial meio de comunicação entre governos, instituições e indivíduos.

Ainda em 1963, ele convenceu Ivan Sutherland e Bob Taylor de que o conceito era importante. Taylor especificamente tinha três terminais de computador em seu escritório, que não se comunicavam entre si. Ele percebeu que seria melhor ter um único terminal que o conectasse a qualquer computador que quisesse.

Em 1968, Taylor havia preparado um plano completo para a rede de computadores e a licitação para construí-la foi vencida pela BBN Technologies (originalmente Bolt, Beranek and Newman) em abril de 1969. A rede, proposta por Taylor e rapidamente construída, era composta por pequenos computadores chamados de IMP (Interface Message Processor), equivalentes aos atuais roteadores. O primeiro destes aparelhos, instalado na Universidade da Califórnia em Los Angeles (UCLA), é mostrado na [Figura 7.27](#). Esses computadores funcionavam como passagens (*gateways*) conectando recursos locais à rede. Inicialmente, as conexões eram formadas por linhas telefônicas dedicadas com taxas de comunicação inicial de 56 mil bits por segundo (56 kb/s). O sistema todo foi instalado em nove meses.

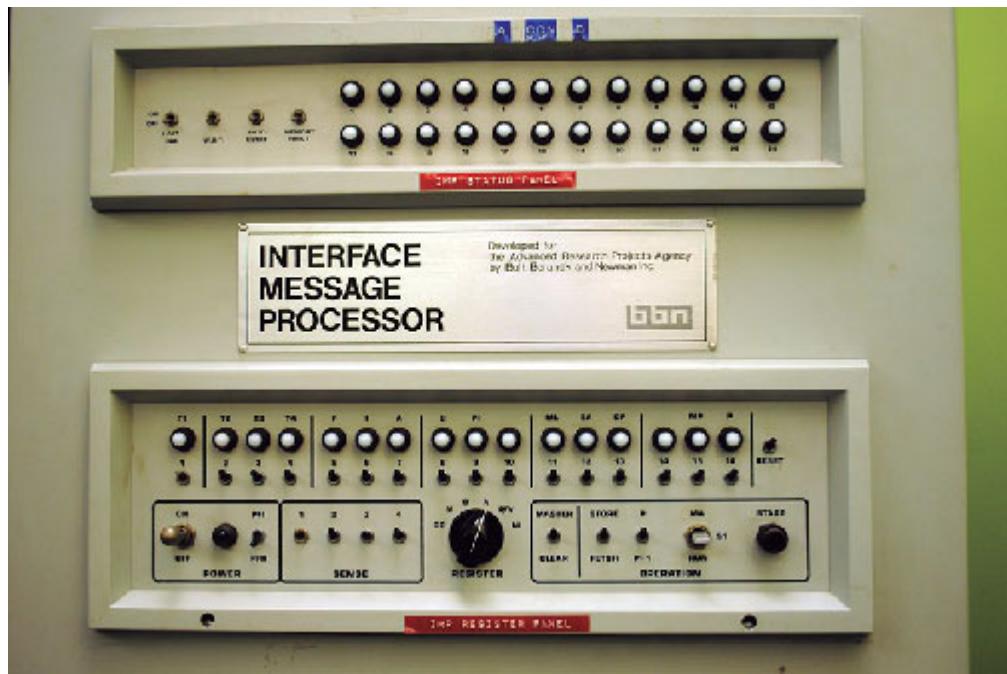


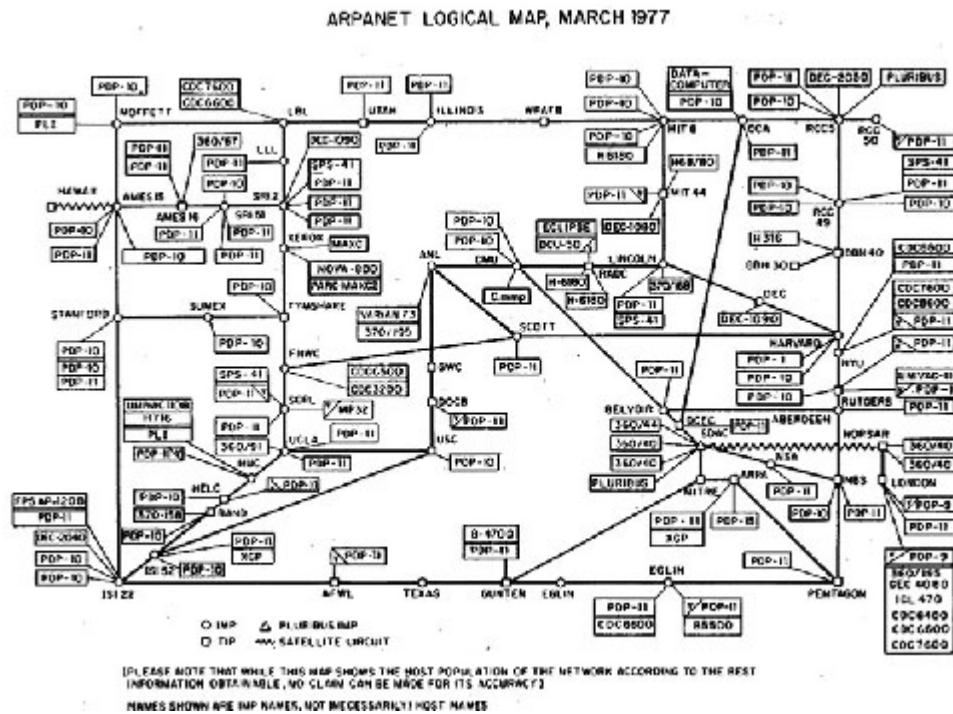
Figura 7.27: O IMP da UCLA.<sup>35</sup>

Inicialmente, a ARPANET consistia de apenas quatro IMPs:

- UCLA (University of California at Los Angeles) com um computador SDS Sigma 7.
- SRI (Stanford Research Institute) com um SDS 940.
- UCSB (University of California at Santa Barbara) com um IBM 360/75.
- University of Utah, com um DEC PDP-10.

A primeira mensagem, “LO”, foi enviada por um estudante da UCLA chamado Charley Kline no dia 29 de outubro de 1969. O primeiro link permanente foi estabelecido em 21 de novembro entre a UCLA e o SRI. Em dezembro do mesmo ano os quatro IMP já estavam todos online.

A partir desse embrião, a rede se expandiu incorporando novas universidades, centros de pesquisa e órgãos do governo. A [Figura 7.28](#) mostra um esquema de como estava a ARPANET em 1977, oito anos após sua criação.



**Figura 7.28: Esquema da ARPANET por volta de 1977.**<sup>36</sup>

Em 1973 ela se liga à Noruega, o primeiro país além dos Estados Unidos a integrar a rede. Em 1975 foi declarada operacional e seu controle foi assumido pela DISA, que a reestruturou separando os nodos militares na MILNET (Military Network). A partir de 1983, o TCP/IP tornou-se o principal protocolo de comunicação da ARPANET e ela própria virou uma subrede da nova Internet. Mas isso é história para mais adiante. A ARPANET foi oficialmente aposentada em 1990, com direito a réquiem (missa fúnebre).

## **7.35 Arkay CT-650, o Computador de Clipe de Papel – 1969**

Irving Becker inspirou-se em um livro de 1967 para construir um modelo de computador pessoal de propósitos educacionais. O livro em questão era *How to Build a Working Digital Computer*<sup>37</sup> de E. Alcosser, J. P. Phillips e A. M. Wolk<sup>38</sup>; ele ensinava como construir partes operacionais de computador usando materiais normalmente

disponíveis em casa, como parafusos, lâminas metálicas e cliques de papel. Daí o CT-650, produzido por Becker, ser também conhecido como o “Computador de Clipe de Papel”.



Figura 7.29: Arkay CT-650.<sup>39</sup>

Mais do que isso, o CT (Computer Trainer) tinha uma unidade de entrada e saída para números decimais, uma unidade de aritmética, uma unidade de controle, memória e um tambor magnético usado para armazenar programas. Além disso, usava uma linguagem de programação flexível e fácil de aprender.

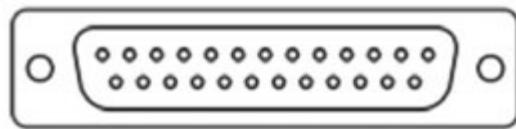
O computador também ficou conhecido como Arkay CT-650, visto que a empresa de Becker se chamava Arkay. O nome vinha da pronúncia inglesa para a sigla “RK”, ou “Radio Kit”, que foi o seu principal produto de vendas.

O CT-650 vendeu poucas unidades, a um preço de cerca de 1.000 dólares. Entre outras características, tinha uma versão destinada a deficientes visuais. Nessa versão, as saídas da máquina eram representadas em pequenas lâmpadas que se aqueciam ligeiramente, permitindo a uma pessoa cega “ler” a saída a partir do toque dos dedos.

## 7.36 RS-232C – 1969

À medida que computadores começaram a ser conectados a periféricos, tornou-se necessário padronizar os plugues de conexão

de forma a evitar que toda a vez fios tivessem que ser soldados, cortados ou mesmo que transformadores de tensão tivessem que ser usados. O padrão de conexão que perdurou dos anos 1960 até bem recentemente foi originado em 1962 na EIA (Electronic Industries Association – hoje Electronic Industries Alliance), uma organização de comércio e padronização. Eles projetaram conectores padronizados para interligar terminais com teletipos. O modelo em questão foi chamado de RS-232 e em 1969 uma nova versão, conhecida como RS-232C ([Figura 7.30](#)), tornou-se padrão para interconexão de computadores e seus periféricos – a qual só foi suplantada a partir dos anos 1990, com a invenção das portas USB.



**Figura 7.30: Conector DB-25 conforme descrito no padrão RS-232.<sup>40</sup>**

O padrão foi usado em situações que iam muito além da especificação original, que acabou ficando incompleta em alguns casos. Assim, ao longo do tempo, problemas de interpretação do padrão acabaram levando a incompatibilidades que precisavam ser resolvidas.

O modelo definia vários tipos de conectores com números de pinos variados. Ele estabelecia o tamanho e posicionamento de todos pinos, bem como a função de cada um e o tipo de corrente ou sinal que deveria usar.

Em comparação com o USB, as interfaces RS-232 são mais lentas e difíceis de implementar, mas elas permitem conexões com cabos mais longos. Assim, embora hoje computadores pessoais praticamente não usem mais essa tecnologia, ela ainda é importante em aplicações especiais.

## 7.37 Impressora Laser – 1969

Antes de falar na invenção da impressora laser precisamos voltar no tempo até 1938, quando foi inventado o processo de fotocópia. Chester Carlson (Estados Unidos, 1906-1968) criou nesse ano um processo que ele chamou de “Xerografia”, que em grego significa “escrita a seco”. Ele conseguiu uma patente pela invenção em 1939, mas somente em 1947 convenceu uma empresa a produzir a máquina. Essa empresa chamava-se Haloid, mas mais tarde mudou o próprio nome em função do processo que a tornou uma gigante no cenário mundial: Xerox.

A Xerox, portanto, fabricava fotocopiadoras e detinha a patente dessa invenção. Em 1967, um de seus jovens pesquisadores, Gary K. Starkweather (Estados Unidos, 1938), teve uma ideia inusitada: “E se em vez de simplesmente copiar um original, nós ligássemos a fotocopiadora em um computador que poderia *produzir* o original?” Com essa ideia nascia a impressora laser.

Apesar de não encontrar encorajamento por parte de seus superiores, ele insistiu no projeto e em 1969 construiu o primeiro protótipo a partir de uma fotocopiadora modificada, com um raio laser que balançava de um lado para o outro de um papel preso a um tambor giratório. Um computador era usado para ligar e desligar o raio laser, produzindo dessa forma impressões em partes do papel, as quais seriam reconhecidas como letras e desenhos. Segundo consta, o hardware ficou pronto em uma ou duas semanas, mas o software levou três meses para ser finalizado.

Em 1970, quando a Xerox fundou o centro de pesquisas PARC, Starkweather foi trabalhar lá e o projeto da impressora laser acabou tornando-se uma das mais importantes contribuições do grupo. A primeira impressora operacional, a SLOT (Scanned Laser Output Terminal), ficou pronta já em 1971.

Após a produção do software para geração de caracteres, o sistema completo ficou conhecido como EARS (Ethernet, Alto, Research Character Generator, Scanned Laser Output Terminal), que em

seguida tornou-se a primeira impressora laser comercialmente disponível: a Xerox 9700, lançada no mercado em 1977 e mostrada na [Figura 7.31](#).



**Figura 7.31: Xerox 9700.<sup>41</sup>**

O princípio de funcionamento da impressora laser consiste na produção de imagens em um cilindro eletrostático. A fotocondutividade desse cilindro faz com que as áreas não expostas à luz do laser atraiam uma tinta em pó muito fina, chamada “toner”. O cilindro transfere essa tinta ao papel através de contato direto e um processo de aquecimento conhecido como “acabamento” faz a fixação da tinta no papel. A [Figura 7.32](#) mostra esquematicamente como é o procedimento.

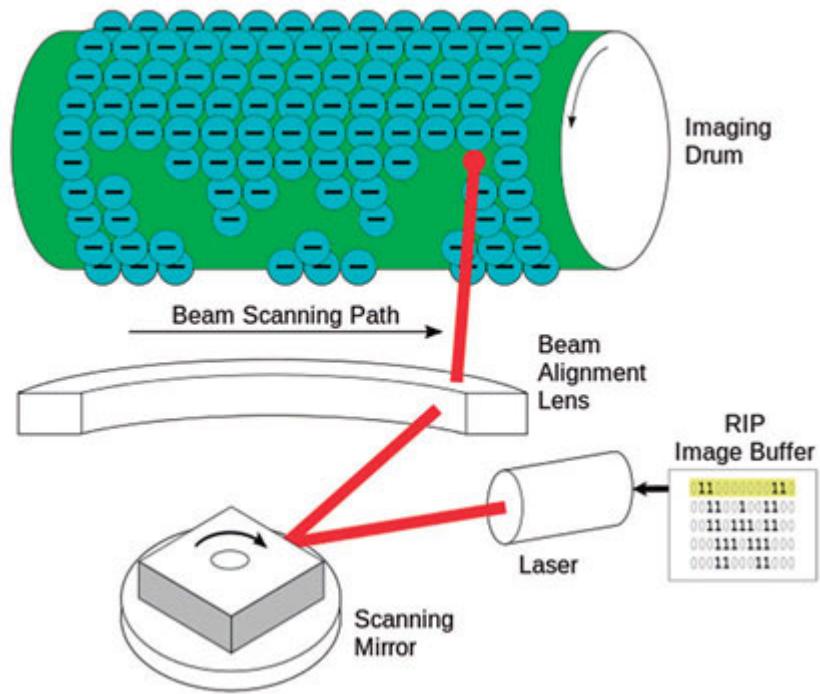


Figura 7.32: Processo de impressão a laser.<sup>42</sup>

A Xerox, infelizmente, não percebeu que a principal fonte de renda desse negócio não estava nas impressoras, mas no toner e no papel especial. Assim, ela quase foi expulsa do mercado pela HP, que passou a produzir impressoras muito mais baratas e lucrar com a venda dos insumos. A HP LaserJet, lançada em 1984 a um custo bastante acessível, tornou-se praticamente sinônimo de impressora laser.

## 7.38 Apollo Guidance Computer – 1969

Em 1969 o ser humano chegou à Lua e junto com ele foi um computador chamado AGC, ou Apollo Guidance Computer ([Figura 7.33](#)).

O AGC tinha palavra de 16 bits dos quais um era usado para verificar paridade. A maior parte do software armazenado nele, 36 kB, estava em memória estática (ROM), que só podia ser lida. Apenas uma pequena memória, 2 kB, que podia ser lida e escrita

(RAM) estava disponível. O AGC foi um computador baseado em circuitos integrados desenvolvido pelo Laboratório de Instrumentação do MIT.

Nas naves do projeto Apollo, havia dois AGCs: um no módulo de comando e outro no módulo lunar. Ambos eram responsáveis pela navegação das naves.



Figura 7.33: Apollo Guidance Computer.<sup>43</sup>

O AGC foi projetado para responder a interrupções de hardware, ou seja, de tempos em tempos o processamento podia ser interrompido para executar alguma ação; depois, o processamento era retomado. Havia interrupções para atualizar o display do usuário, interrupções para indicar falhas de hardware ou alarmes, para indicar que o usuário pressionou uma tecla, para atualizar o relógio de tempo real e para indicar que uma palavra de 16 bits foi carregada.

O software para o AGC foi programado em Assembly. Havia um sistema operacional simples chamado Exec que permitia multiprogramação, ou seja, vários programas (até oito) rodarem ao mesmo tempo. O sistema funcionava assim: de tempos em tempos um programa que estivesse rodando tinha que devolver o controle da CPU ao Exec para que ele verificasse se um programa de prioridade mais alta estava esperando para rodar. O programa com

a mais baixa prioridade era o Dummy, que não fazia nada além de apagar uma luz verde no painel. Quando Dummy estava ativo, isso significava que o computador não tinha outras coisas a fazer e a lâmpada ficava apagada. Porém, quando algum processo de maior prioridade tomava conta da CPU, a lâmpada acendia.

O projeto do software para o AGC foi liderado por Margareth Hamilton (Estados Unidos, 1936), diretora da Divisão de Engenharia de Software do Laboratório de Instrumentação do MIT. Ela tinha experiência como programadora no PDP-1 no projeto MAC de Marvin Minsky e trabalhou também no projeto SAGE, para o qual escreveu o software que procurava por aeronaves inimigas.

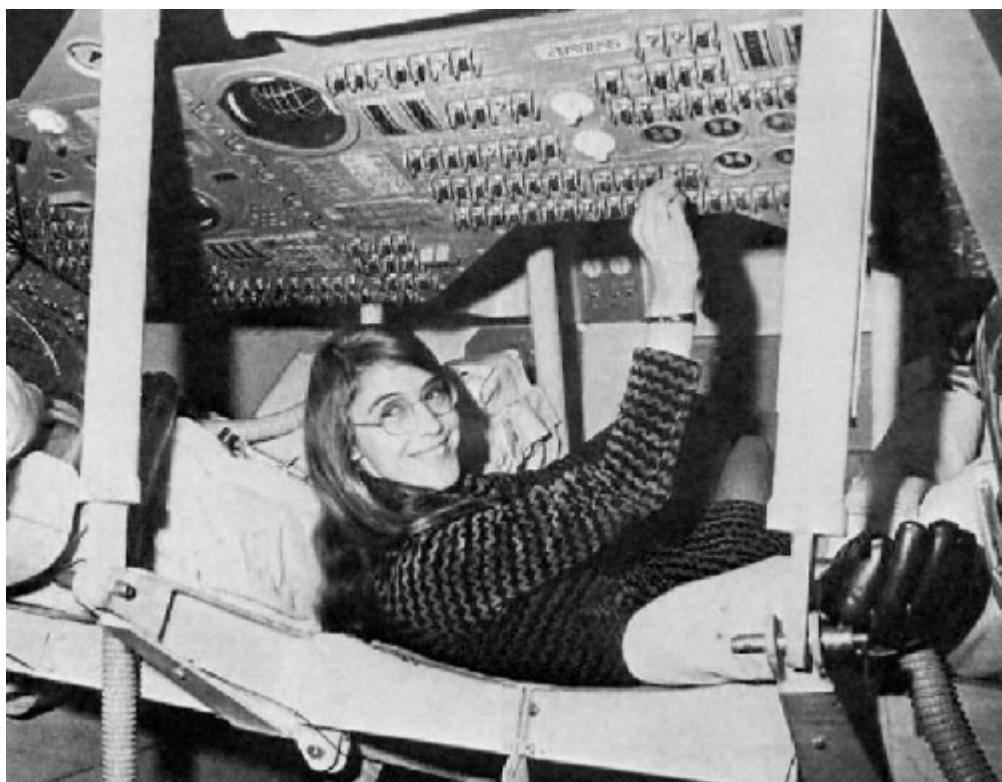


Figura 7.34: Margareth Hamilton no programa Apollo.<sup>44</sup>

O trabalho dela no software de controle do AGC evitou que o primeiro pouso na Lua (Apollo 11) fosse abortado nos instantes finais. Como ela mesma relembra em uma carta para a revista

*Datamation* em 1971: devido a um erro no manual de verificação, um interruptor do radar de encontro foi posicionado no local errado. Isso fez com que ele enviasse sinais errôneos ao computador. O resultado foi que o computador estava sendo solicitado a realizar todas as suas funções normais enquanto recebia uma carga extra de dados espúrios que usavam 15% de seu tempo de processador. O software no computador era suficientemente inteligente para reconhecer que estava sendo solicitado além de sua capacidade, então enviou um alarme que significava para o astronauta: “Estou sobrecarregado com mais tarefas do que posso realizar agora e vou manter apenas as mais importantes”, ou seja, as necessárias para a aterrissagem. Na verdade, o computador estava programado para fazer mais do que simplesmente reconhecer situações de erro. Um conjunto completo de programas de recuperação foi incorporado a ele. A ação do software, neste caso, foi eliminar os processos de prioridade baixa e reestabelecer os mais importantes. Se o computador não tivesse reconhecido este problema e tomado a ação apropriada, a missão Apollo 11 provavelmente não teria sucedido em nos poupar na Lua.

Em 1986, Hamilton recebeu o prêmio Ada Lovelace da Associação das Mulheres na Computação, dentre outras distinções. De fato, ela fez muito mais do que simplesmente programar um computador: Hamilton introduziu conceitos de tolerância a falhas, robustez, garantia de qualidade, detecção de erros, recuperação automática de sistemas e muitos outros que a tornaram uma das mais influentes engenheiras de software da história.

## 7.39 Até Aqui...

Os anos 1960 foram particularmente férteis para a computação. Como vimos, várias ideias que hoje são fundamentais surgiram nesse período: circuito integrado, compartilhamento de tempo, mouse, computação gráfica, comutação de pacotes, tela sensível ao toque, realidade virtual e aumentada, notebooks, microprocessador,

impressora laser e mesmo a rede de computadores, inicialmente ARPANET.

Em termos de linguagens de programação, vimos que muitos exemplos importantes surgiram, tais como COBOL, BASIC, MUMPS, Simula 67 e Forth. Knuth inicia nessa década seu tratado épico sobre a arte de programar computadores, em um tempo em que isso nem era reconhecido como profissão. No final dos anos 1960, Margareth Hamilton cria e aperfeiçoa conceitos de engenharia de software que hoje são fundamentais.

Em termos de equipamentos, merece destaque a série PDP, da DEC, que eram computadores interativos – diferentes, portanto, das máquinas de processamento de dados que até então eram dominantes. Com computadores interativos foi possível iniciar toda uma nova cultura de informática, com videogames, comunicação entre usuários em tempo real e mesmo uma iniciativa na área de inteligência artificial, que foi um marco quando ELIZA pela primeira vez simulou um humano conversando com outros humanos.

As ideias do projeto Dynabook da Xerox rapidamente começaram a evoluir e nos anos 1970, como veremos, a evolução do hardware especialmente motivada pela miniaturização dos circuitos integrados possibilitou o início da era dos computadores pessoais.

---

<sup>1</sup> “Replica IC” by Florian Schäffer - Own work. Licensed under CC BY-SA 4.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Replica\\_IC.png#/media/File:Replica\\_IC.png](https://commons.wikimedia.org/wiki/File:Replica_IC.png#/media/File:Replica_IC.png)

<sup>2</sup> Imagem de domínio público. Disponível em: [https://en.wikipedia.org/wiki/File:COBOL\\_Report\\_Apr60.djvu](https://en.wikipedia.org/wiki/File:COBOL_Report_Apr60.djvu).

<sup>3</sup> “PDP-1” by Matthew Hutchinson - <http://www.flickr.com/photos/hiddenloop/307119987/>. Licensed under CC BY 2.0

- via Wikimedia Commons.* Disponível em:  
<https://commons.wikimedia.org/wiki/File:PDP-1.jpg#/media/File:PDP-1.jpg>
- <sup>4</sup> “Spacewar!-PDP-1-20070512” by Joi Ito from Inbamura, Japan - Spacewar running on PDP-1. Licensed under CC BY 2.0 via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:Spacewar!-PDP-1-20070512.jpg#/media/File:Spacewar!-PDP-1-20070512.jpg>
- <sup>5</sup> Domínio Público. Cortesia de Valdemar W. Setzer. Disponível em: <http://www.aeitaonline.com.br/wiki/index.php?title=Arquivo:T63-Setzer-com-Zezinho.jpg>
- <sup>6</sup> “Titan computer” by University of Cambridge - [http://www.cl.cam.ac.uk/Relics/archive\\_photos.html](http://www.cl.cam.ac.uk/Relics/archive_photos.html). Licensed under CC BY 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Titan\\_computer.jpg#/media/File:Titan\\_computer.jpg](https://commons.wikimedia.org/wiki/File:Titan_computer.jpg#/media/File:Titan_computer.jpg)
- <sup>7</sup> “SRI Computer Mouse” by SRI International - SRI International. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:SRI\\_Computer\\_Mouse.jpg#/media/File:SRI\\_Computer\\_Mouse.jpg](https://commons.wikimedia.org/wiki/File:SRI_Computer_Mouse.jpg#/media/File:SRI_Computer_Mouse.jpg)
- <sup>8</sup> “Telefunken Rollkugel RKS 100-86” by Marcin Wichary - Flickr: Unique mouse from Telefunken, pt. 2. Licensed under CC BY 2.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Telefunken\\_Rollkugel\\_RKS\\_100-86.jpg#/media/File:Telefunken\\_Rollkugel\\_RKS\\_100-86.jpg](https://commons.wikimedia.org/wiki/File:Telefunken_Rollkugel_RKS_100-86.jpg#/media/File:Telefunken_Rollkugel_RKS_100-86.jpg)
- <sup>9</sup> Sigla para “Army-Navy / Fixed Special eQuipment”.
- <sup>10</sup> “SAGE computer room” by United States Air Force - h Licensed under Public Domain via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:SAGE\\_computer\\_room.jpg#/media/File:SAGE\\_computer\\_room.jpg](https://commons.wikimedia.org/wiki/File:SAGE_computer_room.jpg#/media/File:SAGE_computer_room.jpg)
- <sup>11</sup> “SketchpadDissertation-Fig1-2” by Scanned by Kerry Rodden from original photograph by Ivan Sutherland - Electronic edition of Sutherland’s Sketchpad dissertation, edited by Blackwell & Rodden. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:SketchpadDissertation-Fig1-2.tif#/media/File:SketchpadDissertation-Fig1-2.tif>
- <sup>12</sup> Disponível em: [http://www.tiobe.com/index.php/tiobe\\_index](http://www.tiobe.com/index.php/tiobe_index)
- <sup>13</sup> “Electronic Memory” by Steve Jurvetson from Menlo Park, USA - 64-bit Chip. Licensed under CC BY 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Electronic\\_Memory.jpg#/media/File:Electronic\\_Memory.jpg](https://commons.wikimedia.org/wiki/File:Electronic_Memory.jpg#/media/File:Electronic_Memory.jpg)

<sup>14</sup> “360-91-panel”. Licensed under Public Domain via Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:360-91-panel.jpg#/media/File:360-91-panel.jpg>

<sup>15</sup> “CPT-internet-packetswitching” by Computer-blue.svg: OpenClipartderivative work: Pluke (talk) - Computer-blue.svg. Licensed under CC0 via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:CPT-internet-packetswitching.svg#/media/File:CPT-internet-packetswitching.svg>

<sup>16</sup> “CDC 6600.jc” by Jitze Couperus - Flickr: Supercomputer - The Beginnings. Licenced under CC BY 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:CDC\\_6600.jc.jpg#/media/File:CDC\\_6600.jc.jpg](https://commons.wikimedia.org/wiki/File:CDC_6600.jc.jpg#/media/File:CDC_6600.jc.jpg)

<sup>17</sup> Tradução: “Tela sensível ao toque – um novo dispositivo de entrada e saída para computadores.”

<sup>18</sup> “CERN-Stumpe Capacitance Touchscreen” by Maximilien Brice - Disponível em: <http://www.flickr.com/photos/50410282@N06/4626709556/>. Licensed under CC BY 1.0 via Commons. [https://commons.wikimedia.org/wiki/File:CERN-Stumpe\\_Capacitance\\_Touchscreen.jpg#/media/File:CERN-Stumpe\\_Capacitance\\_Touchscreen.jpg](https://commons.wikimedia.org/wiki/File:CERN-Stumpe_Capacitance_Touchscreen.jpg#/media/File:CERN-Stumpe_Capacitance_Touchscreen.jpg)

<sup>19</sup> Tradução: “A gestação de uma criança leva nove meses, não importa a quantas mulheres se atribua a tarefa.”

<sup>20</sup> Tradução: *O mito do homem-mês*.

<sup>21</sup> “Logo Quadrado” by Leonardo Carneiro de Araujo - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Logo\\_Q\\_uadrado.png#/media/File:Logo\\_Q\\_uadrado.png](https://commons.wikimedia.org/wiki/File:Logo_Q_uadrado.png#/media/File:Logo_Q_uadrado.png)

<sup>22</sup> “Logo estrela5” by Leonardo Araujo - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Logo\\_estrela5.png#/media/File:Logo\\_estrela5.png](https://commons.wikimedia.org/wiki/File:Logo_estrela5.png#/media/File:Logo_estrela5.png)

<sup>23</sup> © FORTH, Inc. Usada com permissão. Fonte: [www.forth.com](http://www.forth.com).

<sup>24</sup> © FORTH, Inc. Usada com permissão. Fonte: [www.forth.com](http://www.forth.com).

<sup>25</sup> Tradução: Pequena máquina de computação eletrônica.

<sup>26</sup> Tradução: Grande máquina de computação eletrônica.

<sup>27</sup> By Sergey Vakulenko – Disponível em: <http://ramlamyammambam.livejournal.com/110695.html>, with the agreement from the author (on the same page, in the post comments)., CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=7914366>.

<sup>28</sup> Tradução: Espada de Dâmocles.

- <sup>29</sup> “Dot matrix example text” by Fourohfour - *Own work. Licensed under CC BY-SA 2.5 via Wikimedia Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Dot\\_matrix\\_example\\_text.png#/media/File:Dot\\_matrix\\_example\\_text.png](https://commons.wikimedia.org/wiki/File:Dot_matrix_example_text.png#/media/File:Dot_matrix_example_text.png)
- <sup>30</sup> “HP 9100A Calculator 1968 front” by Swtpc6800 en:User:Swtpc6800 Michael Holley - *Own work. Licenced under Public Domain via Wikimedia Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:HP\\_9100A\\_Calculator\\_1968\\_front.jpg#/media/File:HP\\_9100A\\_Calculator\\_1968\\_front.jpg](https://commons.wikimedia.org/wiki/File:HP_9100A_Calculator_1968_front.jpg#/media/File:HP_9100A_Calculator_1968_front.jpg)
- <sup>31</sup> “Alan Kay and the prototype of Dynabook, pt. 5 (3010032738)” by Marcin Wichary from San Francisco, U.S.A. - *Alan Kay and the prototype of Dynabook, pt. 5. Licenced under CC BY 2.0 via Wikimedia Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Alan\\_Kay\\_and\\_the\\_prototype\\_of\\_Dynabook\\_pt.\\_5\\_\(3010032738\).jpg#/media/File:Alan\\_Kay\\_and\\_the\\_prototype\\_of\\_Dynabook,\\_pt.\\_5\\_\(3010032738\).jpg](https://commons.wikimedia.org/wiki/File:Alan_Kay_and_the_prototype_of_Dynabook_pt._5_(3010032738).jpg#/media/File:Alan_Kay_and_the_prototype_of_Dynabook,_pt._5_(3010032738).jpg)
- <sup>32</sup> © Hector Garcia-Molina. Cortesia. Disponível em: <http://www-cs-faculty.stanford.edu/~knuth/taocp.html>
- <sup>33</sup> © Tom Van Vleck. Cortesia. Disponível em: <http://www.multicians.org/multics-fer.html>
- <sup>34</sup> Tradução: Rede de computadores intergaláctica.
- <sup>35</sup> “Interface Message Processor Front Panel” by FastLizard4 - *Own work. Licenced under CC BY-SA 3.0 via Wikimedia Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Interface\\_Message\\_Processor\\_Front\\_Panel.jpg#/media/File:Interface\\_Message\\_Processor\\_Front\\_Panel.jpg](https://commons.wikimedia.org/wiki/File:Interface_Message_Processor_Front_Panel.jpg#/media/File:Interface_Message_Processor_Front_Panel.jpg)
- <sup>36</sup> “Arpanet logical map, march 1977” by ARPANET - The Computer History Museum ([1]), en:File:Arpnet-map-march-1977.png. Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Arpanet\\_logical\\_map,\\_march\\_1977.png#/media/File:Arpanet\\_logical\\_map,\\_march\\_1977.png](https://commons.wikimedia.org/wiki/File:Arpanet_logical_map,_march_1977.png#/media/File:Arpanet_logical_map,_march_1977.png)
- <sup>37</sup> Tradução: *Como construir um computador digital que funciona.*
- <sup>38</sup> Disponível em: [https://archive.org/stream/howtobuildaworkingdigitalcomputer\\_jun67/HowToBuildAWorkingDigitalComputer\\_Jun67#page/n7/mode/2up](https://archive.org/stream/howtobuildaworkingdigitalcomputer_jun67/HowToBuildAWorkingDigitalComputer_Jun67#page/n7/mode/2up)
- <sup>39</sup> vintagecomputer.net. Cortesia. Disponível em: <http://www.vintagecomputer.net/pictures/2014/SystemSource/>
- <sup>40</sup> “DB-25” by Abisys - *DSubminiatures.svg + self-made. Licenced under CC BY-SA 3.0 via Wikimedia Commons.* Disponível em: <https://commons.wikimedia.org/wiki/File:DB-25.svg#/media/File:DB-25.svg>

<sup>41</sup> Digibarn Computer Museum e Xerox Corp. Cortesia. Disponível em:  
<http://www.digibarn.com/collections/printers/xerox-9700/>

<sup>42</sup> “Laser printer-Writing-es” by Toniperis - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em:  
[https://commons.wikimedia.org/wiki/File:Laser\\_printer-Writing-es.svg#/media/File:Laser\\_printer-Writing-es.svg](https://commons.wikimedia.org/wiki/File:Laser_printer-Writing-es.svg#/media/File:Laser_printer-Writing-es.svg)

<sup>43</sup> “Agc view” by The original uploader was Grabert at German Wikipedia - Transferred from de.wikipedia to Commons by henristosch.. Licensed under Public Domain via Commons. Disponível em:  
[https://commons.wikimedia.org/wiki/File:Agc\\_view.jpg#/media/File:Agc\\_view.jpg](https://commons.wikimedia.org/wiki/File:Agc_view.jpg#/media/File:Agc_view.jpg)

<sup>44</sup> “Margaret Hamilton in action” by NASA - Disponível em:  
[http://www.nasa.gov/50th/50th\\_magazine/scientists.html](http://www.nasa.gov/50th/50th_magazine/scientists.html). Licensed under Public Domain via Commons.  
[https://commons.wikimedia.org/wiki/File:Margaret\\_Hamilton\\_in\\_action.jpg#/media/File:Margaret\\_Hamilton\\_in\\_action.jpg](https://commons.wikimedia.org/wiki/File:Margaret_Hamilton_in_action.jpg#/media/File:Margaret_Hamilton_in_action.jpg)

## **PARTE VIII**

### **O Caminho para o Computador Pessoal**

A década de 1970 foi para a computação uma década de ramificação: de um lado, o desenvolvimento de supercomputadores cada vez mais poderosos e, de outro, o surgimento do computador de uso pessoal. Tudo isso foi possibilitado principalmente pela evolução do hardware. A década de 1970 viu a consolidação do circuito integrado, mas mais do que isso, a criação do microprocessador, ou seja, um circuito integrado que sozinho implementa um computador completo.

Em relação aos computadores pessoais, apesar dos kits disponibilizados na década de 1950 e 1960, com relativamente pouca aceitação fora do circuito dos hobistas, os anos 1970 viram o surgimento dos primeiros microcomputadores, como Kenbak, Xerox Alto, Altair, Apple I e Compucolor. Especialmente o ano de 1977 é considerado fundamental para a história da computação pessoal porque foi quando foi lançada a “Trindade de 77”, ou seja, três computadores de grande sucesso para o público geral: o Apple II, o Tandy TRS-80 e o Commodore PET.

Em termos de software, o final da década viu o surgimento do processador de texto WordStar e da planilha VisiCalc, os quais foram os grandes responsáveis por colocar os microcomputadores também nas empresas, e não apenas nos lares.

No Brasil, essa década viu o aparecimento da indústria nacional de computadores, com a criação do Pato Feio em 1972, a partir do qual os engenheiros obtiveram experiência para criar a primeira empresa construtora de computadores nacionais em 1974, a COBRA.

Ainda nessa década fértil de ideias, foram criadas várias linguagens de programação tais como Pascal, Smalltalk, C e Prolog. No final dos anos 1970 a computação era uma área em franco desenvolvimento, e começava a estar presente não apenas em grandes corporações e governo, mas nos lares, escolas e pequenas empresas – e muito ainda estava por vir.

## 8.1 Memória Intel 1103 – 1970

Como vimos anteriormente, a nova memória DRAM foi inventada por Robert H. Dennard (Estados Unidos, 1932) da IBM em 1966. Porém, só em 1970 ela se tornou um produto comercial, produzido pela Intel e conhecido como o circuito integrado Intel 1103.

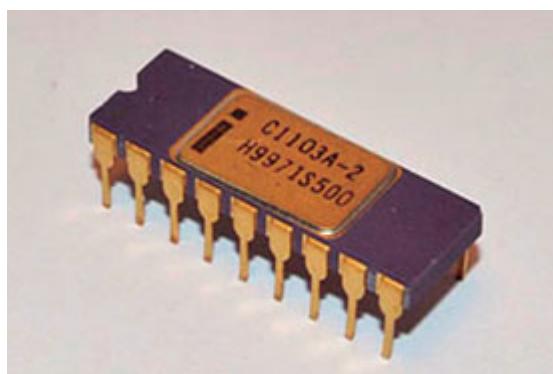


Figura 8.1: Intel 1103.<sup>1</sup>

Em 1970 a Intel estava pouco interessada em patentear invenções. Ela queria desesperadamente colocar produtos no mercado e começar a coletar lucros. No ano anterior, William Regitz, da Honeywell, havia procurado várias empresas de hardware para o projeto de produção de um chip de memória DRAM que ele inventara. A Intel viu nesse projeto uma possibilidade interessante

para um produto novo e resolveu investir nele. A proposta de Regitz seria de um chip com 512 bits de memória dinâmica, mas os engenheiros da Intel concluíram que seria viável fazer o chip com 1.024 bits de memória, e assim o fizeram.

O circuito foi desenhado por Joel Karp da Intel. O projeto produziu algumas provas de conceito e um artigo científico que descrevia o i1103. Ele não foi fácil, pois muitas dificuldades técnicas surgiram. Em outubro de 1970, entretanto, o circuito tornou-se suficientemente confiável para ser comercializado.

Esse circuito foi o responsável pelo fim das memórias de núcleo magnético, pois sua produção era muito mais simples. Em vez de passar fios minúsculos através de pequenos anéis, um circuito 1103 podia ser produzido por um processo semelhante ao da fotografia.

## 8.2 Pascal – 1970

Entre 1968 e 1970, o professor Nicklaus Wirth (Suíça, 1934) projetou uma linguagem de programação inspirada em ALGOL, a qual era fortemente direcionada para o ensino de programação de computadores usando as recém-inventadas técnicas de programação estruturada. A linguagem se chamou “Pascal” em homenagem a Blaise Pascal, inventor da calculadora Pascalina.

A programação estruturada aboliu o uso do GOTO como comando. Os programadores não deveriam mais alterar o fluxo de execução de um programa do ponto A para um ponto B arbitrário. Em vez disso, eles teriam que ser organizados exclusivamente com estruturas de seleção e repetição com início e fim bem delimitados e sem “saídas de emergência”.

Pascal foi um sucesso, especialmente nas universidades, onde, até a virada do século XX, era dominante como linguagem para ensinar programação. Wirth já havia proposto duas outras que não ficaram tão conhecidas: Euler e ALGOL W. Depois de Pascal, ele ainda criou Modula-2 e Oberon, que chegaram a ser usadas, mas não tão intensamente quanto Pascal.

A implementação de compiladores leves para Pascal nos primeiros microcomputadores, como os da Apple, também ajudou a torná-la bastante popular.

Ao contrário das linguagens anteriores, Pascal usava o conceito de tipagem forte, que é aplicado para evitar que um programador consiga, por exemplo, somar quilômetros com minutos. Se um programador declarar, digamos:

```
VAR K: INTEGER; T: INTEGER;
```

Ele poderá somar K com T sem problema algum, mesmo se a primeira variável representar distância e a segunda representar tempo. Porém, usando a tipagem forte de Pascal, o programador iria declarar algo como:

```
TYPE DISTANCIA, TEMPO = INTEGER;
```

```
VAR K: DISTANCIA; T: TEMPO;
```

Neste caso, uma tentativa de somar K com T fará com que o programa apresente um erro quando for compilado. Como se trata de tipos diferentes, mesmo ambos sendo baseados em números inteiros, essa operação é considerada ilegal.

Em 1976, Wirth publica um livro que teria muita influência na adoção de programação estruturada: *Algorithms + Data Structures = Programs*, no qual usa uma linguagem que ele chamou de Tiny Pascal.

Anos mais tarde, Pascal foi duramente criticada por Brian Kernighan (Canadá, 1947), que defendia o uso da linguagem C, criada em 1972. Entretanto, praticamente todas as críticas de Kernighan foram resolvidas pelas mais modernas implementações de Pascal baseadas na versão Extended Pascal. Uma das mais importantes críticas era o fato de que a versão original não permitia declarar variáveis textuais (*string*) nem vetores (*array*) de tamanho variável. Isso forçava os programadores a declarar essas estruturas com um tamanho muito maior do que iriam realmente usar. Por exemplo, para trabalhar com uma lista de números que poderia conter uma quantidade variável deles, o programador teria que estimar um

número limite e declarar o *array* com este tamanho. Se considerasse, por exemplo, que a lista não teria mais do que 100 ou 200 itens ele precisaria declará-la, por exemplo, com 500 posições, por segurança, pois se por algum motivo tentasse armazenar um elemento a mais na lista depois que ela atingisse o tamanho máximo, poderia haver problemas de execução e até o travamento da máquina. Assim, essas 500 posições de memória seriam em boa parte desperdiçadas, e isso era um custo que não se podia pagar, pois memórias eram muito caras. Porém, conforme visto, as versões atuais de Pascal não sofrem mais desse problema.

Atualmente, as suas versões mais populares são Object-Pascal, usada no ambiente Delphi, e Free Pascal, usada no ambiente Lazarus.

### **8.3 Imlac PDS-1, o Computador Gráfico – 1970**

Uma pequena empresa americana chamada Imlac produziu em 1970 um computador com monitor gráfico vetorial que viria a ser muito popular nessa década, o PDS-1. Ele era considerado barato para a época, apenas o preço de quatro fuscas, mas com a mesma funcionalidade de um IBM 2250 que custava 30 vezes mais.

Ele era composto por um minicomputador de 16 bits e memória de núcleo magnético. O principal dispositivo que o tornava popular era o monitor CRT de 1024x1024 pontos. Ele podia ser operado como o Sketchpad de Sutherland, com uma caneta luminosa.

A diferença entre esse modelo de gráficos vetoriais em relação a outros deve-se ao fato de que os monitores usualmente comportavam-se como televisores CRT. Em um televisor o feixe de elétrons move-se horizontalmente e verticalmente varrendo linha por linha da tela. Porém, o monitor vetorial não funciona assim. Se uma reta dessesse ser desenhada no ponto A até o ponto B, o feixe de elétrons seria simplesmente dirigido do ponto A para o ponto B, sem passar por nenhum outro ponto da tela fora da reta. Assim, o

monitor efetivamente desenhava figuras baseadas em linhas (figuras de arame), obtendo uma ótima qualidade gráfica.

Hoje esse tipo de monitor já é obsoleto, mas na época ele foi muito importante porque exigia muito menos memória de vídeo, visto que apenas as coordenadas dos pontos inicial e final das retas precisavam ser armazenadas e não uma matriz completa de 1024x1024 pontos, como seria o caso nos monitores tradicionais baseados em rastreio.

O PDS-1 redesenhou sua tela 40 vezes por segundo. Dessa forma, o olho humano seria incapaz de notar que o feixe luminoso se movia ou que a tela piscava. Porém, se mais do que 800 polegadas de linhas ou 1.200 caracteres fossem desenhados, a tela começava a piscar porque neste caso o feixe demorava demais para fazer o desenho todo, e o atraso já seria perceptível pelo olho humano.

O computador tinha rotinas específicas para desenhar cada letra, número ou símbolo. Inclusive fontes diferentes poderiam ser implementadas.

De fato, apenas os pontos do início e fim de cada linha eram armazenados. Os pontos intermediários em uma linha eram calculados a partir do algoritmo Bresenham, inventado em 1962, que permitia calcular pontos intermediários de uma reta entre dois pontos apenas com somas e subtrações, sem ter que recorrer a divisões e multiplicações, que seriam muito mais lentas nos computadores.

A programação do PDS-1 era feita em seu Assembly, o qual não era compatível com nenhuma outra máquina. Apesar disso, várias aplicações foram desenvolvidas, como editores de texto, por exemplo. Consta que o famoso livro *Gödel, Escher, Bach: An Eternal Golden Braid*<sup>2</sup> publicado em 1979 por Douglas Hofstadter (Estados Unidos, 1945) teria sido escrito em um destes editores. O videogame *Spacewar!* também teve uma versão implementada no PDS-1, para dois jogadores. O jogo *Frogger* (que tinha um sapinho que tentava atravessar uma rua movimentada) foi criado no PDS-1.

Outro jogo que ficou famoso foi *Maze War*, que podia ser jogado por até oito jogadores em computadores diferentes. Diz-se que no início da ARPANET a maior parte do tráfego na rede era devido às batalhas de *Maze War* entre estudantes do MIT e de Stanford.

## 8.4 Bancos de Dados Relacionais – 1970

Embora existam muitos modelos diferentes para estruturação de dados em arquivos, o mais universalmente utilizado hoje é o banco de dados relacional. Neste modelo, todos os dados são organizados em tabelas com colunas representando propriedades e linhas representando registros. Assim, uma tabela com dados sobre pessoas pode ter como colunas, por exemplo, nome, número do documento, endereço e telefone. As linhas seriam representações de diferentes pessoas:

Nome	Número do documento	Endereço	Telefone
João	54765754	Rua Sem Fim, 12	555-7383
Maria	23565723	Rua Norte, 27	555-1567
Pedro	23657622	Rua A, 220	555-9875

O termo “banco de dados relacional” foi inventado por Edgar F. Codd (Reino Unido, 1923-2003) em seu famoso artigo “A Relational Model of Data for Large Shared Data Banks”.<sup>3</sup> Neste, ele considera que o modelo relacional é superior aos modelos hierárquicos ou em rede, usados na época para estruturar dados especialmente pela simplicidade com que as tabelas relacionais poderiam ser tratadas. Foi criado para esse tipo de tabelas o chamado “cálculo relacional”, que permite dar um tratamento matemático muito eficiente a sistemas desse gênero.

Em outras publicações, Codd define 13 regras, numeradas de 0 a 12, com os requisitos que um sistema deve ter para que seja considerado relacional:

- Regra 0 – regra fundamental: qualquer sistema para ser considerado relacional deve ser capaz de gerenciar seus dados

totalmente através das capacidades relacionais.

- Regra 1 – regra da informação: toda informação é representada da mesma forma: como valores em tabelas.
- Regra 2 – regra do acesso garantido: toda informação atômica (*datum*) é acessível através do nome da tabela, chave primária e nome da coluna.
- Regra 3 – tratamento sistemático dos valores nulos: valores nulos são suportados e representam ausência de informação.
- Regra 4 – catálogo online dinâmico baseado no modelo relacional: a descrição do banco de dados é representada no nível lógico exatamente da mesma forma que quaisquer outros dados.
- Regra 5 – regra da sublinguagem de dados comprehensível: um sistema relacional pode suportar várias linguagens, mas pelo menos uma delas deve suportar todos os itens abaixo:
  - Definição de dados.
  - Definição de visões.
  - Manipulação de dados.
  - Restrições de integridade.
  - Autorização.
  - Limites de transação (*begin*, *commit* e *rollback*).
- Regra 6 – regra da atualização da visão: todas as visões que sejam teoricamente atualizáveis são atualizadas pelo sistema.
- Regra 7 – inserção, remoção e atualização de alto nível: a capacidade de gerenciar uma relação simples ou derivada com operações únicas não se aplica apenas à recuperação de informação, mas também à sua inserção, remoção ou alteração.
- Regra 8 – independência física de dados: os programas que usam o banco de dados relacional não devem ser afetados no caso de mudanças na forma física de representação dos dados.

- Regra 9 – independência lógica de dados: os programas que usam o banco de dados relacional não devem ser afetados no caso de mudanças que alteram a disposição dos dados, mas não afetam a informação neles representada (por exemplo, uma coluna sendo movida de uma para outra posição).
- Regra 10 – independência de integridade: restrições de integridade que dizem respeito a um banco de dados relacional devem poder ser representadas no próprio banco e não apenas na linguagem de programação.
- Regra 11 – independência de distribuição: um sistema gerenciador de banco de dados relacional deve ter independência de distribuição.
- Regra 12 – regra de não subversão: se o banco de dados tiver uma linguagem de baixo nível, ela não pode violar ou subverter as restrições de integridade estabelecidas pela linguagem de alto nível.

Codd foi um dos programadores do IBM SSEC, como já vimos. Depois de algumas mudanças, ele acabou trabalhando em um dos laboratórios de pesquisa da IBM na Califórnia. Suas ideias sobre bancos de dados relacionais foram consideradas em um primeiro momento, na melhor das hipóteses, como curiosidade científica, e, na pior, como ameaças aos produtos da IBM.

Porém, seus conceitos foram aceitos e adotados por empresas que estavam se formando na época, como a Oracle, Ingres, Informix e Sybase. Essas empresas perceberam que a linguagem de consulta para bancos relacionais produziria programas muito mais simples e curtos. Uma consulta que em CODASYL poderia necessitar de cinco páginas podia ser escrita em uma única linha na linguagem de consulta de um banco de dados relacional.

Quando a concorrência se tornou relevante, a IBM resolveu implementar seu primeiro produto relacional, o Sistema R, em 1973. Codd não participou do projeto. Um dos resultados deste foi a criação da linguagem SQL (Structured Query Language). O primeiro

produto relacional da IBM só chegou ao mercado em 1981: o SQL/DS.

Codd recebeu o Prêmio Turing em 1981, mas nunca ficou rico com sua invenção.

## 8.5 PDP-11 – 1970

Em 1970 a DEC lançou um minicomputador que seria um ícone dos anos 1970: o PDP-11. Ele foi possivelmente um dos maiores sucessos em termos de vendas, que chegaram a mais de 600 mil unidades entre 1970 e o início dos anos 1990. Seu preço relativamente baixo possivelmente deveu-se ao fato de que foi idealizado de maneira que pudesse ser construído por funcionários sem alta especialização. A [Figura 8.2](#) mostra o painel frontal original do modelo PDP-11/20.



Figura 8.2: Painel frontal original do PDP-11/20.<sup>4</sup>

Consta que as usinas nucleares robotizadas com equipamentos General Electric devem continuar a utilizar os confiáveis PDP-11 até pelo menos 2050. Há muito código escrito em Assembly em uso nessas usinas e nenhuma razão para substituí-lo no momento.

O PDP-11 era um minicomputador de 16 bits com oito registradores, sendo seis de propósito geral, mais o contador de programa

(registrador que contém o endereço de memória da instrução sendo executada pelo computador) e o registrador do topo da pilha (registrador que contém o endereço de memória da pilha de dados usada para armazenar dados de forma semelhante à que vimos quando explicamos a linguagem Forth).

O PDP-11 foi o primeiro computador a rodar o sistema operacional AT&T Unix e BSD Unix. Sua arquitetura influenciou outras importantes, como a Intel x86 e a Motorola 68000. Seu sistema operacional influenciou alguns que vieram depois, como o CP/M e o MS-DOS.

Estima-se que dezenas de milhares de clones do PDP-11 tenham sido construídos nos países por trás da cortina de ferro (Leste Europeu) durante a Guerra Fria.

Larry McGowan, um dos programadores contratados para trabalhar no projeto do PDP-11 relembra que o setor de marketing da DEC queria que os comentários da linguagem de programação fossem iniciados com uma barra “/”. Com bastante senso de humor, disse que aceitava e sugeriu então que a divisão entre dois números passasse a ser representada por ponto e vírgula “;” para não haver confusão.

## 8.6 Modelo Cascata – 1970

Em 1970 a indústria de software já contava com um número de projetos de grande porte suficiente para que houvesse preocupação em definir um processo efetivo para concepção e produção de programas em larga escala. Por essa época, especialmente considerando-se o aprendizado de Brooks com o OS/360, já se sabia que processos adotados por outros tipos de indústria não funcionavam muito bem com a indústria de software.

Porém, o modelo de processo de desenvolvimento mais usado durante todo o século XX surgiu de forma um tanto irônica. Em 1970, Winston Royce (Estados Unidos, 1929-1995) publicou um artigo no qual examinava algumas opções de processos de

desenvolvimento de software. Nesse artigo, em uma figura ele apresenta um modelo baseado em fases bem definidas:

1. Requisitos de sistema.
2. Requisitos de software.
3. Análise.
4. Design de programa.
5. Codificação.
6. Teste.
7. Operação.

Logo em seguida, ele apresenta várias críticas a esse modelo, visto que ele dificilmente funcionaria na prática já que requisitos mudam com frequência e não é possível basear, na maioria das vezes, um projeto em um documento de requisitos que vai ser modificado. Além disso, aspectos de análise e design podem ter que ser revistos durante a codificação e deixar para fazer os testes apenas no final, depois que todo o código está produzido, é praticamente suicídio.

Porém, ao longo dos anos 1970 muitos desenvolvedores tomaram o modelo apresentado por Royce como referência positiva sobre como o software deveria ser desenvolvido. Possivelmente, muitos usaram a figura sem observar as críticas de Royce sobre o modelo. Ela lembrava um pouco uma escada com setas indicando o fluxo descendente (Figura 8.3) e, assim, ele passa a ser conhecido como modelo Cascata (*Waterfall*).

O esquema até tinha vantagens por organizar o processo produtivo em fases bem definidas e bem documentadas, mas também foi provavelmente o mais criticado em toda a história, visto que a exigência da produção de vários documentos ao final de cada fase faz com que seja um modelo pesado e burocrático que exige muita especificação antes que se possa começar a produzir algum código.

Além disso, ele é pouco adaptável a mudanças, coisa que ocorre com frequência em projetos de software: o modelo Cascata não lida bem com mudanças porque exige que todos os documentos

anteriores sejam reescritos desde a fase onde houve a mudança até a fase atual do desenvolvimento.

O modelo acabou tendo muitas variantes que procuravam reduzir suas deficiências, como “cascata com subprojetos”, “cascata com redução de risco”, “cascata orientado a cronograma”, dentre outros. Modernamente, os principais modelos de desenvolvimento de software não são mais baseados em grandes fases como o Cascata, mas em iterações curtas de desenvolvimento nas quais a equipe trabalha desde requisitos até implementação, teste e validação de partes do sistema.

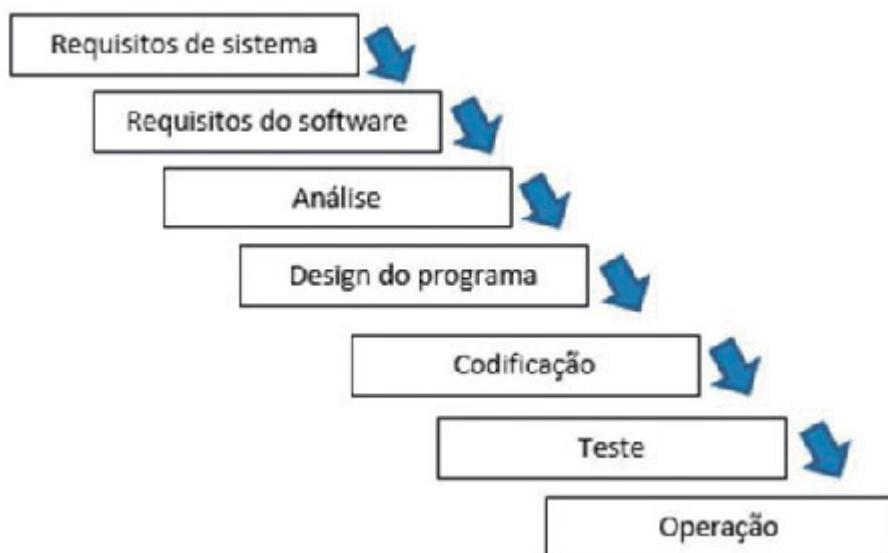


Figura 8.3: Modelo Cascata.

## 8.7 Unix – 1970

Como foi visto anteriormente, o MIT, a Bell Labs e a GE desenvolveram o sistema operacional de tempo real MULTICS na década de 1960, o qual introduziu muitas inovações, mas também muitas dificuldades. Em função do progresso lento, o grupo da Bell se retirou do projeto e começou a desenvolver um novo sistema operacional com as mesmas ideias do MULTICS, mas com um

projeto bem mais simples. Inicialmente, eles trabalharam por conta própria em um PDP-7 que estava “encostado em um canto” na Bell Labs. Porém, depois conseguiram apoio e um computador maior, um PDP-11/20 ([Figura 8.4](#)).



**Figura 8.4: Thompson e Ritchie trabalhando no Unix no PDP-11.<sup>5</sup>**

O principal responsável pelo projeto foi o engenheiro Kenneth Thompson (Estados Unidos, 1943). O nome do novo sistema inicialmente sugerido foi UNICS (Uniplexed Information and Computing System), dado pelo engenheiro Brian Kernigham (Canadá, 1942) como uma brincadeira relacionada ao MULTICS (Multiplexing Information and Computer Services). Mais tarde, foi abreviado para Unix. Também trabalharam no projeto Dennis Ritchie (Estados Unidos, 1941-2011) e Joe Ossanna (Estados Unidos, 1928-1977).

Eles criaram um sistema hierárquico de arquivos que é basicamente o sistema de diretórios usado nos computadores atuais. Ainda em 1971-72 o Unix já contava com um editor de textos. Ele foi

disponibilizado juntamente com seu código fonte, sob licença, a universidades, empresas e órgãos do governo, ficando assim muito popular. Em 1972-73, foi completamente reescrito com a nova linguagem de alto nível C, contrariando a noção de que software básico como sistemas operacionais deveriam ser escritos em linguagem de máquina para serem eficientes. Esse fato tornou o Unix um sistema operacional relativamente independente de máquina, pois apenas pequenas partes do código precisavam ser reescritas quando ele fosse instalado em computadores diferentes.

Hoje em dia, Unix e sistemas derivados dele, como Linux e FreeBSD, são bastante usados, especialmente nos meios acadêmicos. Em função da aparência física dos criadores e incentivadores de Unix, um de seus famosos “credos” afirma: “*No beard, no belly, no guru.*”<sup>6</sup>

## 8.8 INTEL 4004, o Primeiro Microprocessador – 1971

Como vimos anteriormente, o conceito de microprocessador ou CPU em um único chip foi inventado na Intel no final dos anos 1960. Porém, apenas em 1971 o primeiro produto comercial, o Intel 4004, um microprocessador de 4 bits, foi disponibilizado ao mercado. Ele foi anunciado na revista *Electronic News* ([Figura 8.5](#)) neste ano.

O Intel 4004 tinha aproximadamente o tamanho de uma unha humana e o mesmo poder de processamento do ENIAC, construído 25 anos antes. Ele possuía 2.300 transistores e é considerado o precursor da família de processadores Intel x86 até hoje usados. Era capaz de processar entre 60 e 90 mil instruções por segundo, e custava apenas 200 dólares.

O projeto ficou a cargo de Federico Faggin (Itália, 1941), ex-funcionário da Fairchild, trabalhando na Intel. Na mesma família de circuitos havia duas memórias, uma RAM (leitura e escrita), a 4002, com 80 palavras de 4 bits, e uma ROM (apenas leitura), a 4001, com 256 palavras de 8 bits, além de um registrador de

deslocamento, 4003, que era usado para entrada de dados, como por exemplo teclados. Na [Figura 8.6](#), pode-se ver a silhueta de Faggin no lado direito de uma imagem ampliada do 4004.

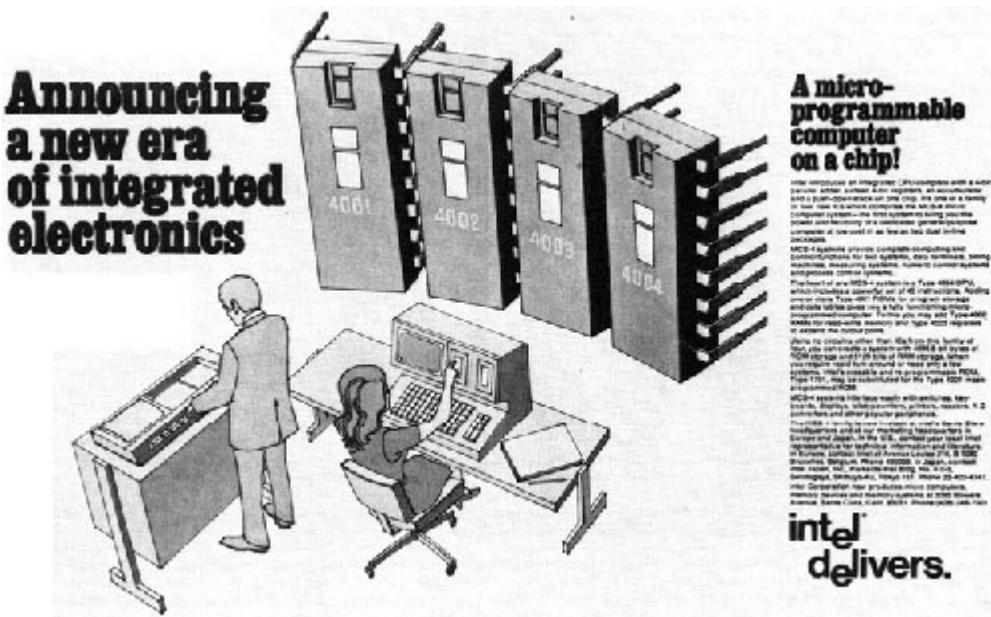
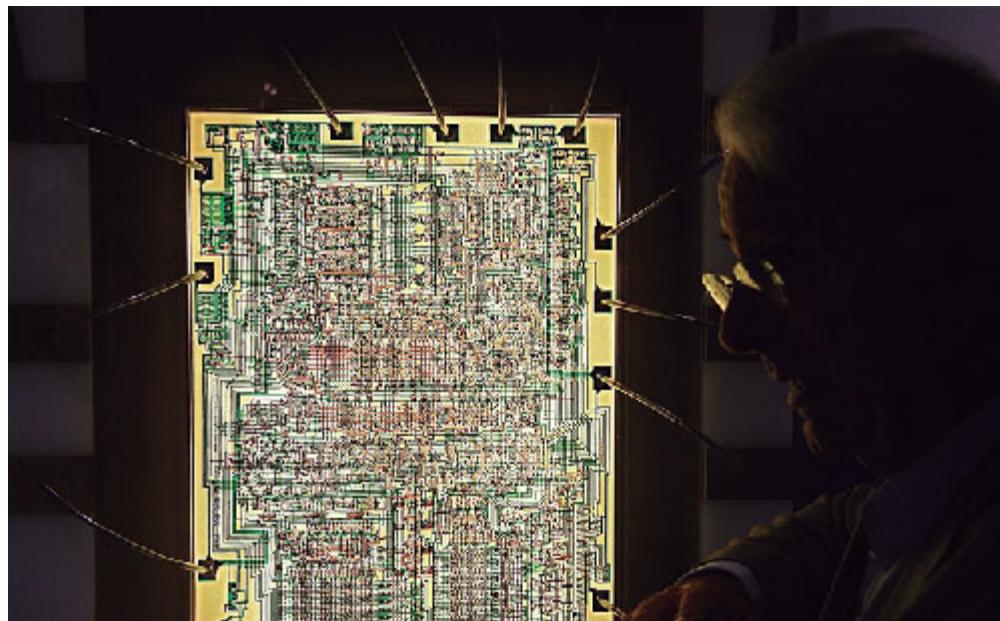


Figura 8.5: Anúncio do Intel 4004 na *Electronic News* em 1971.<sup>7</sup>



**Figura 8.6: Faggin e uma visão ampliada do 4004.<sup>8</sup>**

O primeiro uso comercial do 4004 foi na calculadora eletrônica da japonesa Busicom, a 141-PF. Pouco tempo depois, o 4004 foi sucedido por processadores de 8 bits como o 8008 e o 8080.

## **8.9 E-Mail – 1971**

O correio eletrônico, ou e-mail – como passou a ser conhecido a partir de 1993 –, é uma forma de troca de mensagens entre usuários de computadores. Os primeiros sistemas de troca de mensagens exigiam que dois usuários estivessem online ao mesmo tempo para poderem conversar, mas o procedimento de e-mail é baseado em envio de mensagens que ficam armazenadas até que o receptor resolva lê-las.

Nos anos 1970 o termo “*electronic mail*” era usado indiscriminadamente para qualquer tipo de comunicação eletrônica, inclusive fax. A primeira implementação de um sistema de correio eletrônico da forma como o entendemos hoje ocorreu em 1962 na rede Autodin (Automatic Digital Network) do Departamento de Defesa norte-americano. A Autodin funcionava com 350 terminais em países como Estados Unidos, Inglaterra e Japão. Um programa chamado NOVA roteava as mensagens e operava os circuitos de comunicação. Em 1982, a Autodin foi absorvida pela ARPANET.

Porém, ela era um sistema de troca de mensagens entre terminais de um mesmo computador. A primeira comunicação eletrônica através de uma rede de computadores ocorreu após a introdução da ARPANET, em 1971. Ray Tomlinson (Estados Unidos, 1941) fazia parte de um pequeno grupo de programadores que desenvolvia um sistema de compartilhamento de tempo para o PDP-10, chamado TENEX.

Tomlinson tinha trabalhado com um sistema de envio de mensagens semelhante ao da Autodin, que permitia que diferentes usuários conectados ao mesmo computador por meio de terminais enviassem mensagens de texto uns aos outros. O programa com o

qual ele trabalhou chamava-se SNDMSG (abreviação de *send message*, ou enviar mensagem). O SNDMSG já tinha o conceito de caixa de correio (*mailbox*), que era simplesmente um arquivo com um nome especial, no qual mensagens iam sendo adicionadas. Ele era protegido de forma que os usuários que não fossem o dono da caixa de correio não pudessem ver as mensagens. A única forma de adicionar texto à caixa de correio era usando o SNDMSG.

Ao mesmo tempo, ele estava trabalhando com um programa de transferência de arquivos entre computadores, o CPYNET, que permitia a um usuário transmitir um arquivo de um computador na ARPANET para outro. Ele, então, percebeu que bastaria juntar os dois programas, o CPYNET e o SNDMSG, para criar um sistema que permitiria, pela primeira vez, o envio de mensagens estilo correio através de uma rede de computadores.

A única coisa que faltava era dar ao CPYNET a capacidade de adicionar um texto a um arquivo de caixa de correio remota (em outro computador), o que, segundo Tomlinson, foi muito fácil de fazer.

Restava a ele definir uma forma de identificar as mensagens que deviam ser enviadas para outros computadores. Mensagens locais eram transmitidas para o nome de usuário. Por exemplo SNDMSG PEDRO 123 enviaria a mensagem “123” a uma pessoa cujo nome de usuário era “PEDRO”. Entretanto, se o usuário estivesse em outro computador ele precisaria de uma maneira diferente para representar esse nome, até porque poderia haver outros usuários PEDRO em diferentes computadores, e eles precisariam ser diferenciados.

Tomlinson resolveu então usar o símbolo @, que em inglês é lido como “at” ou “em”. Assim, uma mensagem enviada para PEDRO@MIT seria interpretada como o usuário “PEDRO” no computador identificado por “MIT”.

A primeira mensagem, enviada em 1971 foi literalmente de um computador para outro computador do seu lado, mas que estava

ligado a ele apenas pela Arpanet. A mensagem teria sido “QWERTYUIOP” ou algo similar.

## **8.10 Projeto Gutenberg, a Primeira Biblioteca Digital – 1971**

O Project Gutenberg (PG) é considerado a mais antiga biblioteca digital do mundo. Ele foi criado por Michael S. Hart (Estados Unidos, 1947-2011). A maioria dos textos disponibilizados são livros de domínio público. O primeiro documento digitalizado inserido nela foi a Declaração de Independência dos Estados Unidos.

O projeto iniciou depois que Hart ganhou acesso ilimitado a um computador Xerox Sigma V ligado à ARPANET na Universidade de Illinois e, grato, comprometeu-se a fazer algo de útil em retribuição: no caso, a digitalização e disponibilização até o ano 2000 de uma biblioteca eletrônica com os 10 mil livros mais consultados disponíveis ao público, sem custo.

O projeto foi batizado em homenagem a Johannes Gutenberg (Alemanha, 1398-1468), inventor da impressão com tipos móveis. Até 1989, todos os livros do projeto eram digitados manualmente; então foram introduzidos os scanners e OCR, que facilitaram o trabalho.

Atualmente o projeto, disponível em [www.gutenberg.org](http://www.gutenberg.org), conta com mais de 50 mil títulos de e-books totalmente gratuitos. A digitalização e revisão do material é feito por dezenas de milhares de voluntários e existe inclusive muito material disponível em língua portuguesa.

## **8.11 The Creeper, o Primeiro Vírus – 1971**

“Vírus de computador” é um termo que só foi cunhado em 1983, mas esse tipo de programa já incomodava usuários desde bem antes. O vírus é um tipo de *malware*, ou seja, um programa malicioso que tem a tendência de se auto reproduzir, infectando assim várias máquinas com as quais ele tem contato e

eventualmente (embora não sempre) danificando arquivos nestes computadores.

De fato, já em 1966 John von Neumann publicou um artigo sobre autômatos autoduplicadores, no qual estaria definindo o princípio de funcionamento dos futuros vírus. Ainda antes disso, em 1961, programadores da Bell Labs implementaram um jogo chamado *Darwin* no qual diversos programas competiam entre si por uma maior fatia de tempo no processador compartilhado. Porém, esses programas ainda não se espalhavam como infecções.

Apenas em 1971, com a ARPANET, pode-se dizer que as primeiras infecções tiveram lugar. Bob Thomas Morris, da BBN Technologies, escreveu nesse ano um programa autoduplicador experimental chamado “The Creeper”<sup>9</sup>, que conseguia transmitir e instalar cópias de si próprio através da ARPANET. Uma vez instalada, a cópia exibia a mensagem: “*I am the creeper. Catch me if you can!*”<sup>10</sup> Com esse programa, Thomas queria testar as ideias de von Neumann. O vírus infectava computadores DEC PDP-10 rodando o sistema operacional TENEX. O primeiro antivírus foi então criado para caçar e destruir cópias do The Creeper: seu nome era “The Reaper”.<sup>11</sup>

Outro vírus, criado em 1974, ficou conhecido como Rabbit<sup>12</sup> ou Wabbit,<sup>13</sup> em função da velocidade com que se reproduzia. Uma vez tendo infectado uma máquina, ele começava a criar várias cópias de si próprio as quais eram colocadas em operação. Rapidamente a performance da máquina era reduzida e logo ela simplesmente travava, por não dar conta de rodar tantos processos ao mesmo tempo.

Em 1975 foi criado um jogo popular chamado *Animal*, no qual o computador faz perguntas aos usuários para tentar adivinhar em que animal estão pensando, juntamente com um vírus, o Pervade,<sup>14</sup> que infeccionava vários diretórios das máquinas UNIVAC à medida que o usuário jogava. Esse programa também se espalhava através de fitas compartilhadas. Entretanto, ele foi feito de forma a não

danificar os dados nem as máquinas. Uma atualização do sistema operacional pôs fim à infecção.

Em 1988 um vírus chamado Jerusalém causou uma epidemia mundial. Ele só se tornava ativo nas sextas-feiras 13. Porém, quando isso acontecia ele destruía todos os arquivos executáveis no computador.

O primeiro programa antivírus de propósito geral foi lançado em 1990 pela Symantec: tratava-se do Norton Antivírus, ainda hoje bastante usado. Esse tipo de programa não só detecta e elimina milhares de vírus, mas também evita que computadores sejam infectados, ao fornecer um sistema de proteção em tempo real.

## **8.12 Computador Pessoal Kenbak1 – 1971**

O Kenbak1 é considerado por alguns como o primeiro computador pessoal comercial. Porém, como vimos, ele foi antecedido pelo Simon e outros. Ainda assim, pode ser considerado um dos primeiros computadores pessoais que pelo menos não precisavam ser montados pelo comprador.

Ele foi inventado e construído por John Blankenbaker da Kenbak Corporation em 1970 e 1971 com o intuito de ser simples e barato. Assim, poderia ser vendido a escolas e usado com fins educacionais. No entanto, a empresa não conseguiu convencer quase ninguém de que se podia comprar um computador por apenas 750 dólares e assim as vendas foram baixas. Estima-se que apenas 40 máquinas foram vendidas.

A Kenbak Corporation funcionava na garagem do próprio inventor, na época desempregado, onde ele projetou construiu o computador. Como o projeto foi realizado antes da divulgação da existência de microprocessadores como o Intel 4004, ele acabou sendo montado a partir de 132 circuitos integrados. Como aconteceu com a maioria dos primeiros computadores pessoais, a entrada de dados se dava através de chaves binárias e o resultado do processamento era

visível a partir de pequenas lâmpadas, também representando um número binário.



Figura 8.7: Kenbak1.<sup>15</sup>

A produção acabou em 1973 quando a empresa falhou e foi adquirida por uma companhia especializada em produtos educacionais. O produto foi rebatizado como H5050, mas as vendas continuaram insignificantes.

## 8.13 Disco Flexível – 1971

O disco flexível ou disquete, como ficou conhecido, foi o meio mais popular de compartilhamento de arquivos da era pré-internet e permaneceu como principal mídia para distribuição de programas e dados até ser suplantado pelos CDs óticos nos anos 2000.

O disco flexível foi inventado pela IBM no final dos anos 1960 por David L. Noble, e tornou-se produto comercial em 1971 juntamente

com os dispositivos capazes de ler (e a partir de 1973 também gravar) dados nestes discos.

Os discos flexíveis usavam um material muito semelhante às fitas magnéticas, mas tinham uma grande vantagem em relação a elas: as fitas eram de acesso sequencial e rebobiná-las ou fazê-las avançar para a posição onde estavam os dados costumava ser um processo extremamente demorado; já um disquete poderia ter todas as suas posições alcançadas em frações de segundo com um simples movimento do cabeçote de leitura de uma para outra trilha concêntrica do disco.

O disquete também tinha outra vantagem em relação aos discos rígidos e tambores magnéticos: seu tamanho e peso diminutos. Um disco magnético na época podia pesar alguns quilos enquanto que disquetes eram tão finos e leves que até se contam boatos de que ocasionalmente eles eram grampeados a memorandos ou relatórios, de forma que ficavam arruinados.

A motivação da IBM para criar estes dispositivos veio da necessidade de ter um meio barato (menos de 5 dólares) para enviar atualizações de microcódigo para clientes com máquinas IBM 370.

O primeiro disquete produzido, o Minnow, tinha 8 polegadas (20 cm) e não possuía capa; era realmente um disco plástico flexível. Sua capacidade de armazenamento era de 80 kB (cerca de 80 mil palavras). Mas a poeira e sujeira tornaram-se rapidamente um empecilho ao seu uso e assim, uma capa plástica internamente revestida por tecido que mantinha a superfície limpa foi introduzida antes do produto ser lançado ao mercado.

Em 1972 Alan Shugart, o gerente de produtos de memória de acesso direto da IBM que atribuiu o projeto do disquete para Noble, deixou a IBM e passou a trabalhar na empresa Memorex, que então se tornaria uma grande fabricante deste tipo de mídia. Rapidamente a Memorex começou a produzir um disquete melhorado, com capacidade de 175 kB.

Até 1973 os leitores de disquete não eram capazes de gravar dados, mas neste ano a IBM lançou o modelo IGAR, capaz de ler e gravar até 250 kB de dados em disquetes de 8 polegadas.

Com o surgimento dos microcomputadores em meados dos anos 1970 os disquetes rapidamente se tornaram o meio dominante de memória secundária, visto que até meados dos anos 1980 raramente uma máquina dessas contaria com discos rígidos, cujo custo ainda era bastante alto.

Muitos modelos e tamanhos foram produzidos ao longo dos anos 1970 e 1980, mas dos que foram mais usados, além dos de 8 polegadas, deve-se destacar os de 5½ e 3½ polegadas. O de 5½ foi introduzido em 1976 e tornou-se muito popular nos primeiros computadores pessoais, embora unidades de fitas K7 também fossem usadas. A partir de meados dos anos 1980, eles passaram a ser substituídos pelos modelos de 3½ que eram diferentes dos anteriores, com seu envelope externo bem mais rígido e um dispositivo que deixava exposta a área de leitura/gravação apenas quando o disquete estava dentro do dispositivo que o acessaria, protegendo-o dos danos causados pela poeira. Os três modelos de disquete mais usados na história podem ser vistos na [Figura 8.8](#).

Até hoje a imagem do disquete de 3½ polegadas é o ícone da ação de “salvar” arquivos em muitos sistemas.



Figura 8.8: Disquetes de 8, 5 $\frac{1}{4}$  e 3 $\frac{1}{2}$  polegadas.<sup>16</sup>

## 8.14 Smalltalk – 1971

Smalltalk foi uma linguagem única e influente na história da computação. Enquanto o paradigma de programação estruturada se tornava dominante nos anos 1970 como uma forma mais organizada de desenvolver programas, Smalltalk era gestado no Xerox PARC por Alan Kay (o mesmo que concebeu o Dynabook), Adele Goldberg (Estados Unidos, 1945) e outros, como o embrião do paradigma de programação seguinte, ainda hoje dominante: a programação orientada a objetos.

Em 1966, enquanto trabalhava na Universidade de Utah, Kay travou conhecimento com a tese de Ivan Sutherland sobre o Sketchpad e com a linguagem de programação Simula 67. Kay conta que percebeu uma ligação entre esses dois sistemas, que lhe acometeu como uma espécie de epifania: os objetos do Sketchpad poderiam ser programados como as estruturas de Simula. Juntando as duas ideias, ele concebeu o conceito de orientação a objetos como nós os conhecemos hoje.

Basicamente, a motivação para este novo paradigma está no fato de que a programação estruturada divide as capacidades de um computador em dois grandes domínios: dados e programas, onde

os dados representam a estrutura da informação e os programas são sequências de instruções para modificar ou consultá-los. Essa divisão aparece claramente na linguagem COBOL, que, como vimos, tem partes diferentes do programa para a definição completa de um e outro domínio e também na programação estruturada, na qual estruturas de dados e algoritmos são elementos pertencentes a domínios completamente distintos, embora sua união produza os programas de computador, como Wirth claramente representou no título de seu famoso livro de 1976, *Algorithms + Data Structures = Programs*.

Porém, Kay visualizou uma maneira diferente de lidar com a complexidade dos programas. Ele pensou: porque não dividir o computador em pequenos computadores virtuais e estes em outros cada vez menores até chegar nos computadores mais básicos? Esses computadores virtuais seriam os objetos.

Smalltalk, cuja primeira versão em laboratório foi produzida em 1971 era uma linguagem *sui generis*. Nela, qualquer valor que pudesse ser representado era considerado um “objeto” e podia ser tratado como tal. Um objeto seria capaz de fazer apenas três coisas:

- Manter seu estado interno, ou seja, um conjunto de referências (links) para outros objetos.
- Receber e executar mensagens enviadas por si próprio ou por outros objetos.
- Enviar mensagens a si próprio e a outros objetos.

Ao contrário de outras linguagens de programação onde havia, por exemplo, instruções representando estruturas de controle, em Smalltalk não existia nada disso. Para repetir uma instrução certo número de vezes, por exemplo, devia-se enviar uma mensagem a um número inteiro e passar como argumento um bloco de programa para ser repetido; por exemplo, para fazer o objeto “turtle” desenhar um quadrado, o programa em Smalltalk consistiria em enviar a mensagem “timesRepeat:” ao número 4, passando como argumento o bloco de programa [turtle forward: 10; turnRight: 90]:

4 timesRepeat: [turtle forward: 10; turnRight: 90]

Dentro do bloco, temos o objeto “turtle” que recebe duas mensagens em sequência: “forward:”, que o faz andar 10 passos (o argumento é 10), e “turnRight:” que o faz girar 90 graus à direita (pois o argumento é o objeto 90).

A primeira versão comercial de Smalltalk foi lançada em 1980 e ficou conhecida como Smalltalk 80. Kay teve também influência de Papert e sua linguagem Logo na sua concepção. A tartaruga semelhante à de Logo foi uma constante em quase todas as implementações.

Smalltalk deveria ser muito mais do que uma linguagem de programação: ela seria um ambiente de desenvolvimento mental, semelhante ao Logo. Na verdade, deveria substituir todos os outros programas do computador, incluindo o sistema operacional, sistema de arquivos, a interface gráfica, o sistema de comunicações, enfim tudo que pudesse ser representado em um computador seria um objeto Smalltalk, desde valores booleanos até as próprias janelas da interface gráfica.

Para alcançar essa homogeneidade, Kay propôs que todo objeto deveria ser instância de uma classe. Ele compara esse mecanismo com o “mundo das ideias e mundo dos sentidos” de Platão, o qual comprehende a classe como a concepção ou a ideia, enquanto o objeto seria a realização física ou concreta dessa ideia. Mesmo as classes em Smalltalk também são instâncias de uma classe chamada “Metaclass”, que por sua vez é uma instância de “Class”, que é uma classe, fechando assim a definição recursiva.

Smalltalk criou o conceito de compilação *just in time*.<sup>17</sup> Normalmente, programas são escritos em longos trechos de arquivos de texto e, uma vez completados, são submetidos a um programa compilador, que traduz os comandos em alto nível para comandos em linguagem de máquina. Em Smalltalk, entretanto, cada pequeno trecho de programa, chamado “método”, uma vez concluído e salvo é imediatamente convertido em um código

intermediário, chamado “*byte code*”, que depois pode ser interpretado diretamente pelo computador sem necessidade de tradução para linguagem de máquina.

Essa característica possivelmente foi uma das causas do declínio de Smalltalk a partir dos anos 2000. A sua performance era baixa quando comparada com linguagens híbridas orientadas a objetos nas quais nem todos os valores são objetos, como C++, Java e Object Pascal. Além disso, Smalltalk demorou muito a se tornar compatível com bancos de dados relacionais, visto que a filosofia da linguagem era não ter elementos que não fossem objetos. Porém, os bancos relacionais eram ferramentas extremamente populares e não dar suporte a eles inviabilizou muitos dos possíveis usos da linguagem.

Hoje, linguagens híbridas como Python e Ruby implementam muitos dos bons princípios de programação inspirados em Smalltalk. Mesmo Java se inspirou muito nessa linguagem.

## 8.15 C – 1972

A famosa linguagem C, muito usada hoje em dia, originou-se de uma linguagem chamada B, a qual por sua vez *não* se originou de uma linguagem A. Está certo: não houve a linguagem A. Tudo começou com a linguagem BCPL (Basic Combined Programming Language) que originou B. Vamos ver...

BCPL foi criada por Martin Richards (Reino Unido 1940), na Universidade de Cambridge em 1966. Era uma linguagem estruturada, inicialmente pensada para programação de compiladores.

B, por sua vez, foi implementada como simplificação de BCPL por Ken Thompson (Estados Unidos, 1943) e Dennis Ritchie (Estados Unidos, 1941-2011). Ela foi criada para ser uma linguagem simples, compacta e altamente eficiente. Havia apenas um tipo de dado: a palavra de máquina; ou seja, em um computador de 16 bits o único tipo de dado seria o número binário de 16 bits, e ponto final!

A sucessora de B, chamada de C simplesmente em função da sucessão alfabética, seria bem mais completa. Ela ainda seria uma linguagem extremamente compacta e eficiente, mas possuiria vários tipos de dados e estruturas de controle, coisa que B não tinha.

A criação de C está fortemente ligada à criação do sistema operacional Unix, que foi originalmente implementado em Assembly por Ritchie e Thompson em um PDP-7. Eles pretendiam reescrever uma nova versão de Unix para o PDP-11 em B, mas a falta de tipos de dados e estruturas de controle levou-os a evoluir a linguagem B até chegar à versão inicial de C.

Um compilador foi construído em 1972 e, neste mesmo ano, o sistema operacional Unix foi reescrito na linguagem C. A publicação do livro *The C Programming Language* em 1978 causou uma revolução no mundo da programação, tornando a linguagem altamente popular. Até hoje a linguagem C ou suas variantes, como C++, são fortemente usadas especialmente nos setores de desenvolvimento de software embarcado e em automação industrial, onde código compacto e eficiente é indispensável.

Porém, a sintaxe por vezes criptográfica de C levou a algumas brincadeiras, como por exemplo a dos adeptos da cultura Smalltalk, que, após o lançamento da versão orientada a objetos de C, chamada C++, criaram o ditado: “C++ runs faster, but frequently in the wrong direction.”<sup>18</sup>

Outro fato interessante é a existência do concurso internacional “Obfuscated C Code”<sup>19</sup> que dá prêmios para os mais criativos programadores C que sejam capazes de criar programas praticamente incompreensíveis que realizam coisas notáveis. O programa em C mostrado na [Figura 8.9](#) é a implementação de um simulador de voo em C, o qual venceu o concurso em 1998.

```

#include <math.h>
#include <sys/time.h>
#include <X11/Xlib.h>
#include <X11/Xkeysym.h>
double L ,o ,P
,_dt,T,z,D=1,d,
i[999].E,h= S,I,
J,C,u[999],M,m,O
,n[999],j=32e-3,i-
103,r,t, u,v ,W,S-
74.5,l=221,x=T,26,
a,B,A=32.2,c, F,H;
int N,q, C,y,p,U;
Window z; char f[s];
: GC k; main(){ Display=e-
XOpenDisplay( 0); z=RootWindow(e,0); for (XSetForeground(e,k=XCreateGC (e,z,0,0),BlackPixel(e,0))
: scanf("11451f2f",y +n,w,y, ys)+1; y ++); XSelectInput(e,z= XCreateSimpleWindow(e,z,0,0,400,400,
0,0,WhitePixel(e,0 ),KeyPressMask); for(XMapWindow(e,z)); T-sin(0){ struct timeval G-{ 0,dt*i64}
; K- cos(j); N=ln4; M=- H"; Z=D^K; F=- P; r-E'K; W-cos( O); n-E'W; H-K'T; O+D" "F/ K+d/K'E",_; E-
sin(j); w=B" T-D-E" W; XClearWindow(e,z); t=T'E+ D'B'W; j+=d" "D_ "F'E; P=W"E" B-T'D; for (o+=(I=D'W+E
*T" B,E" d/K "B+v+B/K" F" D" ); p<y; ){ T=p[s]+1; E=c.p[w]; D=n[p].L; E-D'm-B" T-H"E; if(p [n]w[ p]+p[s]
]=@|K <fabs(w-T" r-I" E" D" P) |fabs(D-t" D+z "T-a "E); K)N=le4; else{ q=W/X "4E2+2e2; C- 2E2+4e2/ K
'D; N-1E488 XDrawLine(e ,z,k,M ,U,q,C); N=q; U=u; } ++p; } L+=- " (X't +P" Nm" 1); T-X" X; l+=- " M;
XDrawString(e,z,k ,20,180,f,17); D=v/1" IS; l+=- (B "l-M" r -X" Z" ); for(; XPending(e); u ="-CS!-N){
XEvent z; XNextEvent(e ,z);
++"((N-XLookupKeysym
(&z,xkey,0))-IT?
N-LT? UP-N?& E:&
J:& U: &h); --"(DN
-N? N-DT ?N=-
RT?&u: & W;&h:83
); } m=15^F/l;
c+=(I*M/ 1,1*H
+I" H+u*X" Y" _; H
-A" r+v*X-F" *i+
E-.1+X" 4.9/1,t
=T" m/32-I" T/24
)/S; K=F'M+(h"
104/1-(T+
E" 5" T" E)/3e2
)/S-X" d-8'A;
a=?.63 /1*d;
X+=- ( d" 1-T/S
"(.19" E +a
+.64+3/1e3
)-M" V +A"
Z" ; l +=
K " ; W=d;
sprintf(f,
"Xsd %ld"
"X7d",p,-1
/1.7,(C-9E)+
0"57.3)%6558,(int)1); d+=T" (.45-14/1"
X-a"130-J" .14)"/125e2+F" _"v; P=(T" (s7
*I-m" 52+e" 94 "D-t".38+u".21"E) /1e2+w"
179" v)/2312; select(p=0,0,0,0,8G); v-=(W" F-T" (.63" m-I" .086+i" E" 19-D" 25-..11" u
)/107e2" ; D=cos(o); f=sin(o); } }

```

Figura 8.9: Obfuscated C Code.<sup>20</sup>

Em fevereiro de 2016 a linguagem C aparecia em 2º lugar na tabela TIOBE, ou seja, é a segunda mais popular no mundo. Outras variações, como C++, C# e Objective-C, aparecem respectivamente em 3º, 4º e 14º lugar.

## 8.16 PROLOG – 1972

Um novo paradigma de programação também nasceu no início dos anos 1970, em 1972 para ser mais preciso. Tratava-se da linguagem PROLOG (PROgrammation en LOGique) e, acompanhado a ela, o paradigma de programação em lógica.

Segundo seus autores, Alain Colmerauer (França, 1941) e Philippe Roussel (França, 1945), ela não foi criada para ser uma linguagem de programação, mas para ser um interpretador de linguagem natural, em especial, a língua francesa. O projeto deu origem a uma versão preliminar de PROLOG em 1971 e uma mais definitiva em 1972.

A linguagem inicialmente baseava-se na lógica de predicados, como a criada por Frege, com expressões do tipo  $(\forall x) \text{ subconjunto}(x,x)$ , ou seja, “para todo  $x$ ,  $x$  é subconjunto de si próprio”.

Porém, essa forma de escrita usando todo o potencial da lógica de predicados era por demais difícil de implementar e, em alguns casos, tornava impossível determinar resultados. Eles precisavam de uma linguagem mais previsível. Assim, resolveram desistir de usar uma linguagem lógica pura e completa a favor de um subconjunto mais simples de implementar e mais previsível em termos de resultados: as Cláusulas de Horn. Assim, um programa em PROLOG é uma sequência de definições da forma:

A :- B1, B2, ..., Bn.

Nessa expressão, A e os Bs são fatos, ou seja, afirmações como *gato(tom)*, onde *tom* é um objeto e *gato* um predicado, que aqui funciona como uma asserção, ou afirmação com valor verdade. Afirmar *gato(tom)* significa dizer “*tom* é um gato”.

Uma expressão PROLOG, que define que gatos comem ratos poderia então ser descrita assim:

come(x,y) :- gato(x), rato(y).

Dessa forma, se afirmarmos:

gato(tom).

rato(jerry).

come(tom,?).

A resposta será:

jerry

Por outro lado, se trocarmos a terceira expressão por:

come(?,jerry)

A resposta será:

tom

Ainda, se escrevermos:

come(jerry,?)

O sistema responderá que não sabe, pois nada foi afirmado sobre o que jerry come.

Usando essa linguagem, Colmerauer e Roussel implementaram um sistema de comunicação homem-máquina em 1972, com 610 cláusulas (definições). Segue um exemplo (traduzido) de diálogo entre homem e máquina obtido por eles com este sistema. Inicialmente, apresentam as regras, fatos e questões:

Todo psiquiatra é uma pessoa.

Toda pessoa que ele analisa está doente.

Jacques é um psiquiatra em Marseille.

Jacques é uma pessoa?

Onde está Jacques?

Jacques está doente?

As respostas obtidas do computador foram:

Sim.

Em Marseille.

Eu não sei.

O sistema PROLOG era mais do que uma linguagem, mas um sistema interativo e um banco de dados de fatos lógicos. Ele foi inicialmente implementado em FORTRAN e mais tarde parcialmente

em PROLOG, em um IBM 360. Hoje a linguagem ainda é usada por grupos que trabalham com inteligência artificial, mas tem sido cada vez mais substituída por outras mais específicas. Na tabela de popularidade TIOBE de fevereiro de 2016 ela aparecia em 35º lugar.

## 8.17 Pato Feio – 1972

Dez anos depois do Zezinho, o Brasil construiu outro computador que seria bem mais influente na criação de uma indústria de informática no país. O projeto foi realizado no programa de pós-graduação da Escola Politécnica da Universidade de São Paulo (USP) em 1972. Os professores e alunos do Laboratório de Sistemas Digitais (LSD) resolveram competir para um contrato com a Marinha Brasileira, que queria encomendar a construção de um computador nacional para substituir os modelos Ferranti presentes nas fragatas compradas da Inglaterra. Eles souberam que a Universidade Estadual de Campinas (UNICAMP) também pretendia concorrer e que o seu computador se chamaria “Cisne Branco”, em homenagem ao Hino da Marinha. Dessa maneira, o grupo bastante jovem da USP resolveu fazer uma brincadeira e nomearam seu projeto de “Pato Feio” ou “Patinho Feio” ([Figura 8.10](#)).



**Figura 8.10: Pato Feio.<sup>21</sup>**

O projeto do Pato Feio foi possível porque a USP comprou um computador da IBM disponível para ser desmontado e remontado de forma que os professores e estudantes pudessem entender melhor como ele funcionava. O processo chama-se “engenharia reversa”. A IBM não era muito favorável a que suas máquinas fossem desmontadas, analisadas e reconectadas com outros sistemas desenvolvidos pelos usuários e alegou que não mais se responsabilizaria pela manutenção do caríssimo computador. A USP respondeu que “tudo bem”, os próprios professores e alunos cuidariam da manutenção da máquina.

O Pato Feio foi por muito tempo considerado o primeiro computador brasileiro, pois o Zezinho teve pouca divulgação e o Lourinha, nenhuma. O mais importante, contudo, é que o Pato Feio foi a base de estudos para a criação de toda uma geração de engenheiros capacitados a iniciar a indústria de computadores no Brasil.

O projeto foi coordenado pelo professor Antonio H. Guerra Vieira (Brasil, 1930), que também foi reitor da USP entre 1980 e 1982. O Pato Feio realmente funcionou. Ele era formado por 450 circuitos

integrados e pesava cerca de uma tonelada. Sua memória era de 4 kB, ou seja, 4 mil palavras de 8 bits cada, e a única linguagem de programação era a linguagem de máquina.

Com a experiência adquirida na construção do Pato Feio, a equipe foi chamada pela Marinha para desenvolver um novo modelo, o G-10. Mais tarde, em 1974, esse projeto serviu de base para a criação da empresa Computadores Brasileiros, ou COBRA, a primeira empresa totalmente nacional (estatal) a produzir computadores.

## 8.18 Magnavox Odyssey – 1972

O ano de 1972 viu o lançamento do primeiro console de videogame comercial da história. Era o Odyssey da empresa Magnavox ([Figura 8.11](#)). Um console é uma espécie de computador de propósito especial, neste caso, o de servir para entretenimento através de jogos. Ele podia ser ligado a um aparelho de TV para visualização das imagens e tinha um controle com dois botões, um para mover na vertical e outro na horizontal. O projeto foi de autoria de Ralph H. Baer (Alemanha, 1922-2014).



Figura 8.11: Magnavox Odyssey.<sup>22</sup>

O projeto iniciou em 1966 e o primeiro protótipo ficou pronto em 1968. Além do console, ele era acompanhado por um conjunto de cartões de jogo, semelhantes aos “cartuchos” que passaram a ser usados anos depois nos videogames. Porém, o cartão de jogo do Odyssey não continha componentes eletrônicos, mas apenas uma placa de circuito impresso que formava diferentes conexões entre os pinos de entrada do console, configurando assim diferentes jogos.

O Odyssey era vendido juntamente com vários acessórios que se encontram em jogos de tabuleiro como fichas, dinheiro de mentira, dados etc. Porém, o mais interessante era um conjunto de telas translúcidas que podiam ser colocadas em frente ao tubo de imagem das TVs antigas, a maioria das quais em preto e branco, para dar uma sensação mais realista ao jogo. Assim, você poderia, por exemplo, colocar sobre a tela da TV uma película translúcida que lembrava uma quadra de tênis com grama verde e outros desenhos. O Odyssey apenas mostrava os personagens e a bola se movimentando, visíveis através da tela semitransparente.

Outro periférico opcional era uma arma de luz que podia ser apontada para a TV e disparada. Ela simplesmente analisava a quantidade de luz emitida pelo ponto da tela no qual estava apontada. Assim, era possível, por exemplo, apontar a arma de luz para uma lâmpada acesa e marcar pontos no jogo.

O primeiro console também deu origem ao primeiro processo judicial relacionado a videogames. Ainda em 1972, Nolan Bushnell (Estados Unidos, 1943), que depois fundaria a Atari, viu uma demonstração do jogo de tênis do Odyssey. Meses depois, lançou uma máquina com o game *PONG*, muito semelhante. Ao contrário do Odyssey, que era um console para uso doméstico, o primeiro *PONG* era uma máquina do tipo Arcade, ou seja, um equipamento grande, para uso em locais como bares e galerias de jogos. Para jogar *PONG* era necessário colocar uma moeda na máquina e a Atari faturou muito com sua invenção, o que levou a Magnavox processá-la por quebra de patente.

Apesar disso, consta que somente em 1972 a Magnavox tenha vendido cerca de 100 mil unidades do Odyssey: um tremendo sucesso. Até 1975 mais de 350 mil unidades haviam sido vendidas. Porém, mais tarde a Atari acabou dominando o mercado com seus consoles domésticos de baixo custo e bons gráficos coloridos para a época.

## 8.19 Calculadora HP 9830A – 1972

A calculadora HP 9830A ([Figura 8.12](#)), lançada em 1972, diminuiu a distância entre as calculadoras e computadores porque tinha uma unidade de disco rígido e podia ser programada com a linguagem BASIC.



**Figura 8.12: HP 9830A.<sup>23</sup>**

Outra característica pouco usual dessa calculadora era a possibilidade de avaliar várias expressões ao mesmo tempo. Expressões separadas por vírgulas seriam avaliadas simultaneamente, produzindo o resultado de cada uma no display de 80 caracteres.

A calculadora tinha uma memória bastante grande para a época, com 15 kB expansível para 31 kB.

Modelos anteriores usavam uma linguagem de programação mais simples do que BASIC, chamada HPL. Ela tinha uma característica interessante porque usualmente nas linguagens de programação a instrução de atribuição é escrita como  $X \leftarrow Y+Z$ , indicando que a variável X assume como novo valor o resultado da soma de Y e Z. Porém, em HPL ela é invertida:  $Y+Z \rightarrow X$ . Assim, a leitura é  $Y+Z$  “vai para” X. Em inglês “vai para” é “*goes into*” e, assim, a instrução ficou conhecida como “gazinta”, uma corruptela de “*goes into*”.

## **8.20 Pong – 1972**

A Atari foi fundada em 1972 por Ted Dabney (Estados Unidos, 1943) e Nolan Bushnell. Anos antes, quando Bushnell estudava engenharia elétrica conheceu o *Spacewar!* de Steve Russell no MIT. Já na Universidade de Utah, foi apresentado a Ivan Sutherland e o seu Sketchpad. Bushnell foi, portanto, uma das poucas pessoas de sua época a ter contato tanto com um videogame quanto com as possibilidades gráficas que os computadores poderiam permitir. Para completar, era frequentador assíduo de locais onde podia jogar jogos do tipo Arcade, que na época eram basicamente máquinas mecânicas como Pinball.

Bushnell queria muito fabricar uma máquina eletrônica de jogos, mas os computadores dos anos 1960 eram do tamanho de uma sala e não se prestariam para isso. Apenas no final da década, com a miniaturização crescente dos componentes, é que se tornou viável

pensar em máquinas de jogos de tamanho razoável, ou seja, um pouco menores que uma geladeira.

Assim, em 1971 Bushnell e Dabney, agora trabalhando para a Nutting Associates, projetaram e construíram o primeiro equipamento do tipo, com um jogo chamado *Computer Space*. Apesar do jogo ter ficado abaixo das expectativas da Nutting, eles venderam cerca de 1.500 dessas máquinas, o que lhes permitiu angariar recursos suficientes para pensar em um projeto próprio.

Sua empresa, inicialmente chamada Syzygy Co., foi fundada em 1971. Depois de descobrir que, por incrível que pareça, uma empresa com esse nome já estava registrada, trocaram-no para Atari, Inc. em 1972. O nome vem da expressão japonesa “ataru”, que significa “acertar no alvo”. Bushnell descobriu isso enquanto jogava Go: e “atari” significaria “quase chegando no alvo”, algo como anunciar o “xeque” em uma partida de xadrez.

Dabney foi o responsável por criar o sistema dedicado para os videogames. Eles não poderiam usar os enormes e caríssimos computadores, como o que rodava o *Spacewar!*. Porém, não precisavam de tanto: o necessário mesmo era mover pontos pela tela. Dabney criou um sistema eletrônico que fazia exatamente isso: movia pontos pela tela. E assim nasceu o videogame moderno. E o melhor de tudo era que esses circuitos podiam ser construídos e colocados em pequenos gabinetes, suficientemente adequados para uma sala de jogos Arcade.

Bushnell conheceu o *Tênis para Dois* da *Odyssey* e resolveu fazer algo parecido. Ele contratou um programador chamado Al Alcorn (Estados Unidos, 1948), que desenvolveu então o software para o *Pong*. Ele não foi o primeiro videogame, e sequer foi o primeiro a ser comercializado, pois foi antecedido pelo *Computer Space*. Entretanto, foi o primeiro a ser maciçamente vendido e realmente entrar na cultura popular de sua época.

As diferenças entre o *Pong* e o *Tênis para Dois* eram sutis. No *Pong*, a velocidade da bola aumentava à medida que o jogo

prosseguia. Além disso, quando a bola quicava em posições diferentes da raquete isso podia alterar o ângulo em que era rebatida. Mais ainda: o *Pong* tinha som.

O primeiro protótipo foi colocado em um bar conhecido como “Andy Tapp’s Tavern”. Ele logo começou a chamar a atenção; havia inclusive pessoas que iam ao bar apenas para jogar e sequer tomavam uma bebida. A máquina foi construída com uma TV preto e branco enfiada em um gabinete pintado de amarelo ([Figura 8.13](#)). Para jogar, o usuário devia depositar uma moeda de 25 centavos. Cada jogador movia sua raquete com um botão giratório. Um dia, o dono do bar chamou Alcorn para consertar a máquina, e este descobriu que não havia nada errado com ela, exceto o recipiente das moedas, feito com uma jarra de leite, que estava completamente abarrotado.



Figura 8.13: Uma máquina de jogar *Pong*.<sup>24</sup>

A produção podia começar. Inicialmente, procuraram duas fabricantes para terceirizar o trabalho, mas depois decidiram que seria melhor se eles mesmos fabricassem as máquinas. Para isso,

porém, precisavam investir capital que não tinham. Não era fácil encontrar recurso inicial pelo seguinte motivo: jogos Arcade, como o *Pinball*, eram muito mal vistos pela sociedade da época, que considerava que suas luzes, cores e sons eram uma forma de corromper a juventude (afinal, era o início dos anos 1970). O *Pong* era muito parecido com esse tipo de máquina e poucas empresas queriam associar seu nome a um “corruptor de jovens mentes”. Porém, eles acabaram conseguindo o dinheiro e no final de 1972 começaram a enviar suas máquinas para os quatro cantos do mundo.

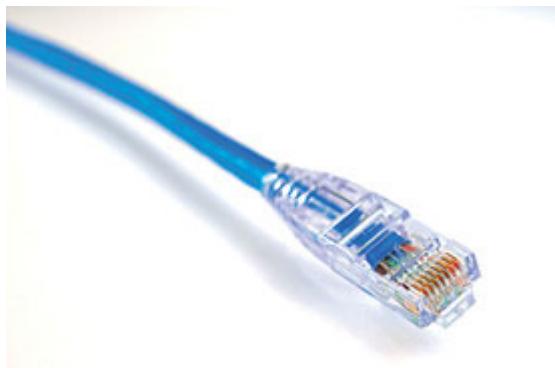
Ela era um sucesso, chegando a faturar até 40 dólares por dia cada uma. As encomendas eram mais do que a empresa suportava: 2.500 em 1973 e 8 mil em 1974. Quando a versão doméstica do *Pong* foi lançada na forma de console em 1975, 200 mil unidades foram vendidas só no primeiro Natal.

Porém, como a Atari não se preocupou em patentar o *Pong* até seu lançamento, e como patentes demoram anos para serem concedidas, rapidamente várias empresas começaram a lançar clones do jogo. Além disso, a *Odyssey* tinha uma patente para o *Tênis para Dois*, e ela processou a Atari, acusando a empresa de ter copiado a ideia do jogo. O livro de presença da demonstração do *Tênis para Dois* em 1972, onde consta a assinatura de Bushnell foi usado como prova no tribunal.

A Atari, ao invés de lutar, preferiu um acordo. Eles pagariam à *Odyssey* algo entre 400 mil e 1 milhão de dólares para continuar produzindo o *Pong*. Logo, as patentes da *Odyssey* é que estavam sendo infringidas pelos clones e, enquanto a *Odyssey* lutava contra as outras empresas por imitar o *Pong*, digo, o *Tênis para Dois*, a Atari seguia em frente e inventava novos jogos, como *Space Race*, *Tank*, *Gotcha* e *Breakout*, além de novas versões do *Pong* – uma das quais para ser jogada por quatro pessoas ao mesmo tempo.

## 8.21 Ethernet – 1973

Se você já ligou seu computador a uma rede local, então provavelmente já usou uma conexão no padrão Ethernet, mesmo sem saber. A [Figura 8.14](#) mostra um moderno cabo Ethernet RJ45 usado para conectar computadores a uma rede local.



**Figura 8.14: Cabo Ethernet RJ45.<sup>25</sup>**

A história do padrão Ethernet se inicia em 1973, quando Robert Metcalfe (Estados Unidos, 1946), que havia acabado de concluir seu doutorado em Harvard, foi trabalhar no Xerox PARC. Ele tomou contato com um artigo obscuro publicado na conferência da Federação Americana de Sociedades de Processamento de Informação (AFIPS) em 1970. O artigo em questão, de Norman Abramson, tinha como título “The Aloha System – Another Alternative for Computer Communications”. Ele descrevia uma inovadora tecnologia de criação de redes de computadores por rádio, que foi chamada de “ALOHAnet”.

O artigo inspirou Metcalfe com o auxílio de David R. Boggs a criar um modelo de comunicação entre computadores em rede local: a Ethernet. O primeiro protótipo testável foi finalizado em novembro de 1973, um sistema que conectava 100 estações de trabalho com 1 km de cabos a uma velocidade de 2,93 Mbit/s (2,93 milhões de bits por segundo). A Xerox solicitou patente da invenção em 1975.

Em 1979, Metcalfe saiu do PARC para fundar a empresa 3Com e convenceu a Xerox, DEC e Intel a unir esforços para tornar o

Ethernet um padrão de porta de comunicação entre computadores. Um comitê começou a trabalhar no padrão, que foi finalizado em 1983 com o nome de IEEE 802.3. Ele definia aspectos de hardware e software básico que seriam necessários para orientar os fabricantes a construir interfaces Ethernet compatíveis entre si.

O nome “Ethernet” foi inspirado no conceito do século XIX de éter luminoso (*luminiferous ether*), que seria o meio hipotético pelo qual as ondas de luz se propagavam. Essa hipótese jamais foi provada e hoje é considerada falsa em favor de outras teorias.

## 8.22 Xerox Alto – 1973

Desde o início de 1973, o centro de pesquisas da Xerox em Palo Alto (PARC) desenvolveu um projeto para verificar os limites da tecnologia, no sentido de produzir computadores pessoais que pudessem substituir os terminais ligados a grandes mainframes.

O principal resultado dessa pesquisa foi um computador pessoal chamado “Xerox Alto”, ou simplesmente, “Alto” ([Figura 8.15](#)). Ele nunca chegou a ser comercializado, mas a Xerox produziu cerca de 2 mil unidades que foram usadas na própria empresa e doados a instituições de pesquisa e universidades.

O Xerox Alto foi o primeiro computador a rodar Smalltalk. Ele inovou em vários aspectos, como no uso do mouse com três botões, que, embora fosse conhecido até então, não era muito usado em computadores devido à pouca aplicabilidade. Como o Alto foi o primeiro computador a ter uma interface totalmente gráfica, incluindo o sistema operacional, o mouse acabou sendo bastante útil e, com isso, se tornou até hoje o dispositivo mais popular, depois do teclado, para controlar as ações de um usuário.

Em 1979, Steve Jobs (Estados Unidos, 1955-2011), um dos fundadores da Apple, visitou o PARC e conheceu o Alto. Ele imediatamente percebeu a vantagem de construir e difundir computadores com interfaces gráficas e incorporou essa ideia aos da Apple, inicialmente o LISA e depois o Macintosh. Quando as

máquinas de Jobs passaram a dominar o mercado de computadores pessoais a Xerox percebeu que tinha deixado passar uma excelente oportunidade de negócios e que já era tarde para tentar recuperar o terreno.



Figura 8.15: Xerox Alto.<sup>26</sup>

O Alto pode não ter sido o primeiro computador pessoal, mas ele foi bastante influente no projeto de computadores pessoais que vieram depois, como os da Apple e da IBM. Ele lançou o conceito de computador de mesa (desktop). A máquina toda, com exceção do

monitor, vinha encapsulada e uma única caixa, que caberia no espaço de uma mesa de escritório (sobre ela ou, mais frequentemente, abaixo, visto que essa caixa tinha o tamanho de um frigobar). Sua memória era de 64 kB de 16 bit, expansível para 256 kB.

Ele tinha uma interface de comunicação Ethernet de 3 milhões de bits por segundo, para ser ligado a ARPANET ou redes locais. Com essa porta de comunicação era possível interligar vários Altos de forma a criar uma rede de processamento e comunicações.

Inicialmente o software para o Alto foi desenvolvido com a linguagem BCPL e com a linguagem Mesa, que não foi muito difundida fora do PARC, mas que inspirou Nicklaus Wirth, o criador da Pascal, a definir a Modula-2. O teclado do Alto não tinha uma tecla para o sublinhado (“\_”) porque os engenheiros resolveram substituí-la por uma seta “←” para representar a instrução de atribuição na linguagem Mesa. Estima-se que talvez por isso tenha-se criado o padrão de escrita chamado “CamelCase”, porque usualmente identificadores formados por palavras compostas eram representados com sublinhados como “bolo\_de\_noiva”, notação conhecida como “snake\_case”. (Lembre-se que o hífen não podia ser usado nestes casos porque “bolo-de-noiva” poderia ser interpretado nas linguagens de programação como duas subtrações envolvendo as variáveis “bolo”, “de” e “noiva”.) Bem, com a falta do sublinhado, palavras compostas como essa passaram a ser escritas como “BoloDeNoiva”. O uso das maiúsculas, por sua semelhança a corcovas de camelo, deu origem ao CamelCase.

Outra hipótese para o surgimento do CamelCase, porém, teria sido simplesmente a preguiça dos programadores, pois em relação ao snake\_case ela usa menos teclas.

## 8.23 Micral – 1973

O Micral, produzido na França a partir de 1973, foi o primeiro computador pessoal comercial baseado em microprocessador, no

caso, o Intel 8008 – sucessor do 4004. O Micral modelo N, o primeiro da série, é mostrado na [Figura 8.16](#).



**Figura 8.16: Micral N.<sup>27</sup>**

O Micral não foi projetado para ser um computador de propósito geral, mas sim um controlador de processos. Ele podia ser conectado a até 14 placas eletrônicas através de um slot traseiro chamado “pluribus”. O computador era programado em Assembly através de uma fita perfurada.

Ele substituiu o PDP-8 em algumas aplicações do Institut National de la Recherche Agronomique (INRA) por um quinto do preço. Ele era construído pela Réalisation d’Études Électroniques (R2E) e seu projetista foi François Gernelle (França, 1944).

## **8.24 Salas de Chat Talkomatic – 1973**

O primeiro sistema de salas de conversa online foi criado em 1973 por Doug Brown e David R. Woolley. Ele se chamava “Talkomatic” e

consistia de apenas seis salas virtuais, ou canais, cada qual capaz de acomodar até cinco participantes.

O Talkomatic rodava no sistema PLATO da Universidade de Illinois, o qual era um sistema de propósito educacional online que, além das salas de bate-papo, ainda disponibilizava um serviço de e-mail, mensagens, quadros de mensagens, compartilhamento de tela e jogos multiusuário.

Uma das características de Talkomatic que hoje a maioria dos sistemas de conversa online não tem era o fato de que os caracteres eram transmitidos no mesmo instante em que eram digitados, não sendo necessário assim aguardar que a linha fosse finalizada para ser transmitida.

A tela do sistema era dividida em setores horizontais, cada qual com um dos usuários da conversa online. Além dos cinco participantes ativos na conversa, um número ilimitado de monitores (pessoas que apenas liam as mensagens) podia ser adicionado. O número de participantes ativos em uma conversa era limitado a cinco por conta do tamanho da tela disponível.

Canais abertos podiam ser protegidos pelos seus usuários, de forma a permitir que fossem monitorados apenas por pessoas autorizadas.

Uma das desvantagens é que não era possível usar o Talkomatic para chamar uma pessoa para a conversa. Assim, quando você entrava em uma sala de bate-papo, tinha que conversar com quem estivesse lá. Aparentemente, para a maioria das pessoas que estavam apenas interessadas em flertar isso não era um problema.

Atualmente, um sistema com a mesma funcionalidade do Talkomatic foi recriado na Internet e está disponível em <http://talko.cc/>.

## **8.25 Telefone Celular Pessoal – 1973**

Quando Martin Cooper (Estados Unidos, 1928) fez a primeira chamada em um telefone celular em 3 de abril 1973 ele provavelmente não imaginava o quanto essa tecnologia seria importante para o desenvolvimento da indústria de computadores

três décadas a frente. O primeiro telefone celular pesava cerca de 1 kg ([Figura 8.17](#)) e o próprio Martin confessa que, apesar de ser seu criador, ele não os carregava até ficarem tão pequenos que podiam ser pendurados no cinto e não serem sentidos.

No final dos anos 1960 havia uma única empresa de telefonia na maioria dos países. Nos Estados Unidos era a AT&T: a maior companhia de telefonia do mundo e inventores do celular. Entretanto, o telefone que eles criaram era um aparelho para ser usado exclusivamente em automóveis. As baterias eram muito grandes e o aparelho muito pesado para ser carregado por uma pessoa. Martin, que trabalhava para uma pequena empresa chamada Motorola, acreditava que as pessoas não queriam falar com carros: elas desejavam falar umas com as outras, e assim os celulares montados nos automóveis não satisfaziam a demanda. A meta de Martin era que o número de telefone passasse a indicar uma pessoa e não mais um lugar. A única forma e fazer isso era criando telefones que pudessem efetivamente ser carregados.



**Figura 8.17: Martin Cooper fotografado em 2007 com o primeiro telefone celular de mão.<sup>28</sup>**

O protótipo usado na famosa ligação de 1973 é considerado o primeiro celular verdadeiro porque efetivamente não era um aparelho de rádio, como os que os soldados usavam nas guerras: ele fazia ligações usando o serviço de telefonia comercial. Além disso, não precisava de um veículo para ser levado de um lado a outro, mas podia efetivamente ser carregado na mão. A primeira ligação feita por este aparelho foi simbólica: Martin ligou para seu concorrente na Bell Labs, Joel Engel, que fabricava celulares para

automóveis e disse: “*Joel, I’m calling you from a ‘real’ cellular telephone. A portable handheld telephone.*”<sup>29</sup>

No entanto, a tecnologia ainda não estava totalmente pronta para ir às ruas: 1 kg era muito peso. Foram dez anos de pesquisas e cinco modelos progressivamente menores até que em 1983 o serviço comercial de telefones celulares de mão começou a ser ofertado. Por essa época, o aparelho já pesava cerca de 400 gramas. Os primeiros telefones disponíveis comercialmente, os Motorola DynaTAC 8000x, custavam cerca de 3.500 dólares. Apesar do altíssimo preço, logo se tornaram bastante populares, pelo menos entre os executivos que tinham condições de comprá-los.

Os primeiros celulares eram usados apenas para fazer ligações telefônicas. Algum tempo depois correio de voz foi adicionado como uma característica extra. Levou mais tempo ainda até os fabricantes perceberem que poderiam integrar nesses aparelhos a tecnologia de envio e recebimento de mensagens. Porém, quando isso começou a acontecer – com sistemas de mensagens e e-mails nos anos 1990 – o próprio objetivo do telefone celular mudou. Hoje, com os smartphones, sucessores dos celulares, há pessoas que talvez nem saibam que eles podem ser usados para fazer ligações telefônicas; e muitos que sabem disso não sentem a menor necessidade.

## 8.26 Community Memory, o Primeiro BBS – 1973

Os Bulletin Board System ou BBS, foram uma forma simplificada e precursora das modernas redes sociais. Um BBS é basicamente formado por um ou mais computadores servidores que permitem que usuários armazenem mensagens ou arquivos que são acessíveis publicamente.

Considera-se que o primeiro sistema que poderia ser chamado de BBS foi o Community Memory, disponibilizado em uma rede de terminais de computador na região de Berkeley na Califórnia, em 1973. Ele permitia que pessoas digitassem mensagens em terminais

após inserirem uma moeda de 25 centavos (era assim que o sistema se pagava). Todas as mensagens eram públicas, ou seja, podiam ser lidas gratuitamente por qualquer pessoa que acessasse o sistema. A [Figura 8.18](#) mostra um destes terminais.

Ele permitia que as mensagens fossem identificadas com palavras-chave para facilitar pesquisas. Inicialmente, o sistema foi pensado para ser uma espécie de “classificados” no qual as pessoas podiam comprar e vender coisas, mas rapidamente as pessoas começaram a achar outras aplicações, criando histórias, poesia e diversas formas de comunicação.

Infelizmente, a operação do sistema era cara e quando o computador central parou de funcionar em 1975 o investimento necessário tornou impeditivo que fosse substituído por outro. Assim, o Community Memory foi descontinuado.

Porém, outros BBS passaram a existir depois disso. Eles se tornaram especialmente populares após a disponibilização de microcomputadores e, especialmente de modems, que permitiam aos usuários se conectar aos BBS simplesmente discando um número telefônico através de seu modem. Isso, claro, ocorreu antes do advento da Web. Com o surgimento da World Wide Web, os BBS entraram em declínio e hoje praticamente não existe mais nenhum sistema do tipo em atividade.



Figura 8.18: Um terminal do Community Memory.<sup>30</sup>

## 8.27 Intel 8080 – 1974

Depois do sucesso dos microprocessadores 4004 e 8008 a Intel lança em 1974 um novo modelo, o i8080 (oitenta-oitenta). Ele é

considerado o processador que iniciou a revolução dos computadores pessoais. De fato, o 8080 é o ancestral direto da maioria dos novos processadores ainda em uso. A [Figura 8.19](#) mostra o anúncio dele na revista *Electronics* de 2 de maio de 1974.

**From CPU to software, the puter is here.**

Intel's new 8080 n-channel microcomputer is here—immediately easy to interface, simple to program and with up to 100 times the performance of current MCS microcomputers.

Most of all, the 8080 is now—in production at Intel and available in volume quantities, too! It is also available through distributors and independent suppliers, giving you a wide choice of peripheral circuits and a new variety of the Intel's unique program and hardware development tools.

Supplied with software packages, design documentation and manuals, and backed by more than 100 managers of microcomputer expertise, the 8080 is the ideal processor to complete custom MCS and memory-interface logic requirements. It is the industry's first general purpose n-channel microcomputer and the first high performance single-chip CPU, with extremely simple instruction requirements and excellent compatibility with existing 4004 and 8008 microprocessors.

As such, the 8080 extends the concepts based on Intel's p-channel microprocessor to a wide range of systems that need two multipoint controllers and processors. These systems include intelligent terminals, point-of-sale systems, process and resource controllers, advanced

calculator, word processors, self-calibrating instruments, data loggers, communications controllers, and many more.

You can use 8080 input and 8080 output channels, in addition to unbuffered internal levels, directly addressable memory and peripherals, and provide a wide range of possibilities in a single microcomputer.

Interface in a natural and obvious way with the 8080 because all connections are by direct-on-the-CPU chip, bus and improved interfaces. There are separate data, address and control buses.

The 8080 microcomputer has 271 basic instructions, including the RISC-style new ones that make possible such features as vectorized multi-level interrupt, unlimited stackable nesting and very fast decimal and binary arithmetic.

Program development for the 8080 may be done either on a large scale, by using the Intel software tools products (P-ASM, assembly language compiler, macro assembler and monitor) or on an Intel 8080 development system, using the 8080 ROM monitor, PCON, PDISK and UC2000s. All are available now, and will be expanded over 8080LS integrated circuit.

The 8080 is easier to use and more powerful than the high performance microprocessor it supplanted. It is here now to take over from the invention of the microcomputer. Use it and help lead the industry in production and design using Intel Corporation, 2801 Bayshore Avenue, Santa Clara, CA 95051, (408) 924-1971.

**intel** Microcomputers. First from the beginning.

Electronics May 2, 1974

20

21

22

23

Figura 8.19: Anúncio do Intel 8080.<sup>31</sup>

Comparado com seus antecessores 4004 e 8008, o 8080 era bem mais poderoso. Ele funcionava a uma taxa de 2 MHz (dois milhões de ciclos de relógio por segundo) e integrava em um único chip circuitos que os antecessores requeriam que fossem adicionados separadamente.

Outra diferença foi a motivação de sua construção. Os modelos anteriores foram projetados a pedido de companhias específicas, o 4004 para a Busicom e o 8008 para a Datapoint. Já o 8080 foi criado para ser um microprocessador “de varejo”, ou seja, sem um cliente específico em mente.

Federico Faggin (Itália, 1941), que havia criado o 4004 foi o arquiteto-chefe do projeto do 8080. Ele comentou certa vez que

quando dava palestras sobre o 4004 e 8008 em empresas, os engenheiros se queixavam da arquitetura e performance desses circuitos. Como muitas das críticas eram procedentes, ele se propôs a eliminar essas falhas com o projeto do 8080.

A ideia dele foi integrar os vários circuitos acessórios do 8008 em um único chip mais eficiente. Assim, ele teria 40 pinos e 2MHz contra os 18 pinos e 0,5 MHz do 8008. Em palavras simples, ele estava buscando aquilo que hoje se conhece como “computador em um chip”.

A Intel demorou alguns meses para aprovar a proposta de Faggin porque queria ver como o mercado reagiria ao 4004 e 8008, mas em outubro de 1972 o projeto iniciou. Faggin contratou o engenheiro Masatoshi Shima (Japão, 1943), que havia trabalhado para a Busicom e o ajudara no projeto do 4004.

O chip era vendido em 1974 a 360 dólares, o que muitos viram como uma provocação à IBM e a seu sistema IBM 360, que custava milhões.

## 8.28 Altair 8800 – 1974

Com a disponibilização do Intel 8080 veio também o primeiro grande sucesso em termos de computador pessoal, o Altair 8800 ([Figura 8.20](#)), fabricado pela Micro Instrumentation Telemetry Systems (MITS). Essa empresa até 1974 fabricava apenas calculadoras e componentes de telemetria para modelos de foguetes. Porém, Henry Edwards Roberts (Estados Unidos, 1941-2010) viu no lançamento do Intel 8080 a possibilidade de vender um computador montado ou parcialmente montado. O maior problema na época é que quem quisesse ter seu próprio computador precisava pagar uma fortuna para as empresas que os construíam ou procurar projetos e peças e montar a própria máquina. Apenas pessoas com grandes habilidades técnicas eram capazes de construir computadores com sucesso, e as peças quase sempre só eram encontradas no Vale do Silício na Califórnia. Assim, todo um

mercado de computadores como eletrodomésticos ainda estava por ser explorado.

O maior problema que Roberts previu era o fato de que o chip do 8080 custava 360 dólares no varejo e ele queria que o computador finalizado não custasse muito mais do que 400 dólares. Para viabilizar o projeto, propôs à Intel comprar 1.000 chips pagando 75 dólares por cada. A Intel aceitou, mas com a condição de que ele receberia chips de segunda escolha, ou seja, aqueles com pequenos defeitos na aparência, que não afetavam seu funcionamento elétrico. O negócio foi viabilizado e o caminho para o Altair 8800 foi aberto.

O Altair era um computador simples. Não tinha teclado nem monitor. Comandos e dados eram inseridos a partir de chaves binárias e resultados de computações eram visualizados em uma fileira de LEDs que podiam estar ligados ou desligados. Pensar que uma máquina como essa pudesse ser bem-sucedida no mercado doméstico foi uma aposta alta, mas Roberts venceu. O modelo gerou muito interesse entre aficionados em eletrônica e curiosos por novos brinquedos, e vendeu grandes quantidades.

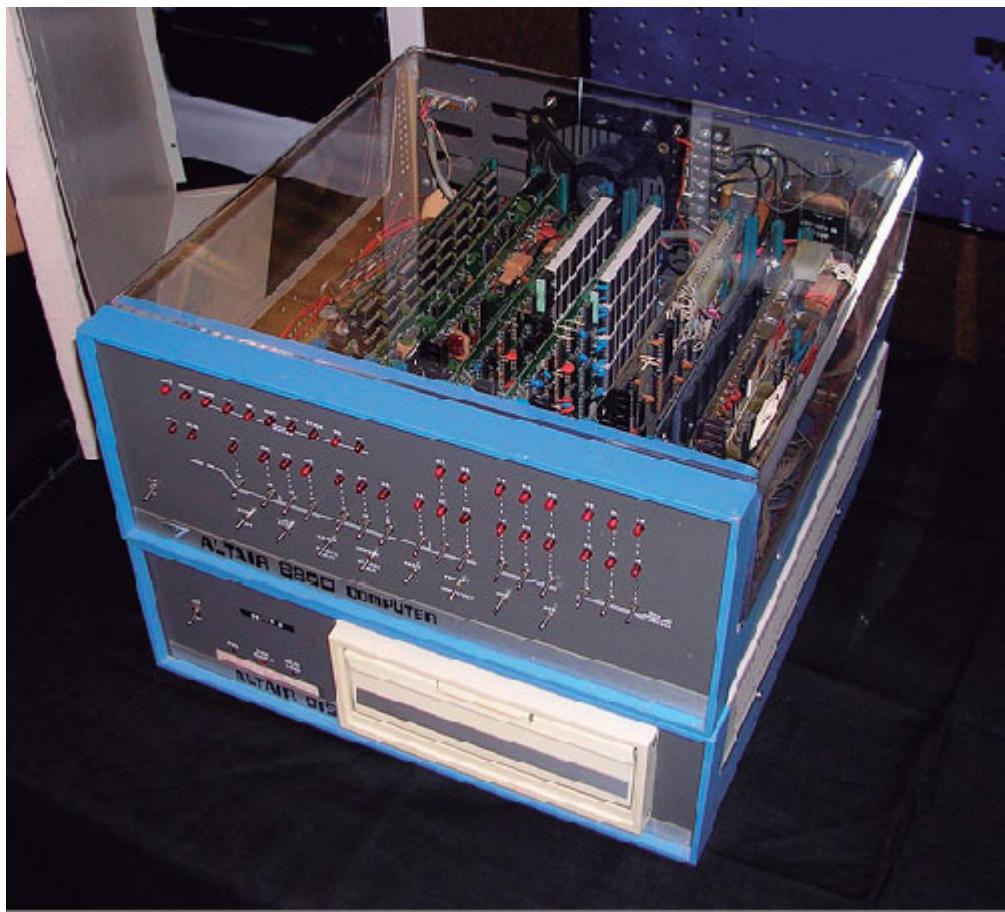


Figura 8.20: Altair 8800.<sup>32</sup>

O Altair 8800 foi destaque na capa da revista *Popular Electronics* em janeiro de 1975. A máquina básica era vendida por 439 dólares e ainda podiam ser comprados alguns opcionais: 1, 2 ou 4 kB adicionais de memória RAM (o Altair vinha com apenas 256 bytes instalados), interfaces de comunicação serial ou paralela e interface de comunicação com gravadores de fitas magnéticas K7. A revista tinha na época 450 mil leitores e Roberts esperava que alguns deles se interessassem pelo computador mesmo sem ter uma ideia clara sobre o que fazer com ele.

O computador que se via no anúncio, porém, era uma casca vazia. Ocorre que Roberts enviou o único protótipo existente do Altair para Les Solomon, diretor da *Popular Electronics*, usando o único meio

seguro e rápido que existia na época: a Railway Express. Só que, para seu azar, a Railway Express sofreu uma greve de funcionários; a empresa foi a falência e perdeu o pacote com o Altair. Ele nunca mais foi recuperado. Para criar o anúncio, então, eles montaram uma casca atrás da qual não havia nenhum circuito eletrônico, ou seja, era uma foto apenas para demonstração. No interior da revista, porém, pode-se ver algumas partes internas do protótipo perdido que foram tiradas antes que fosse despachado.

Existem duas teorias para explicar a escolha do nome “Altair” e aparentemente não há consenso sobre qual seria verdadeira. São as seguintes:

- A primeira teoria provavelmente é falsa, mas é a mais interessante: dizem que Les Solomon perguntou para sua filha, que assistia a *Jornada nas Estrelas*, como eles chamavam o computador da *Enterprise*. Ela respondeu “computador”, ao que o pai comentou que ela não tinha ajudado muito. Então a menina sugeriu que ele chamassem o novo computador de “Altair”, porque era para onde a *Enterprise* estava indo naquele episódio. De fato, no episódio “Amok Time” a *Enterprise* visita este sistema estelar.
- Outra teoria, não tão bonita, mas provavelmente mais verdadeira diz que o computador ia se chamar PE-8, ou “Popular Electronics 8”, mas os criadores acharam o nome muito bobo. Um deles comentou que seu lançamento seria um evento estelar e que, portanto, deveriam dar ao computador o nome de uma estrela. Assim, escolheram Altair, a décima primeira estrela mais brilhante do céu.

Mais importante é que a *Popular Electronics*, com o anúncio do Altair, caiu nas mãos de um estudante de Harvard chamado William (Bill) H. Gates (Estados Unidos, 1955) e seu amigo Paul Allen (Estados Unidos, 1953), que trabalhava na Honeywell. Eles já haviam escrito algum software para o Intel 8008 e perceberam rapidamente que o Intel 8080 poderia rodar um interpretador BASIC.

Eles enviaram uma carta à MITS informando que já tinham o BASIC para o Altair. Como Roberts ficou interessado, começaram a trabalhar no projeto.

Eles desenvolveram uma primeira versão usando um PDP-10 para o qual eles escreveram um emulador de 8080, visto que não tinham nenhum Altair disponível. Em março de 1975, Allen voou para o Novo México para testar o BASIC em um Altair verdadeiro. Roberts recebeu-o no aeroporto e havia lhe reservado o hotel mais caro da cidade a um custo de 40 dólares por noite. Allen disse que aquilo era mais dinheiro do que tinha trazido e Roberts pagou a conta, mas ficou se perguntando que tipo de profissional de software era aquele que não tinha nem 40 dólares.

No dia seguinte, um Altair de 7 kB de memória levou 15 minutos para carregar uma fita com o interpretador BASIC que Gates havia produzido um pouco antes de Allen sair de Boston. Ao final, o teletipo ligado à máquina escreveu: "MEMORY SIZE?". Allen entrou com 7.168 (7 kB) e a máquina respondeu "READY". Para espanto deles, o interpretador estava funcionando. Ele ainda tinha vários bugs, é verdade, mas foram consertados. Gates e Allen foram contratados pela MITS, Gates como especialista em software e Allen como vice-presidente e diretor de software.

Em 22 de julho de 1975, MITS assinou um contrato com Gates e Allen por 10 anos de direito sobre o BASIC do Altair, dando a eles 3 mil dólares na assinatura e mais royalties por cada cópia vendida. Uma cópia isolada do software era comercializada a 150 dólares, mas para quem comprasse a máquina e periféricos do MITS custava apenas 60 dólares no pacote. Para computadores com mais memória, podiam ser compradas versões mais sofisticadas e mais caras do BASIC. Este valor, considerado alto, foi uma das causas do início da pirataria de software moderna, pois os usuários compravam seus computadores mais baratos de outros fornecedores e depois trocavam entre si cópias do BASIC. Isso enfureceu Gates, que ameaçou descontinuar o programa se a pirataria continuasse, mas sem nenhum resultado.

Em 1977 houve um processo judicial entre a MITS e uma nova empresa criada por Gates e Allen, chamada “Micro-Soft”, que também passou a vender o interpretador BASIC a um preço bem mais baixo, de forma a desmotivar a pirataria. Roberts entendia que, uma vez que havia contratado e pago Gates e Allen para produzirem o software, ele pertencia à MITS. O juiz não concordou, entretanto, e deu à Micro-Soft o direito de também vender cópias do BASIC. Dessa forma, Gates prosseguiu no caminho para se tornar o primeiro bilionário da computação.

## 8.29 TCP-IP e o Surgimento da Internet – 1974

Para que dois ou mais computadores consigam se comunicar através de uma rede é necessário que se estabeleçam padrões de comunicação. Esses padrões são chamados de “protocolos”. O mais importante e difundido protocolo de comunicação hoje é o TCP/IP da Internet, que significa “Transmission Control Protocol / Internet Protocol”.<sup>33</sup>

Esse protocolo nasceu em 1974 e seus criadores foram Vinton Gray Cerf (Estados Unidos, 1943) e Robert Elliot Kahn (Estados Unidos, 1938).

Cerf trabalhou durante sua pós-graduação na UCLA, onde ajudou Leonard Kleinrock (Estados Unidos, 1934) a conectar os dois primeiros computadores da ARPANET. Ele foi responsável pela definição dos primeiros protocolos da ARPANET. Ainda nessa época, conheceu Kahn, que desenvolvia hardware para a ARPANET através da empresa BBN (Bolt, Beranek and Newman).

Em 1972, Kahn trabalhava em um projeto para estabelecer uma rede de computadores usando ondas de rádio e durante esse projeto ele se convenceu da necessidade de criar um protocolo aberto para que não apenas computadores, mas redes de computadores, pudessesem se interconectar, independentemente de sua arquitetura ou configuração de software. Kahn definiu, então,

quatro princípios que seriam a base para o protocolo TCP (Transmission Control Protocol):

- *Conectividade de rede.* Qualquer rede pode se conectar a outra rede através de uma interface, ou *gateway*.
- *Distribuição.* Não deve haver uma administração ou controle central para esta rede.
- *Recuperação de erro.* Pacotes perdidos devem poder ser retransmitidos.
- *Design caixa preta.* Nenhuma mudança interna deve ter que ser feita em uma rede para que ela possa se conectar a outras redes.

Em 1973, Cerf se reuniu a Kahn no projeto. Após muito estudo, chegaram a definir um primeiro protocolo núcleo, que foi chamado Transmission Control Program (TCP), hoje rebatizado como Transmission Control Protocol. O documento revisado foi publicado em 1974.

Com esse protocolo, foi possível ligar a ARPANET a um número de outras redes bastante heterogêneas criando-se assim a Internet.<sup>34</sup> Um ditado da área até diz que com TCP era possível ligar em rede até um telefone de lata, do tipo que as crianças costumam fazer. Com a criação da Internet, anos mais tarde, o TCP/IP passou a ser seu protocolo padrão.

Não há uma data ou evento preciso que marque o nascimento da Internet. Considera-se que a ARPANET tenha sido a precursora legítima. Dessa maneira, quando os dois primeiros computadores da ARPANET se comunicaram, isso marcou o nascimento da rede como a conhecemos hoje.

Porém, havia várias redes espalhadas pelos Estados Unidos e pelo mundo e a ARPANET era apenas uma delas. Podemos considerar, então, que a adoção do protocolo TCP/IP pela ARPANET e posteriormente por outras redes é também um marco fundamental na história da Internet, pois o TCP/IP marcou o início da

possibilidade de interconexão das várias redes que acabaram aos poucos formando o que hoje conhecemos como Internet.

O termo “Internet”, inclusive, nasceu com esse protocolo, visto que IP significa “Internet Protocol”. Nos anos 1970, o termo só era usado como referência ao protocolo. Apenas no final dos anos 1980, quando a ARPANET se ligou à NSFNET, as duas maiores redes de computadores do mundo, através do protocolo TCP/IP é que “Internet” passou a designar a rede como um todo. Hoje, entende-se por Internet o conjunto dos computadores e redes interligados por esse protocolo.

## **8.30 COBRA, Computadores Brasileiros – 1974**

Um dos desdobramentos do projeto do Pato Feio, como já visto, foi a criação da Computadores Brasileiros (COBRA), uma empresa estatal que tinha como objetivo transformar o sucessor do Pato Feio, o G-10, em um produto comercial nacional.

A fundação da empresa teria sido apoiada pela Marinha Brasileira, o Banco Nacional de Desenvolvimento Econômico e Social (BNDES) e a Ferranti inglesa. A equipe da empresa foi formada inicialmente por membros do projeto Pato Feio da USP, da Pontifícia Universidade Católica do Rio de Janeiro (PUC/RJ) e do Serviço Federal de Processamento de Dados (SERPRO), criado em 1964 para dar agilidade a setores estratégicos da administração pública.

Inicialmente, a empresa produzia computadores da Ferranti. Com o conhecimento adquirido, novos modelos e placas foram criados, até que em 1980 nasceu o COBRA 530, o primeiro computador comercial totalmente projetado e construído no Brasil.

Outro modelo que foi muito popular no país nos anos 1980 foi o COBRA 210, fabricado a partir de 1983 com um processador Z80, compatível com 8080 e dois drives para disquetes de 8 polegadas ([Figura 8.21](#) ).

A COBRA beneficiou-se durante os anos 1980 da Lei de Reserva de Mercado, que impedia a importação de computadores estrangeiros

se houvesse similares no Brasil. A falta de competição, porém, foi considerada não saudável e um certo atraso nos parques computacionais brasileiros foi o resultado dessa polêmica política, que acabou sendo relaxada a partir dos anos 1990.

Assim, com a abertura do mercado a partir do governo Collor, a maioria das empresas brasileiras de computadores não era competitiva e acabaram fechando as portas ou sendo vendidas. A COBRA foi adquirida pelo Banco do Brasil, e em 2013 mudou seu nome fantasia para BB Tecnologia e Serviços (BBTS).



Figura 8.21: COBRA 210.<sup>35</sup>

### 8.31 *The Mythical Man-Month – 1975*

O mesmo Fred Brooks que nos anos 1960 coordenou o projeto do sistema operacional OS/360 para a IBM lançou em 1975 um livro que seria um marco para a área de desenvolvimento de software *The Mythical Man-Month: Essays on software engineering*.<sup>36</sup>

A motivação para o livro foi a própria Lei de Brooks, cunhada após o projeto do OS/360: “Adicionar mais programadores a um projeto de software atrasado só faz com que ele atrasse mais ainda.” Na capa do livro é mostrada uma cena pré-histórica, com animais se debatendo em um campo de piche – quanto mais força eles fazem, mais afundam.

Porém, a tendência de que os engenheiros de software continuassem repetindo esses erros mesmo anos após a publicação do livro fez com que Brooks o considerasse a Bíblia da Engenharia de Software: “Todo mundo cita, algumas pessoas leem e poucas efetivamente seguem”.

O principal problema com projetos de software é que eles precisam amadurecer. Não é como construir uma estrada; você pode dividir a estrada em vários lotes e fazer cada um deles em paralelo, multiplicando assim a velocidade do projeto. No entanto, no caso de um projeto de software, a divisão de trabalho não é tão simples. Para que um sistema funcione, todas as suas partes precisam funcionar em harmonia. Quando muitas pessoas trabalham no projeto, muitas linhas de comunicação se estabelecem e muito conhecimento precisa ser compartilhado, o que acaba levando a falhas. Assim, em projetos de alta complexidade que já são grandes, adicionar mais pessoas cria novas linhas de comunicação e, portanto, mais gasto de tempo.

Devido a isso, modificações em programas, mesmo que sejam correções de bugs, muitas vezes acabam gerando novos bugs. Os profissionais de computação cantam uma paródia da música “99 garrafas de cerveja” que, traduzida, é mais ou menos assim: “99 bugs no software, 99 bugs no software, pegue um deles e conserte. 127 bugs no software...”.

### **8.32 Computador Portátil IBM 5100 – 1975**

Seis anos antes de lançar o famoso IBM-PC, a IBM introduziu seu ancestral, o 5100 ([Figura 8.22](#)), um computador portátil com 16 a 64

kB de memória, que rodava BASIC e APL a um preço relativamente acessível, a partir de 9 mil dólares.



**Figura 8.22: IBM 5100.<sup>37</sup>**

Na verdade, quando o IBM-PC foi lançado em 1981 ele inicialmente foi chamado de IBM 5150, ou seja, uma máquina da série 5100. Não demorou a esse nome ser abandonado, contudo, pois a nova máquina pouco tinha a ver com o 5100 ou seus sucessores.

O computador era chamado de portátil porque seus 25 quilos podiam ser transportados com um equipamento opcional disponibilizado. Pode-se ver na figura que uma única caixa incluía a unidade de processamento, teclado, unidade de fita magnética e um pequeno monitor CRT de 5 polegadas (16 linhas de 64 caracteres). O 5100 foi um feito para a época porque até então computadores com o mesmo tamanho e peso não incluíam o monitor CRT e não tinham tanta memória.

## 8.33 Sphere – 1975

Outro computador portátil lançado em 1975 foi o Sphere ([Figura 8.23](#)), vendido em quatro configurações: Sphere-1 com CPU com 4 kB de memória, teclado, monitor de 512 caracteres; Sphere-2 adicionando uma interface para áudio K7 ou modem e comunicação serial; Sphere-3 com 20 kB de memória, capaz de rodar BASIC estendido; e Sphere-4, que substituía a interface com K7 por um sistema de dois drives para disco flexível de 8 polegadas e uma impressora de linha.

O Sphere foi projetado com o microprocessador Motorola 6800, criado em 1974 e inspirado no processador do PDP-11. Esse processador tinha uma vantagem em relação a outros porque todos os seus pinos funcionavam com 5V, enquanto que outros processadores necessitavam de até 3 voltagens diferentes, dependendo do pino, para funcionar.

Uma de suas características inovadoras era a possibilidade de reinicializar a máquina “na marra”, com o pressionar de duas teclas. Assim, ele foi o primeiro computador conhecido a implementar o que hoje entendemos como “Ctrl+Alt+Del”.

O Sphere foi criado pelo dono da empresa, Mike Wise, que tinha uma destas máquinas em casa. Sua esposa dizia que a tampa traseira não fechava direito porque ele, na verdade, não queria que fosse tampada, visto que Wise não parava de fuçar dentro da máquina.

Segundo ele, 1.300 computadores Sphere foram vendidos, metade já montados e metade na forma de kit, bem mais barato.

# Go Computer Now!

## Why not?

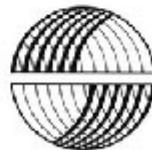
**FROM \$860 TO \$11,300 SPHERE CAN'T BE BEAT!**

SPHERE starts with a CPU using a Motorola 6800 microprocessor, a Real-Time Clock, 4K of dynamic memory, 1K of PROM software. The CRT Board generates 16 lines by 32 characters of ASCII on a television or video monitor. Keyboard is complete with numeric and cursor editing keypads. From here, hardware can be expanded to your hearts desire with extra memory boards (up to 64K), serial communications interface, cassette interface, Modem, digital I/O (as many as you need), Floppy Disk memory (up to 4 disks), 8 computer terminals, line printer etc...all from one MC6800 chip.

With SPHERE Computer, stand-alone development is just the beginning, you can configure your system to handle your problem solving/record keeping needs. All SPHERE Computer Systems come complete with useable software languages. Available are "PDS" 1K Basic, or extended Basic Compiler. When computer is turned on, it immediately goes into a command mode, so that you can instantly start programming. "PDS" contains a mini-assembler, editor, debugger, and utility command set in 1K of PROM. Also available is a 1K subset of Basic. Our extended Basic compiler is complete with string, matrix, and file functions, and requires 12K of memory. With this software you can perform your applications whether it be accounting, home management, education, security monitoring, research, business, etc. Why not invent your own application? For play or for work year biggest problem is no problem at all. \$860 is the start for an operating Computer System Kit. Your computer is ready and comes complete with operator manuals sufficient for first-time computer users. Contact us today for more information.

KIT	ASM	SPHERE	KIT	ASM	SPHERE
550	\$129	ONE-CARD COMPUTER: Motorola 6800 microprocessor, 4K RAM, 1K PROM containing a Program Development System, a REAL-TIME CLOCK, 16 LINES OF TEXT, 32 Character Video Display, and a serial type interface. This is the 100-character per line, extended to the hobby user for a limited time on a single card.	599	\$169*	Includes all features of SPHERE 1 plus serial communication and audio cassette or Modem interface.
560	\$229	CPU BOARD: Motorola 6800 microprocessor, 4K RAM, 1K PROM containing an EDITOR, ASSEMBLER, DEBUGGER, COMMAND LANGUAGE, CASSETTE INTERFACE, DUMPER, UTILITIES, and a REAL-TIME CLOCK.	714	\$399*	SPHERE 2: includes all the features of SPHERE 1, plus memory totaling 256K which includes our extended BASIC language.
860	\$169*	SPHERE 3: includes all the features of SPHERE 2, except the source has been reduced to an IBM-compatible Dual Floppy Disk system. This system includes a 32K operating system and BASIC language and a 95 line line printer.	868	\$995*	SPHERE 4: includes all of the features of SPHERE 3, except the source has been reduced to an IBM-compatible Dual Floppy Disk system. This system includes a 32K operating system and BASIC language and a 95 line line printer.

\*The ASSEMBLED SPHERE System includes the complete chassis, and video monitor as outlined below.



**SPHERE**  
CORPORATION

291 South 560 West Dept. 429 (801) 292-8466  
Bountiful, Utah 84010

23

Figura 8.23: Anúncio dos computadores Sphere.<sup>38</sup>

## 8.34 Circuito MOS 6502 – 1975

Em 1969 foi criada uma empresa chamada MOS Technology Inc. (na qual MOS significa “Metal Oxide Semiconductor”) com a finalidade de ser uma segunda fonte de produção para os circuitos integrados projetados pela Texas Instruments. Porém, no início dos

anos 1970, a Texas resolveu lançar sua própria linha de calculadoras, em vez de apenas vender os chips; e ela venderia a calculadora completa mais barato do que custava um chip no mercado. Isso levou uma boa parte das fabricantes de chips da época à falência. Quem quisesse sobreviver teria que achar outras aplicações para seus chips.

A MOS passou, assim, a ser uma importante fornecedora para a Atari, que então fabricava o videogame *Pong*. Outro importante movimento aconteceu em 1975 quando a Motorola proibiu seus funcionários de pesquisar e desenvolver chips que seriam mais baratos do que o 6800, seu carro-chefe. Vários funcionários saíram da Motorola e se integraram a outras empresas, inclusive a MOS. Chuck Peddle (Estados Unidos, 1937) e Bill Mensh (Estados Unidos, 1945), juntamente com outros engenheiros, juntaram-se assim à MOS e passaram a projetar e construir finalmente o circuito 6501, cujo projeto era uma simplificação do 6800 e que por isso mesmo acabou sendo quatro vezes mais rápido que este. Ambos tinham uma CPU de 1 MHz, mas como o 6501 era mais simples, ele executava mais instruções em mesma quantidade de ciclos de relógio.

O objetivo de Peddle era realizar aquilo que a Motorola não permitiu: produzir um microprocessador que pudesse ser vendido por apenas 25 dólares, um número mágico que ele obteve entrevistando clientes na época que trabalhava na Motorola. Acontece que ele fazia demonstrações do 6800 e impressionava a todos, mas o preço de 300 dólares não atraia.

Conta-se que Peddle teria realizado um feito inédito para a época: ele simplesmente sentou-se e passou a desenhar o novo circuito 6501 no papel. Quando finalizado, ele o construiu e funcionou de primeira: até então ninguém tinha conseguido isso, pelo que se sabe. Usualmente um circuito só funcionava a partir da 10<sup>a</sup> tentativa de construção.

O 6501 foi produzido a um custo de 20 dólares. Por algum motivo, porém, eles não chegaram a comercializar esse circuito e passaram a produzir outro, o 6502. Ambos eram idênticos em funcionalidade, mas, diferente do 6502, o 6501 ainda tinha os mesmos pinos e funções do Motorola 6800.

Sabe-se que nos anos 1970 cerca de 70% dos circuitos integrados que saiam das linhas de montagem eram defeituosos e precisavam ser descartados. Isso, claro, aumentava muito o custo dos circuitos bons, que eram vendidos. A MOS conseguiu criar um processo que reduziu as perdas para 30%, barateando assim a produção.

Peddle participou em 1975 da WestCon (Western Electronics Show and Convention), onde pretendia vender os circuitos 6501 e 6502 em seu *stand* por 25 dólares. Porém, ao chegar lá, ele foi informado pela organização que para manter a feira em um alto padrão não seria permitido realizar vendas nas tendas. Em resposta, ele foi até o hotel mais próximo e montou um *stand* lá mesmo, no qual sua esposa ficou sentada com dois jarros cheios de circuitos integrados que eram vendidos por 25 dólares. Nem todos foram comercializados, felizmente, pois como ainda tinham relativamente poucos circuitos, deixaram os bons na parte de cima e os defeituosos no fundo dos jarros apenas para dar volume e impressionar, mostrando que o circuito estava em franca produção e era abundante.

Rapidamente, a Motorola processou a MOS por quebra de patente, visto que os oito de seus engenheiros que foram para a MOS levaram consigo conhecimento sobre muitos produtos desenvolvidos nela. Para a empresa, o 6502 era uma cópia do 6820. A MOS, ciente de que realmente tinha usado muitas ideias da Motorola, concordou em pagar *royalties* sobre cada circuito vendido.

Nesse meio-tempo, Peddle andava pelo país tentando conquistar mercado baseado em grandes empresas para seu circuito. Durante a viagem, ficou sabendo que dois rapazes chamados Steve Jobs (Estados Unidos, 1955-2011) e Steve Wozniak (Estados Unidos,

1950) estavam tendo dificuldades para fazer funcionar o 6502 em um protótipo de computador que construíam em uma garagem. Peddle passou por lá e deu uma mãozinha a eles, mesmo convicto de que não tinha esquematizado o 6502 para ser a base de um computador pessoal. “Nem em um milhão de anos”, dizia, “o 6502 era para uso industrial, não para brinquedos.” Apesar disso, o pequeno projeto de Jobs e Wozniak funcionou: o primeiro computador Apple.

Dessa forma, o 6502, juntamente com o 8080 e seu similar Z80, acabaram sendo a base para a explosão da computação pessoal que iniciou na segunda metade dos anos 1970. Computadores e consoles de jogo como Apple, Commodore, Atari e Nintendo usaram o 6502 ou suas variantes até pelo menos 1984. A MOS, por sua vez, acabou sendo adquirida pela própria Commodore, mas manteve a fabricação de chips 6502.

### 8.35 Supercomputadores CRAY – 1976

Você deve se lembrar de Seymour Cray, engenheiro da CDC, que queria e conseguiu fazer em 1965 o computador mais rápido do mundo. Bem, ele não parou depois do primeiro sucesso. Em 1972, ele fundou a CRI (Cray Research Incorporated). O primeiro produto da empresa, instalado no Laboratório Nacional de Los Alamos em 1976, foi o supercomputador Cray-1 ([Figura 8.24](#)). Ele foi um sucesso porque era muito mais rápido do que qualquer outro computador da época. Seu custo inicial era de 8,8 milhões de dólares. No total, apenas 80 máquinas do tipo Cray-1 foram vendidas, mas considerando seu preço, isso não foi nada mal.

O Cray-1 era capaz de executar 80 milhões de instruções de ponto flutuante por segundo (80 MFLOPS). Seu sucessor, o Cray X-MP de 1982, era 10 vezes mais rápido. Já o Cray-2, lançado em 1985 era capaz de 1,9 bilhões de instruções de ponto flutuante por segundo (1,9 GFLOPS). Isso era algo fantástico para a época, e batia de longe outros computadores de então. Inclusive, houve até restrições

de venda desses modelos para determinados países por medo de que fossem usados para finalidades bélicas e nucleares. Ironicamente, a velocidade do Cray-2 correspondia mais ou menos à velocidade do processador de um smartphone nos dias de hoje.



Figura 8.24: Cray-1.<sup>39</sup>

O que tornava o Cray tão rápido para a época era o fato de que ele trabalhava com processamento vetorial. A maioria das aplicações científicas lida com grandes volumes de dados e normalmente a mesma operação é executada sobre milhões de dados. Um computador normal buscaria os dados na memória um de cada vez e executaria a operação sobre ele, armazenando os resultados na memória um a um. Porém, o Cray podia pegar todo um vetor de dados uma única vez e operar com todos os valores com a mesma quantidade de ciclos de memória que seriam usados para operar um

só valor. Não era exatamente uma máquina com vários processadores, como temos hoje, mas um computador com processador que operava sobre várias posições de memória ao mesmo tempo.

## **8.36 Primeiro Sistema Operacional para Microprocessador: CP/M – 1976**

Em 1972, Gary Kildall (Estados Unidos, 1942-1994) comprou um processador Intel 4004 e começou a escrever software para ele. Algum tempo depois, visitou a Intel e passou a trabalhar como consultor um dia por semana para a pequena divisão de microprocessadores. Nesta época, escreveu o compilador para a primeira linguagem de programação para um microprocessador, que ele chamou de PL/M (Programming Language for Microcomputers), o que era uma brincadeira com a Linguagem PL/I (Programming Language One) da IBM.

Como pagamento, a Intel deu-lhe um kit de desenvolvimento Intellec-8, que foi em seguida atualizado para funcionar com um processador 8080. Em 1973, em gratidão por mais programas desenvolvidos para a Intel, Kildall ganhou um drive de disquete. Como não havia software para operar esse drive, ele pediu a um amigo, John Torode, que escrevesse um para facilitar o controle do drive.

Com o surgimento desse software, houve um grande avanço para o sistema 8080. Kildall passou, então, a escrever outros programas que ajudariam a controlar arquivos no drive usando sua linguagem PL/M. Esse conjunto de programas passou a ser o núcleo do primeiro sistema operacional para microprocessador, o CP/M, que inicialmente significava Control Program/Monitor, mas depois foi rebatizado como Control Program for Microcomputers.

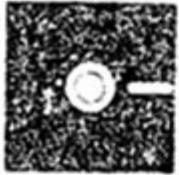
Kildall terminou o CP/M em 1974 e o ofereceu à Intel por 20 mil dólares. A Intel não quis o sistema, mas comprou a linguagem PL/M.

Em 1976, quando saiu de seu emprego na Marinha, Kildall e a esposa abriram uma empresa, a Intergalactic Digital Research (nome depois abreviado para Digital Research Inc., ou DRI). Ainda no mesmo ano, as duas maiores fabricantes de kits com 8080, a MITS e a IMSAI assinaram contratos com a DRI para que pudessem fornecer o CP/M com seus computadores. Dessa forma, o sistema passou a ser amplamente utilizado. Nos anos 1970 e 1980 praticamente qualquer microcomputador com processador 8080 ou Z80 rodaria CP/M. A [Figura 8.25](#) mostra um anúncio do software na revista *InfoWorld* em 1978.

Nos anos 1980, quando a IBM resolveu lançar o PC, com o microprocessador Intel 8088, ela pediu à Microsoft para fornecer o BASIC, e à DRI para fornecer uma nova versão do CP/M – que seria chamada CP/M 86. Entretanto, por uma série de razões, Kildall não fechou contrato com a IBM. Assim, a Microsoft se comprometeu em fornecer tanto o BASIC – que eles já dominavam –, quanto o sistema operacional – coisa que nunca tinham feito. O sistema seria o MS-DOS.

Até 1991, a DRI tentou competir com a Microsoft pelo mercado de sistemas operacionais para microcomputadores, mas sem grandes sucessos. Neste ano, a empresa foi adquirida pela Novell.

Muitas características do CP/M foram repetidas no MS-DOS e ainda hoje são usadas. Por exemplo, programas executáveis seriam arquivos com sufixo “.com”, arquivos de texto teriam sufixo “.txt”. Os drives de disco flexível eram identificados como A: e B:, discos rígidos como C:, D: etc.



**CP/M™**  
**LOW-COST**  
**MICROCOMPUTER**  
**SOFTWARE**

**CP/M™ OPERATING SYSTEM:**

- Editor, Assembler, Debugger and Utilities.
- For 8080, Z80, or Intel MDS.
- For IBM-compatible floppy discs.
- \$100-Diskette and Documentation.
- \$25-Documentation (Set of 6 manuals) only.

**MAC™ MACRO ASSEMBLER:**

- Compatible with new Intel macro standard.
- Complete guide to macro applications.
- \$90-Diskette and Manual.

**SID™ SYMBOLIC DEBUGGER**

- Symbolic memory reference.
- Built-in assembler/disassembler.
- \$75-Diskette and Manual.

**TEX™ TEXT FORMATTER**

- Powerful text formatting capabilities.
- Text prepared using CP/M Editor.
- \$75 Diskette and Manual.

**DIGITAL RESEARCH**  
 P.O. Box 579 • Pacific Grove, CA 93950  
 (408) 649-3896

Figura 8.25: Anúncio do CP/M na revista *InfoWorld* em 1978.<sup>40</sup>

### 8.37 Apple I – 1976

Como visto anteriormente, Steve Jobs e Steve Wozniak estavam concluindo em 1976 o projeto de um microcomputador que ficou conhecido como Apple I.

O projeto iniciou-se a partir de Wozniak, que havia projetado em 1975 um terminal de vídeo para acesso a minicomputadores. Além disso, desde a adolescência ele já havia projetado mais de 50 modelos de computador, mas como não tinha dinheiro para construí-los, eles ficaram apenas no papel. Ele começou a participar de

reuniões do Homebrew Computer Club<sup>41</sup> (HCC) e chegou à conclusão de que poderia criar um microcomputador se conseguisse colocar um microprocessador dentro do seu terminal.

Porém, os microprocessadores disponíveis inicialmente, o Intel 8080 e o Motorola 6800, custavam centenas de dólares. Apenas com o lançamento do MOS 6502 a 25 dólares o projeto deslanchou. Wozniak conseguiu completar o protótipo do computador, soldou a placa a uma pequena TV que ele tinha e levou-o à reunião do HCC, onde reencontrou seu velho amigo Steve Jobs, que se interessou em vender o projeto. Concomitantemente, Wozniak também havia desenvolvido um interpretador BASIC para o computador.

Seu objetivo era ter uma máquina pessoal a qual o usuário pudesse simplesmente ligar e começar a programar, sem burocracia. Ele tinha uma máquina com essas características em seu emprego na Hewlett-Packard, mas o computador custava 10 mil dólares e isso o colocava fora da esfera dos computadores pessoais.

Para iniciar a empresa Apple Computers, Jobs vendeu uma van e Wozniak, uma calculadora HP. Com o dinheiro, eles começaram a construção das placas de computador. O primeiro logo da Apple é mostrado na [Figura 8.26](#), que exibe Isaac Newton sentado sob a famosa macieira.



Figura 8.26: Primeiro logo da Apple Computers.<sup>42</sup>

A Apple Computer Company foi fundada no dia 1º de abril de 1976. Um terceiro sócio, Ronald Wayne, que desenhou o logo, vendeu seus 10% na sociedade por 800 dólares duas semanas depois da fundação porque tinha medo de que a empresa falisse rapidamente. Jobs procurou uma loja de computadores, a The Byte Shop, que se mostrou interessada no projeto e compraria 50 computadores a 500

dólares cada desde que Jobs os entregasse totalmente montados. Jobs procurou a empresa que venderia as peças sem ter nenhum dinheiro e com nada mais nas mãos além do pedido da The Byte Shop. Inicialmente, a empresa ficou preocupada em saber como ele custearia os 20 mil dólares, mas como o pedido da The Byte Shop foi verificado como autêntico, Jobs conseguiu comprar a crédito, com um prazo para pagar em 30 dias.

A partir de então, Jobs, Wozniak e uma pequena equipe passaram a trabalhar ininterruptamente para em 30 dias montar os 50 Apple I e entregá-los em tempo de pagar a dívida. Eles conseguiram fazer isso em 12 dias. Receberam os 25 mil dólares da The Byte Shop e conseguiram pagar os 20 mil dólares de peças e outras despesas.

A [Figura 8.27](#) mostra um Apple I sobrevivente. Ele inicialmente não teria teclado nem gabinete. O teclado e gabinete de madeira mostrados na fotografia possivelmente foram adicionados pelo seu dono.



Figura 8.27: Apple I.<sup>43</sup>

O Apple I tinha uma vantagem em relação aos demais microcomputadores da época: podia ser conectado a uma TV comum que funcionaria como monitor de vídeo. Outros microcomputadores apenas acendiam luzes no painel ou, no máximo, com um custo muitas vezes proibitivo, conseguiam perfurar fitas de papel.

Em relação ao seu sucessor, o Apple II, ele era mais lento e não exibia gráficos, apenas texto. Inicialmente, Wozniak teve problemas com a memória do Apple I. Jobs, então, perguntou porque ele não usava as novas memórias dinâmicas RAM de 16 pinos. Wozniak era um hobbista sem recursos financeiros e normalmente só usava peças que ele conseguia quase de graça. Além disso, não conhecia os fornecedores desses novos circuitos. Jobs fez então mais uma vez seu papel de *businessman* e telefonou para os fornecedores, obtendo algumas amostras das novas memórias com as quais Wozniak pôde trabalhar.

No total, cerca de 200 Apple I foram fabricados em um período de 9 a 10 meses a partir de 1976. O nome Apple, acredita-se, não tinha tido nenhuma motivação especial. Apenas soava bem e eles não conseguiram pensar em um nome melhor. Uma teoria mais interessante diz que escolheram este nome porque Jobs tinha boas lembranças de um verão em que trabalhou em uma fazenda de maçãs e também porque “Apple” vinha alfabeticamente antes de “Atari” na lista telefônica. Na época, a Atari seria uma importante rival de mercado da Apple.

O Apple I era vendido por 666 dólares, um número que, por causa da Bíblia, muitas pessoas associaram com a “marca da besta” ou do demônio do Apocalipse. Como Jobs era ateu, isso provavelmente foi apenas uma brincadeira sem significado esotérico.

## 8.38 Zilog Z80 – 1976

Em 1974, Federico Faggin, o mesmo que projetou os circuitos 4004, 8008 e 8080 da Intel, estava descontente com a forma como a

empresa trabalhava. Uma noite, ele convidou seu colega Ralph Ungermann para tomar uma bebida e disse: “Ralph, eu quero iniciar uma empresa para construir microprocessadores. Você está interessado?” Ao que o outro respondeu: “Sim, vamos fazer isso!” Nascia assim a Zilog, que concorreu seriamente com a Intel na fabricação de microprocessadores a partir de 1976 até meados dos anos 1980.

Porém, eles não tinham o capital necessário para montar a fábrica. Por sorte, a notícia de que dois engenheiros da Intel haviam saído para montar uma empresa de microprocessadores chegou aos ouvidos da Exxon Enterprises, uma subsidiária da petrolífera Exxon Corporation. Um executivo da Exxon ligou para Faggin para marcar uma entrevista com fins de financiar o novo empreendimento. Faggin ainda não tinha a menor ideia do que faria e pediu para o executivo ligar alguns dias mais tarde. Neste meio tempo, ele se concentrou e resolveu que construiria um microprocessador, que ele chamou inicialmente de 2001, e que teria uma capacidade mais ou menos parecida com a do Intel 8048. Faggin começou a adiantar o projeto para mostrar que tinha alguma coisa em andamento, mas na verdade não possuía praticamente nada. Ele afirma ter tido bastante sorte em receber o interesse da Exxon, pois o mercado financeiro na metade dos anos 1970 sofria uma depressão, e investimentos em tecnologia não eram fartos.

Na virada de 1975 para 1976, após mais algum trabalho, ele resolveu renomear o novo circuito de “Super 80”, e depois “Z80”. “Z” porque é a última letra do alfabeto, e esse processador seria a última palavra em processadores. Na [Figura 8.28](#) vemos um dos primeiros Z80 fabricados.

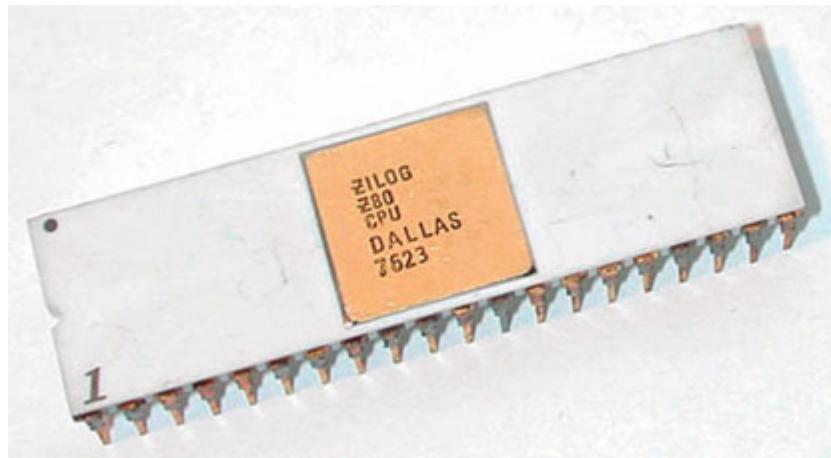


Figura 8.28: Zilog Z-80.<sup>44</sup>

Uma das desvantagens do 8080 que o Z80 iria resolver era a necessidade de três diferentes voltagens nos pinos; no caso do Z80 seria apenas uma. Além disso, o sistema de interrupções precisava ser aprimorado e o Z80 teria um conjunto de instruções maior do que o 8080, embora fosse compatível com ele em código binário. Ou seja, código de máquina do 8080 poderia rodar no Z80 sem adaptação, mas o inverso não seria necessariamente possível. O Z80 teria dentre essas novas instruções, poderosas instruções de bloco, capazes de mover blocos inteiros de memória de uma para outra posição com uma única instrução de máquina. Além disso, o Z80 teria pelo menos o dobro de registradores do que o 8080.

O Z80 superou o 8080 e seu sucessor, o 8085, no mercado de microprocessadores e, por muitos anos, foi o microprocessador mais vendido no mundo. Assim como o 6502, ele foi pensado para aplicações industriais e embarcadas, não para microcomputadores, mas acabou sendo usado em vários modelos populares da época como os microcomputadores TRS-80, Sinclair, Amstrad e Commodore 128. Outros computadores, como o Apple II, permitiam que uma placa com o Z80 fosse adaptada ao computador que assim, contaria com dois processadores: o 6502 original e o Z80, que rodaria o sistema operacional CP/M.

O Z80 ainda é largamente usado hoje, pois é um processador completo de 8 bits, barato e relativamente eficiente, com uma arquitetura bem conhecida. Seu uso, porém, não ocorre mais em microcomputadores, mas em dispositivos embutidos e microcontroladores, como seus criadores originalmente imaginaram.

### 8.39 Sol 20 – 1976

Outra *startup* de rápida ascensão e queda na década de 1970 foi a Processor Technology Corporation. Ela foi fundada em 1975 por Gary Ingram e Bob Marsh em Berkeley, na Califórnia. Seu principal produto era um microcomputador chamado Sol 20, que passou a ser vendido em 1976. Ele era construído com o processador 8080 e era compatível com o Altair 8800. O Sol 20 foi projetado por Lee Felsenstein (Estados Unidos, 1945), Gordon French (um dos fundadores do Homebrew Computer Club) e Bob Marsh a pedido de Les Solomon, editor da *Popular Electronics*.



**Figura 8.29: Sol 20.<sup>45</sup>**

O computador era vendido em kit de componentes ou completamente montado em um gabinete horizontal pintado em “azul IBM”. Infelizmente, depois do Sol 20 a empresa não conseguiu produzir uma nova geração de computadores e, após vender cerca de 10 mil cópias de sua máquina, ela fechou as portas em 1979.

O site <http://www.sol20.org/> é um verdadeiro repositório de informações e recursos sobre o Sol 20. Inclusive, vários programas estão disponíveis ali. A [Figura 8.30](#) apresenta a tela de um programa de xadrez em modo texto implementado no computador.



**Figura 8.30: Tela de um programa de xadrez para o Sol 20.<sup>46</sup>**

## **8.40 Compucolor 8001, o Primeiro Computador a Cores – 1976**

O ano de 1976 foi bastante pródigo para a indústria de microcomputadores. Neste ano, também foi lançado o primeiro microcomputador capaz de processar e exibir imagens coloridas, o Compucolor I também conhecido como Compucolor 8001 ([Figura](#)

[8.31](#)), produzido pela Intelligent Systems Corporation (ISC), fundada em 1973. Ele foi projetado por Charles A. Muench.



Figura 8.31: Compucolor 8001.<sup>47</sup>

O Compucolor na verdade evoluiu a partir de um terminal colorido chamado Intercolor 8001. Esse terminal usava uma interface RS-232 para se ligar a um mainframe, custava 1.395 dólares e vinha na forma de um kit a ser montado, incluindo um tubo de imagem CRT de 19 polegadas.

Com um investimento de mais 1.295 dólares, o terminal podia ser transformado em um computador *standalone*, ou seja, ele não necessitava estar ligado a um mainframe. Esse computador era capaz de rodar a linguagem BASIC.

Uma de suas características singulares era um teclado ótico. Em vez de usar contatos elétricos, as teclas, quando pressionadas, bloqueavam raios de luz produzidas por LEDs e geravam padrões de sombra em receptores ópticos que eram interpretados como entrada de caracteres.

## 8.41 Modem 80-103A – 1977

O equipamento usado para permitir comunicação entre computadores através de rede telefônica é denominado “Modem”. Até 1977 apenas computadores de grande porte tinham esse privilégio. Contudo, neste ano, Dale Heatherington e Dennis Hayes (Estados Unidos, 1950) criaram o primeiro modem para computadores pessoais, o 80-103A, que funcionava acoplado a um microcomputador IMSAI 8080. A velocidade desse modem era de apenas 300 bauds (oscilações por segundo).



Figura 8.32: Dale Heatherington e a placa do 80-103A sobre o IMSAI 8080.<sup>48</sup>

Os primeiros modems eram implementados como placas inseridas em slots na placa mãe do microcomputador e, a partir dali, podiam ser controladas diretamente por programas na memória do computador. O problema é que cada fabricante produzia slots com configurações diferentes, logo, seria necessário produzir um modem diferente para cada tipo de computador. Pior ainda era o caso de computadores que não implementavam slots internos ou que tinham poucos deles, os quais por vezes acabavam ocupados por outros equipamentos.

Assim, eles criaram um modem externo, que se comunicava de um lado com a rede telefônica e, de outro, com o computador através de uma interface RS-232.

Outra inovação foi o *smartmodem* ([Figura 8.33](#)). Até aquela época havia basicamente dois tipos de modem: os do mainframe e os dos terminais. O modem do mainframe basicamente atendia as ligações e fazia a conexão com o computador. Já os modems dos terminais usualmente não eram capazes de discar. Um ser humano precisava estabelecer a ligação telefônica manualmente e depois colocava um adaptador acústico no fone para que os computadores pudessem se comunicar. Com o smartmodem o computador podia ele mesmo dar os comandos de discagem e controlar toda a transmissão de dados.



**Figura 8.33: Smartmodem externo de Hayes.**<sup>49</sup>

Na verdade, fazer o modem discar nem era tão difícil – mais complicado era fazer ele entender que a sequência de informações sendo transmitida havia acabado. Um smartmodem tinha dois estados: um no qual ele interpretava os sinais vindos do computador como dados a serem transmitidos e outro no qual ele devia interpretar os dados vindos do computador como comandos a serem executados, como por exemplo, o comando para desconectar a ligação.

A dificuldade então era descobrir como indicar ao modem durante o estado de transmissão de dados que ele devia mudar de estado;

como fazer ele executar um comando se ele não estava no estado de receber comandos? Uma solução bastante óbvia, mas infelizmente impraticável, seria usar um dos pinos da conexão RS-232 para indicar a mudança de estado. Por exemplo, quando esse pino recebesse um pulso elétrico o modem mudaria de um estado para outro. Os dados e comandos seriam então enviados através dos outros pinos.

Porém, a RS-232 usualmente implementada em microcomputadores tinha apenas nove pinos e em muitos destes alguns pinos não eram sequer conectados. Havia, assim, pouquíssimos pinos realmente úteis para transmitir dados e gastar um deles apenas para indicar a mudança de estado não seria uma boa prática.

Eles decidiram, então, utilizar uma sequência de caracteres pouco usual para que durante o estado de transmissão o modem entendesse que devia mudar para modo de comando. A sequência escolhida foi “+++”. Porém, para evitar que uma transmissão acidental de +++ fizesse o modem trocar de estado inadvertidamente, o circuito só faria a mudança se, após a transmissão da sequência, houvesse um segundo de silêncio. Esse segundo foi chamado de “tempo de guarda”. Somente após esse tempo haveria a mudança de estado.

## 8.42 DEC VAX 11 – 1977

O sucesso da família de computadores PDP da DEC levou a uma evolução em 1977, uma nova família de computadores e sistema operacional que foram inspirados no PDP-11, mas cujo projeto foi refeito desde o início. Trata-se da série VAX 11 (Virtual Address eXtension), um computador de 32 bits capaz de processar um milhão de instruções por segundo. Era uma máquina confiável, amigável e poderosa, e essas características, aliadas a um preço acessível, fizeram com que fosse o computador preferido das universidades na época. A [Figura 8.34](#) mostra a parte frontal de um VAX 11/780, o primeiro modelo da série que foi fabricada até 1988.



Figura 8.34: Vax 11/780.<sup>50</sup>

Por muitos anos, o VAX 11/780 foi o modelo de comparação de velocidade de computadores. Em um benchmark, um computador como o Vax 8700 de 1986 tinha sua velocidade avaliada como 5 em relação à velocidade do VAX 11/780, significando que era cinco vezes mais rápido do que 780.

## 8.43 Apple II – 1977

Apenas um ano após o lançamento do Apple I, Wozniak e Jobs lançavam seu segundo modelo, o Apple II, ou Apple ][ ou ainda Apple //. O design inicial do Apple I era limitado principalmente pelo fato de que, na época, Jobs e Wozniak tinham pouco dinheiro para investir. Porém, com os lucros das vendas do primeiro modelo puderam projetar e construir um computador muito melhor.

O Apple II usava o mesmo circuito MOS 6502, mas vinha em um elegante gabinete plástico que era integrado a um teclado e que suportava a colocação de um monitor sobre ele. Pouco mais tarde a Apple lançou também o Apple Disk II, um drive de disquete de baixíssimo custo para a época. A [Figura 8.35](#) mostra um Apple II de 1977 com dois drives de disquete e um monitor de 1980.



**Figura 8.35: Apple II.<sup>51</sup>**

O Apple II foi um grande sucesso. Seu projeto foi pouquíssimo modificado durante os 17 anos em que foi produzido e vendido. Ele permitia produzir gráficos em cores e a maçã colorida que aparecia no logotipo logo acima do teclado era uma referência à capacidade desse computador de produzir cores em um televisor ou monitor apto.

O Apple também tinha um alto-falante embutido, mas nenhum circuito especial para sintetizar som. Havia um único comando que produzia um “click” no alto-falante. Assim, sons musicais e até polifonias de quatro vozes eram produzidas por software

simplesmente gerando uma quantidade suficiente de clicks na velocidade correta.

A máquina tinha slots na placa-mãe que permitiam conectá-lo a drives de disco, modem, expansão de memória e placa CP/M, ou seja, uma placa com um processador Z80 que permitia ao Apple rodar programas do sistema operacional CP/M, como o editor de texto WordStar, o banco de dados dBASE II e a planilha VisiCalc. Porém, esse processador não dava acesso direto às capacidades gráficas do 6502, ficando restrito ao modo texto.

O drive de disquete do Apple II não tinha um sensor que permitisse ao mecanismo saber em que trilha o cabeçote estava. Havia um limitador físico do cabeçote na trilha 0 que não o deixava passar deste ponto. Assim, o comando que levava o cabeçote para a trilha 0 consistia em mover o cabeçote 40 vezes (havia 40 trilhas) para a anterior. Cada vez que o cabeçote estava na trilha 0 e tentava ir para a anterior, o limitador físico produzia um som alto parecido com “PLEC!”. Assim, se o cabeçote estivesse na trilha 1 e fosse enviado para a trilha 0, as pessoas próximas ouviriam uma sequência rápida de 39 PLECs, o que soava como se a máquina estava se destruindo. Enviar o drive para a trilha 0 era um expediente muito popular para assustar usuários novatos do Apple.

O Apple II foi um dos primeiros microcomputadores a se tornar popular no Brasil. Porém, como até 1992 a importação de equipamentos de informática era difícil por aqui, uma indústria nacional de clones do Apple II foi criada. Entre mais de 20 modelos havia o Elppa (Apple de trás para frente), o Maxtro, o AP II e o Milmar Apple II Plus. Outro clone bastante popular nos anos 1980 no Brasil foi o TK 3000 que era clone do Apple IIe, cujos circuitos não podiam ser copiados e que, logo, foi produzido por engenharia reversa.

## 8.44 Tandy TRS80 – 1977

Outro importante microcomputador lançado no ano 1977 foi o TRS-80 ([Figura 8.36](#)). Ele era fabricado pela Tandy Corporation e vendido através de suas lojas, as RadioShack. Dessa maneira, TRS-80 era uma abreviação de Tandy/RadioShack Z-80 Microprocessor.

Em meados dos anos 1970, um funcionário da Tandy, Don French, comprou um kit do Altair e interessou-se muito pela ideia. Ele projetou um computador próprio e apresentou a seus superiores. O esquema não impressionou, mas a proposta de vender computadores, sim. Infelizmente, nem todos na alta direção tinham a mesma ideia e ela foi ferozmente combatida. As 3 mil lojas RadioShack tinham 11 milhões de clientes que potencialmente poderiam comprar um computador, mas o preço era muito alto visto que a média de um produto RadioShack era de apenas 30 dólares.



**Figura 8.36: TRS-80.<sup>52</sup>**

Porém, quando as vendas dos principais produtos começaram a cair, os executivos tiveram que arriscar em um novo projeto e,

enfatizando a questão do preço baixo, o TRS-80 começa a tomar vida.

Em fevereiro de 1977, a equipe apresenta um protótipo ao presidente da empresa, Charles D. Tandy (Estados Unidos, 1918-1978), o qual rodava um programa de contabilidade escrito em BASIC. Tandy rapidamente fez o software travar porque inseriu no sistema o seu próprio salário anual de 150 mil dólares, valor que o pequeno computador não conseguia tratar. Com isso em mente, a equipe adicionou no projeto um tratador de números de ponto flutuante para lidar com estes casos.

Em agosto do mesmo ano, após um investimento equivalente a um ano de salário de Tandy, o TRS-80 é colocado à venda por apenas 399 dólares, cerca de quatro vezes menos do que um Apple II.

Uma conferência de imprensa no Hotel Warwick em Nova York lançou o TRS-80 em 3 de agosto de 1977. Infelizmente, ela não teve muita audiência, pois no mesmo dia um grupo terrorista fez um ataque na cidade, matando uma pessoa e provocando a evacuação de mais de 100 mil. Apesar disso, as vendas dispararam rapidamente e chegou a existir uma fila de espera para receber o computador, na qual cada pessoa fazia um depósito de 100 dólares para entrar.

Em 1979, o TRS-80 era o microcomputador com a maior quantidade de software disponível, e até 1982 ele era o campeão de vendas entre os microcomputadores.

No Brasil, vários clones do TRS-80 foram produzidos. Entre eles, o Dismac D8000, os Prológica CP-300 e CP-500 e o Digitus DGT-100. Já o CP-400 era clone do TRS-80 Color Computer.

## 8.45 Commodore PET – 1977

Outra empresa que lançou um microcomputador em 1977 foi a Commodore, fundada por Jack Tramiel (Polônia, 1928-2012). O modelo conhecido como PET ([Figura 8.37](#)) foi apresentado em janeiro, antes do TRS-80 e do Apple II, mas só começou a ser

vendido depois deles. O PET, o TRS-80 e o Apple II são conhecidos como a “Trindade de 1977”, que disputam o título de primeiro computador pessoal comercial de sucesso.



Figura 8.37: Commodore PET.<sup>53</sup>

Em relação aos seus competidores, o PET tinha algumas novidades. Entre elas, um teclado numérico separado, para facilitar digitação de números, a capacidade para trabalhar com letras maiúsculas e minúsculas e um sistema operacional gravado em ROM em vez de em software.

Uma de suas características marcantes era o “teclado chiclete”, com teclas macias, semelhantes a chicletes, bem diferente dos teclados clássicos de computador. Na verdade, esse era um dos pontos fracos, porque as marcações das teclas costumavam apagar rapidamente com o uso. Outra característica era o gravador de fita K7 embutido no gabinete para gravação e leitura de programas e dados.

O PET era baseado no microprocessador MOS 6502. De fato, a Commodore comprou a empresa MOS Technology e absorveu todos os seus funcionários, incluindo Chuck Peedle (Estados Unidos, 1937), criador do 6502, o qual acabou como projetista do PET, projeto que já estava em desenvolvimento pela MOS. Consta que em 1976 Peedle procurou a RadioShack na esperança de obter patrocínio e estrutura de vendas para o computador que ele estava projetando, mas não obteve sucesso, pois na época a empresa ainda não estava interessada neste mercado. Ainda, antes de comprar a MOS, a RadioShack procurou Steve Jobs e Steve Wozniak, mas a proposta deles foi considerada muito cara.

Também não há consenso sobre a origem do nome PET, mas o mais provável é que significasse Personal Electronic Transactor.

A empresa infelizmente tinha pouca capacidade de produção. Apenas 30 máquinas podiam ser construídas por dia, bem abaixo da demanda. No primeiro ano de fabricação apenas 500 computadores saíram das linhas de montagem.

O PET foi sucedido por um computador muito barato chamado Vic-20 e em 1982 pelo Commodore 64, que era celebrado por sua memória de 64 kB – embora apenas 38 estivessem disponíveis para o BASIC residente. O 64 usava tecnologia MOS especial para sons e gráficos, e assim, não havia competidor à altura de seus 595 dólares. Tramiel buscou, inclusive, fazer uma guerra de preços e em certos momentos baixou o preço do Commodore 64 para 199 dólares.

Após faturar um bilhão de dólares até 1983, Tramiel deixa a empresa e compra parte da Atari, tornando-se assim competidor da Commodore. Porém, quem acabou com a Commodore de fato foi a IBM, com o lançamento do PC. Em 1994, a empresa precisou declarar falência.

## **8.46 Console de Videogame Atari 2600 – 1977**

Em 1975 a Atari tinha um conjunto de jogos Arcade de sucesso e um console para uso doméstico que consistia em um único jogo: o *Pong*. Nessa época, a empresa desejava de alguma forma levar aos lares todos os seus jogos. O resultado foi o console Atari VCS, depois rebatizado como Atari 2600.

Esse projeto se tornou viável depois do lançamento do MOS 6502, um microprocessador barato que acabou sendo usado nos consoles também. Além disso, a possibilidade de gravar os jogos em ROM na forma de cartuchos baratos e facilmente intercambiáveis foi a solução para disponibilizá-los a um preço acessível ao consumidor.

Porém, a Atari não tinha recursos suficientes para produzir o console e, imaginando que seria um grande sucesso, Bushnell vendeu a empresa para a Warner em 1976 de modo a viabilizar a produção. Assim, quando o console Atari foi lançado em 1977, ele já era um produto da Warner, embora pouca gente realmente soubesse disso.

Levou bastante tempo para as pessoas perceberem que o console permitia jogar outras coisas além de *Pong*, e na época todos já estavam bastante saturados do jogo. Em 1977 e 1978, ele vendeu apenas metade das unidades fabricadas, algo por volta de 750 mil.

Em 1978 houve atritos entre Bushnell e os novos executivos da Warner. Em resumo, eles não ficavam muito à vontade com as festas regadas a drogas e bebidas e com o código de vestir totalmente relaxado dos funcionários da Atari. Após vários desentendimentos e uma briga em frente dos diretores, Bushnell finalmente conseguiu ser demitido da empresa que criou.



Figura 8.38: Atari 2600.<sup>54</sup>

Em 1979, com um novo CEO, Ray Kassar (Estados Unidos, 1928), a Atari conseguiu emplacar um novo sucesso: *Space Invaders*, inicialmente em forma de Arcade. Esse jogo fez com que o interesse por videogames voltasse a crescer e a sua versão em cartucho levou as vendas do Atari 2600 para 2 milhões de unidades em 1980.

A empresa se tornou líder absoluta em videogames e mesmo sinônimo dessa palavra: você compraria um “Atari”, não um videogame. Porém, ela não dava crédito adequado aos desenvolvedores de jogos, e alguns destes, irritados, criaram outra empresa, na qual esperavam desenvolver games melhores que os da Atari e que rodariam no mesmo console.

O grande crescimento da Atari no início dos anos 1980 foi também o responsável por seu fim. Como veremos adiante, problemas de gerenciamento levaram a empresa à bancarrota em um piscar de olhos.

## 8.47 Matlab – 1978

Enquanto a computação pessoal dava seus primeiros passos, a computação científica também não parava. Grandes mainframes ainda eram usados por físicos e engenheiros para resolver problemas cada vez mais sofisticados em suas áreas de pesquisa.

Só que usar um computador para fazer cálculos não era tão fácil quanto às vezes parece. Essas máquinas, com base binária e representação finita de números, de vez em quando produzem resultados imprevisíveis.

Um fato bastante conhecido, entre muitos outros, é que o número decimal 0,1 não tem representação finita em binário. Em base binária, ele corresponde a uma dízima periódica: 0,0001100110011001100110011001100.... Assim, a representação de 0,1 em binário é infinita. O fato é que computadores apenas representam números finitos e, dessa forma, o número representado internamente deverá ser um pouquinho menor do que 0,1, dependendo do número de bits da palavra que o computador usa em sua memória. Por exemplo, 0,1 poderia na verdade ser representado internamente como 0,0999999999999999.

Se você apenas representa o número dessa maneira pode até ignorar essa pequena diferença, mas quando você faz uma série de cálculos com números assim, o pequeno erro vai se propagar. Se você somar um milhão de vezes 0,1, com esse pequeno erro, não vai obter 100 mil, como esperaria, mas um número significativamente menor, dependendo do tamanho da palavra usada em seu computador.

Esse é apenas um dos muitos problemas numéricos com os quais os engenheiros e físicos se deparam quando usam essas máquinas para cálculos científicos. A grande maioria dos problemas conhecidos pode ser resolvida pela aplicação de uma ou outra técnica, mas isso implica em escrever programas, frequentemente complexos, e implica em escrevê-los muitas e muitas vezes.

Em função destes problemas, surgiram iniciativas de desenvolvimento de bibliotecas de rotinas científicas que pudessem

ser definidas apenas uma vez e então reusadas. Por exemplo, se você quisesse um programa para inverter matrizes, não precisaria escrever um – bastaria buscá-lo na tal biblioteca, visto que é uma operação bastante comum em aplicações científicas.

Clever Moler (Estados Unidos, 1939) trabalhou em uma dessas iniciativas. No início dos anos 1970, ele desenvolveu bibliotecas de rotinas para computar valores associados a matrizes, a EISPACK, e para resolver equações lineares, a LINPACK. Com essa experiência e o conhecimento adquirido a partir do livro *Algorithms + Data Structures = Programs* de Niklaus Wirth, ele desenvolveu em 1978 uma extensa biblioteca de rotinas científicas em FORTRAN, que passou a ser conhecida como Matlab, ou Matrix Laboratory.

A biblioteca era composta inicialmente por 80 rotinas científicas e trabalhava com um único tipo de dados: a matriz de números complexos de precisão dupla (números de precisão dupla são números representados com o uso de duas palavras do computador; assim, se o computador tem 16 bits, o número de precisão dupla terá 32).

A biblioteca logo ganhou a possibilidade de impressão de gráficos para suas funções, mas como poucas máquinas de grande porte na época trabalhavam com terminais gráficos, estes eram simulados de forma grosseira com a impressão de asteriscos em uma tela de texto.

Em 1979, Moler ministrou uma disciplina na Universidade de Stanford e usou Matlab com seus alunos. Ele comenta que metade da turma era formada por matemáticos e cientistas da computação e que não viram muito valor para a ferramenta. Porém, a outra metade da turma era formada por engenheiros e acharam que ela tinha um grande potencial para ajudá-los em seus trabalhos. Assim, vários destes alunos, após concluírem a universidade, foram absorvidos por empresas e centros de pesquisa onde passaram a desenvolver extensões para a biblioteca original, muitas das quais começaram a ser oferecidas como produtos comerciais.

Quando em 1981 a IBM anunciou o lançamento do IBM-PC, Jack Little, um desenvolvedor de extensões do Matlab, juntamente com seu colega Steve Bangert, percebeu a possibilidade de usar a ferramenta nesses novos computadores. Eles reescreveram o código do Matlab na linguagem C e adicionaram novos gráficos poderosos e outras ferramentas para facilitar o seu uso. Estes dois e o criador original, Moler, fundaram em 1984 a empresa MathWorks, que transformou o PC Matlab em um produto comercial de grande sucesso, considerando que em 2006 contava com mais de 1 milhão de usuários. O logotipo da empresa, mostrado na [Figura 8.39](#) corresponde a um gráfico de alta resolução gerado por uma versão moderna do Matlab, o qual mostra uma aproximação numérica do modo de vibração de um tipo de membrana que Moler estudou durante seu doutorado em Stanford no ano de 1965.

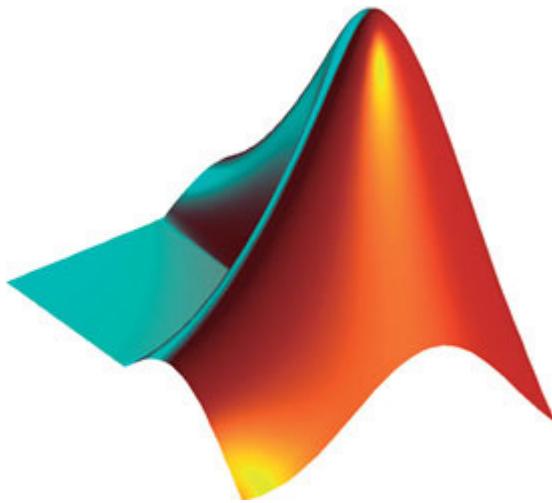


Figura 8.39: Logo da MathWorks.<sup>55</sup>

## 8.48 Intel 8086 – 1978

O Intel 8086 foi um microprocessador projetado entre 1976 e 1978 e trabalhava com 16 bits. Ele foi a base para a segunda grande onda dos microprocessadores, sendo consideravelmente superior ao MOS 6502, Motorola 6800 e Intel 8080, que até então eram os mais

usados no mercado. Uma variação dele, o 8088, foi a base para o primeiro computador pessoal da IBM, o IBM-PC. Além disso, sua arquitetura foi a base para toda uma família de microprocessadores, muitos dos quais compartilharam parte de seu nome, como 80286, 80386 e 80486. A partir do processador que seria o 80586, a Intel passou a dar nomes próprios aos seus processadores. O 80586, no caso, foi chamado de Pentium. Diz-se que isso aconteceu porque a Intel não poderia usar um simples número como marca registrada.

Um dos motivos para este projeto foi a ameaça à hegemonia da Intel por parte da Zilog, já que seu processador Z-80 estava se tornando o líder de vendas no mundo dos microcomputadores. O 8086 foi projetado para ser compatível em termos de comandos de máquina com o 8008, 8080 e 8085, seu antecessor direto. Porém, se ele rodasse código desses processadores anteriores, não seria necessariamente o mais eficiente possível. O importante, no primeiro momento, era que os programas抗igos continuariam rodando.

Novas instruções foram adicionadas ao 8086, muitas delas inspiradas no Z80. Segundo o arquiteto-chefe do projeto Stephen P. Morse (Estados Unidos, 1940), o projeto do novo processador também foi muito influenciado pelas linguagens de programação de alto nível; ele foi projetado para dar suporte direto a algumas estruturas de linguagens aninhadas como Pascal e ALGOL.

A [Figura 8.40](#) mostra uma foto do processador miniaturizado. Ele tinha cerca de 29 mil transistores em um chip de apenas 3,3 centímetros.

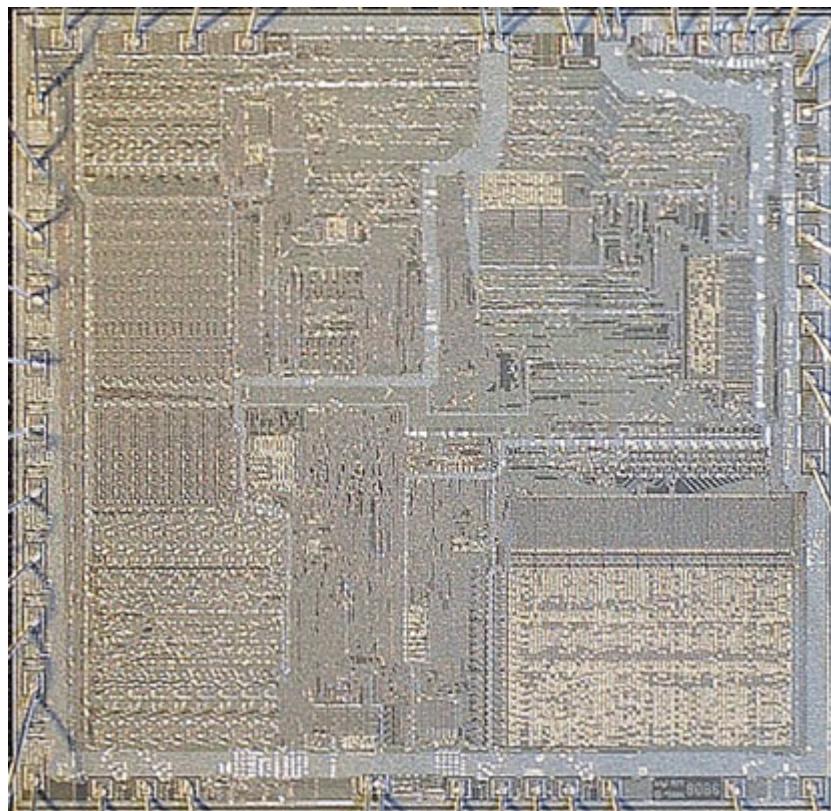


Figura 8.40: Intel 8086.<sup>56</sup>

O 8086 acabou definindo um padrão da indústria que ainda tem reflexos hoje. Quando foi iniciado, contudo, a Intel sequer desconfiava disso. Na verdade, quando o Z-80 começou a dominar o mercado, praticamente todos os processadores existentes para microcomputadores eram de 8 bits. A Intel foi ousada ao iniciar o projeto de um processador de 32 bits, o 8800, mas encontrou grandes problemas técnicos para construí-lo com a tecnologia de circuitos integrados da época e ele acabou fracassando. Entretanto, a ameaça do Z-80 precisava de resposta e a empresa acabou apostando em Morse, na época um jovem engenheiro de 36 anos, que a impressionou com suas críticas pertinentes às falhas do 8800. Assim, a Intel apontou Morse como projetista solo do 8086. Morse comenta que se a empresa tivesse noção do que esse projeto representaria e o impacto que ele teria ao longo de tantas décadas, jamais o teria atribuído a um único engenheiro.

Outro ponto surpreendente foi o fato de que Morse era engenheiro de software e não engenheiro elétrico. Muito provavelmente foi por isso que o 8086 passou a ter uma visão mais voltada a dar suporte às necessidades das linguagens de programação e não apenas em implementar este ou aquele circuito.

Porém, o 8086 não foi um sucesso logo de início. A indústria de microcomputadores do final dos anos 1970 estava satisfeita com o Z80 e 6502. No começo, o 8086 foi usado em microcontroladores e merece destaque seu uso também pela NASA no projeto dos ônibus espaciais: o 8086 era usado para controlar testes de diagnóstico nos foguetes de combustível sólido.

Morse deixou a Intel antes que seu processador se tornasse o ícone que é hoje. Algumas semanas depois de sua saída, a empresa lançou o 8088, que era uma versão, segundo Morse, “castrada” do 8086. Ocorre que na época a maioria dos sistemas de microcomputadores eram baseados em 8 bits e o 8086 era um processador de 16 bits. Assim, o 8088 foi uma versão modificada do 8086 para se tornar compatível com hardware de 8 bits, perdendo performance, mas ganhando compatibilidade.

Em 1981, a IBM resolveu construir seu próprio microcomputador com partes adquiridas fora da empresa. Havia três escolhas de processador de 16 bits, que era o objetivo da IBM, o Motorola 68000, que acabou sendo o microprocessador do Apple Macintosh, e os dois processadores da Intel, o 8086 e o 8088.

Como estava mais acostumada a trabalhar com tecnologia Intel, a IBM preferiu o 8088. Entretanto, como o 8088 e o 8086 compartilhavam o mesmo código de máquina, este passou a ser o padrão da indústria dali em diante.

Comenta-se que o sucesso do processador ocorreu por seus méritos, sim, mas principalmente também pelo fato de ter sido utilizado em uma máquina IBM, que logo alcançou grande popularidade. Diz-se que uma das perguntas-chave na época era: “Você compraria um microcomputador da International Business

Machines, uma empresa centenária, ou de uma empresa de garagem com o nome de uma fruta?"

Um fato impressionante em relação a esse processador e todos os seus sucessores é que um código escrito para o 8086 em 1978 ainda rodaria hoje nos processadores Intel mais modernos, como por exemplo um Core 2. A única diferença é que hoje esse código rodaria cerca de 180 mil vezes mais rápido.

## 8.49 Xerox NoteTaker – 1978

Em 1976, Douglas Fairbairn, que trabalhou no projeto do Xerox Alto, resolveu implementar alguns dos conceitos do Dynabook de Alan Kay. Também participaram da equipe Adele Goldberg e Larry Tesler (Estados Unidos, 1945).

Na época, ainda não havia tecnologia para fazer um Dynabook, que seria praticamente um tablet, mas Fairbairn quis implementar aquilo que fosse possível com a tecnologia da época.

Assim nasceu o Xerox NoteTaker, um computador portátil de 22 kg, que não passou de um protótipo, pois menos de 10 unidades foram construídas. Porém, deixou influências no design de outros computadores portáteis que vieram mais tarde. Ele tinha um monitor monocromático embutido no gabinete com tela sensível ao toque, drive de disquete de 340 kB e um mouse. Seu sistema operacional era o ambiente gráfico Smalltalk 78, que tinha sido implementado para o Xerox Alto.

O computador era montado de forma a ser facilmente transportado. O teclado podia ser colocado sobre o console, tampando, dessa forma, o monitor e o drive. Ou seja, o NoteTaker se fechava no formato de uma maleta equivalente em tamanho a uma máquina de costura portátil.

Era uma máquina relativamente avançada, pois já utilizada o Intel 8086, tinha uma memória RAM de 128 a 256 mil palavras de 16 bits. O monitor era da resolução que seria praticamente padrão por muitos anos em computadores pessoais: 640 colunas por 480

linhas. Ele também contava com modem, alto-falante e uma interface Ethernet para rede local. Estima-se que caso fosse vendido, seu preço chegaria a 50 mil dólares.

## 8.50 Planilha Eletrônica VisiCalc – 1978

Em 1978, um estudante de Harvard chamado Daniel S. Bricklin (Estados Unidos, 1951) teve a ideia e desenvolveu um aplicativo que seria central na história da computação: o VisiCalc (Visible Calculator). Até o desenvolvimento dessa verdadeira planilha eletrônica, microcomputadores eram apenas brinquedos para curiosos. Porém, com o VisiCalc o microcomputador que até então apenas estava nas casas, chegou também aos escritórios.

Ele foi um dos responsáveis pelo grande aumento das vendas do Apple II e pelo enriquecimento de Jobs e Wozniak. Como o VisiCalc foi inicialmente desenvolvido para o 6502, isso fez do Apple II o primeiro microcomputador com um aplicativo realmente útil nas empresas. Comenta-se que foi este software que fez a IBM abrir os olhos para o mercado de microcomputadores, o que culminou no lançamento do PC, três anos mais tarde.

Inicialmente Bricklin desenvolveu o VisiCalc em Basic em um final de semana, mas logo contratou um programador mais experiente, Bob Frankston (Estados Unidos, 1949), para gerar uma versão bem mais eficiente em Assembly. O VisiCalc, entretanto, necessitava de pelo menos 20 kB de memória para rodar. A maioria dos Apple II vendidos na época tinha apenas 16 kB, mas havia uma versão bem mais cara com memória expandida para 32 kB. Para quem tinha uma empresa, a escolha entre uma máquina IBM ou um pequeno Apple com 32 kB era óbvia. As vendas desse tipo de Apple dispararam, juntamente com o VisiCalc.

Uma das características que seria fundamental para uma planilha seria a possibilidade de navegar pela matriz bidimensional de valores ou “células”. Inicialmente, Bricklin pensou em usar o Joystick, recém desenvolvido para o Apple II para movimentar o

cursor pelas células e o botão “fire” para iniciar a edição de uma célula. Entretanto, o dispositivo se mostrou muito impreciso e ele resolveu optar então pelas teclas de setas. Infelizmente, o Apple só tinha duas setas (direita e esquerda) e não as quatro que são mais comuns hoje. Assim, ele definiu a barra de espaço como um controle que alterava as setas entre ir para direita e esquerda ou ir para cima e para baixo.

A [Figura 8.41](#) mostra uma tela do VisiCalc rodando em um Apple II.

C11 <L> TOTAL			
	A	B	C
1	ITEM	NO.	UNIT
2	---	---	---
3	MUCK RAKE	43	12.95
4	BUZZ CUT	15	6.75
5	TOE TONER	250	49.95
6	EYE SNUFF	2	4.95
7			-----
8			SUBTOTAL
9			13155.50
10			9.75% TAX
11			1282.66
12			-----
13			TOTAL
14			14438.16
15			
16			
17			
18			
19			
20			

Figura 8.41: Tela do VisiCalc rodando em um Apple II.<sup>57</sup>

Em janeiro de 1979, Bricklin e Frankston fundaram a empresa Software Arts Corporation para desenvolver e manter o VisiCalc. Ao mesmo tempo, a empresa pessoal de Dan Flystra, que emprestou o Apple para Bricklin desenvolver o primeiro protótipo do VisiCalc, começou a fazer propaganda para vender o produto. Essa empresa foi depois rebatizada como VisiCorp.

O produto era vendido por 100 dólares, possivelmente caro para um usuário doméstico, mas irrisório para a maioria das empresas. Em 1981 a Software Arts já havia recebido 12 milhões de dólares em royalties pelas vendas do VisiCalc.

Porém, nos anos 1980, clones começaram a aparecer. Em 1983 o Lotus 1-2-3, para o IBM-PC, foi escrito por um ex-funcionário da Software Arts e usava com mais eficiência os recursos do PC e permitia importar arquivos do VisiCalc, facilitando a migração. O Lotus rapidamente bateu o VisiCalc em vendas.

Também em 1980 a Microsoft lançou uma planilha, a MultiPlan, rebatizada em 1985 como Microsoft Excel.

Por outro lado, a Software Arts não deu conta de desenvolver uma versão do VisiCalc para PC, o que até provocou um processo judicial por parte da VisiCorp, que viu suas vendas baixarem assustadoramente. Finalmente, o que sobrou das ações da Software Arts foi vendido para a Lotus, que imediatamente cessou a produção do VisiCalc.

## **8.51 SBC - Sociedade Brasileira de Computação – 1978**

Até a construção do Pato Feio, o Brasil apenas importava tecnologia de computação de outros países, principalmente Estados Unidos e Inglaterra. A partir desse projeto, como já vimos, iniciou um movimento nacional pela construção de uma indústria própria. Departamentos de engenharia, física e matemática em várias universidades nacionais começaram, com incentivo do governo, a enviar professores para o exterior, para as grandes universidades americanas e europeias, para fazerem seus mestrados e doutorados relacionados à computação.

Em meados dos anos 1970 começaram a surgir cursos de graduação e pós-graduação em computação no Brasil. Pouquíssimos foram criados nos anos 1960. Entre eles, destaca-se o curso de Pós-Graduação em Informática, na PUC-RIO, que introduziu o neologismo “informática” na língua portuguesa ao preferir o termo francês “*Informatique*” em vez do “*Computer Science*” do inglês. Como era uma época de desbravamento, os professores ensinavam uns aos outros sobre a nova ciência e

procuravam aprender juntos. Chegou a acontecer um fato curioso de um aluno defender seu mestrado antes do seu próprio orientador defender o dele.

Com essa nova massa crítica de pesquisadores na área de computação, foi natural que houvesse encontros de pesquisadores brasileiros. Dois eventos antecedem a fundação da Sociedade Brasileira de Computação (SBC), e ainda hoje são seus eventos oficiais: o SECOMU – Seminário sobre Computação na Universidade, realizado em Porto Alegre em 1971 e o SEMISH – Seminário de Integração Software/Hardware, também realizado em Porto Alegre em 1974.

O I SEMISH ocorreu entre dias 14 a 18 de janeiro de 1974 na Universidade Federal do Rio Grande do Sul (UFRGS), sendo coordenado pelo professor Clésio Saraiva dos Santos, do Programa de Pós-Graduação em Ciência da Computação.

O SEMISH e o SECOMU passaram a ser eventos anuais. Para organizá-los foi então criada em julho de 1978 a Sociedade Brasileira de Computação, em Porto Alegre.

Atualmente a SBC é uma Sociedade plenamente consolidada, com dezenas de eventos anuais promovidos e apoiados. A Sociedade se estrutura em várias comissões especiais como Engenharia de Software, Inteligência Artificial etc. A SBC é membro ativo da IFIP, da IEEE Computer Society e do CLEI (*Centro Latino-americano de Estudios en Informática*).

## 8.52 Editor de Texto WordStar – 1979

O WordStar não foi o primeiro editor de texto, mas foi o primeiro a ser um grande sucesso dos microcomputadores. Até meados dos anos 1980, sua hegemonia foi imbatível. Ele foi, certamente, o pioneiro do gênero WYSIWYG (What You See Is What You Get)<sup>58</sup>, ao contrário de outros editores anteriores, com os quais você só ia saber mais ou menos qual seria a aparência do seu texto após executar um comando especial de formatação. Com o WordStar o

texto que ia ser impresso também ia sendo visualizado na tela à medida que era digitado. O editor inclusive, dependendo da máquina, fazia uso de caracteres especiais dos computadores para mostrar em tela caracteres sublinhados, em itálico ou negrito.

Porém, ele era ainda um editor baseado para telas de texto, não um editor gráfico como o Microsoft Word, que surgiu anos depois. Além disso, sua operação se dava através do uso do teclado. Por exemplo, para subir uma linha pressionava-se Control-E, para descer Control-X. A [Figura 8.42](#) mostra a tela de operação do WordStar com os principais comandos resumidos na parte superior.



Figura 8.42: Tela do WordStar.<sup>59</sup>

O WordStar foi produzido pela MicroPro International, empresa de propriedade de Seymour I. Rubinstein (Estados Unidos, 1934), mas o código foi totalmente produzido por Rob Barnaby, seu único programador até a versão 4.0. O sistema operacional no qual o WordStar rodava era o CP/M inicialmente, mas depois ele foi portado para o MS-DOS em 1982.

Um dos motivos do sucesso talvez tenha sido a portabilidade de sua interface. O programa procurava não fazer uso de características de

computadores que não fossem universais. Por isso, como nem todos os computadores tinham teclado com setas, mas a maioria tinha a tecla Control, a movimentação do cursor era feita com a combinação de Control mais as teclas E, S, D e X, e não com setas ou mouse, pois isso só funcionaria em alguns computadores. Os criadores do WordStar queriam que o mesmo modo de operação da ferramenta fosse usado em qualquer computador; assim, os usuários não estranhariam quando passassem de uma máquina para outra. Um dos fenômenos observados no caso de usuários de certos sistemas é a memória muscular, ou seja, independentemente do raciocínio do usuário, seus dedos vão direto às teclas que ele está acostumado a usar. O leitor que usa, por exemplo, Ctrl+C e Ctrl+V para copiar e colar com muita frequência, provavelmente nem precisa pensar onde estão as teclas; seus dedos irão sozinhos executar essa sequência de ações.

As vendas do WordStar iniciaram na faixa de 500 mil dólares em 1979, mas em 1984 já chegaram a 72 milhões. O produto, porém, era amado e odiado. A revista *PC Magazine* descrevia seu manual como “incrivelmente inadequado” e a *InfoWorld* descreveu o WordStar como “notório por sua complexidade”. Porém, sua capacidade de gerar textos “planos”, ou seja, sem caracteres especiais de controle, fez com que fosse usado pela comunidade de programadores para escrita de código. Uma versão do WordStar chegou mesmo a ser incorporada no ambiente de programação de uma variante da linguagem Pascal.

É possível que um dos motivos da queda do WordStar tenha sido justamente o fato de ele não aproveitar os novos recursos dos teclados dos PCs. Com o surgimento do WordPerfect e Microsoft Word, os comandos complexos do WordStar, como a sequência Ctrl+KS para salvar um arquivo ou Ctrl+PB para deixar o texto em negrito passaram a ser menos apreciados.

Uma característica que ele tinha, porém, que é considerada por muitos como superior aos atuais editores de texto, era a possibilidade de marcar um bloco de texto e deixá-lo marcado

enquanto se fazia outras operações, inclusive digitando. O bloco só seria desmarcado quando um comando específico para isso fosse executado. Nos editores de texto atuais, se um bloco estiver marcado e você digitar uma tecla que corresponda a um caractere qualquer, eles vão substituir todo o bloco por esse caractere.

## 8.53 Motorola 68000 – 1979

Em meados dos anos 1970, a Motorola encarou um novo desafio. O processador 6800, de 8 bits, já não dava conta da demanda de processamento de muitos projetos, e a construção de um processador de 16 bits ou até de 32 se fazia premente. Iniciou, então, o projeto do MC68000, também conhecido como MACSS (Motorola's Advanced Computer Systems on Silicon). Uma versão preliminar do circuito integrado é mostrada na [Figura 8.43](#).

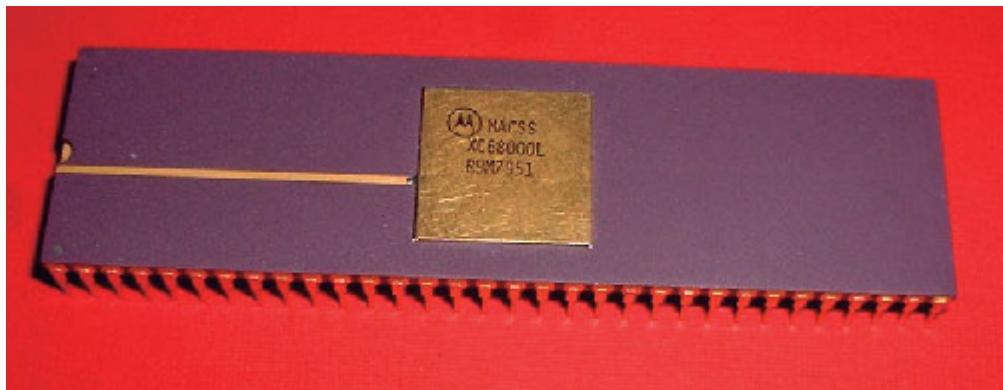


Figura 8.43: MC68000.<sup>60</sup>

Ao contrário de outros projetos do tipo, nos quais se buscava manter alguma compatibilidade com processadores mais antigos, os projetistas da Motorola foram instruídos a esquecer os processadores anteriores e projetar olhando para a frente. O 68000 deveria ser o melhor processador possível, independentemente do que já havia existido antes. A compatibilidade seria feita no futuro, já que futuros processadores teriam que ser compatíveis com o 68000.

Porém, um ponto em que o design do 68000 foi mais conservador foi em relação aos periféricos. Buscou-se uma interface de apenas 8 bits com periféricos para que o novo processador, quando lançado, já dispusesse de um conjunto útil e vasto. Se eles tivessem criado interfaces com 16 ou 32 bits, poderia levar anos até que novas opções fossem projetadas e produzidas; seriam periféricos bem mais rápidos, mas demorariam a surgir.

Porém, as instruções do novo processador seriam de 16 bits e não de 8 bits. Uma das limitações do antigo 6800, também, era a instrução de 8 bits. Não era possível representar 256 ( $2^8$ ) diferentes instruções, como se poderia pensar, porque 2 bits eram usados para identificar o registrador sobre o qual a operação seria feita; mais 2 bits indicavam o modo de endereçamento de memória, imediato, direto, indexado ou estendido. Sobravam assim apenas 4 bits para o código de operação, o que resultava em apenas 16 ( $2^4$ ) operações diferentes.

Já o novo MC68000 teria a possibilidade de representar mais de 64 mil operações diferentes com suas instruções de 16 bits e, assim, instruções muito mais complexas implementadas em hardware ou mesmo microcodificadas poderiam ser criadas. Uma instrução microcodificada não é executada diretamente pelo hardware, mas consta no repertório de instruções de máquina do processador por ser programada no hardware. Por exemplo, um processador pode não ter a instrução para multiplicar dois números. Entretanto, se ele tiver a instrução para somar dois números e uma instrução para repetir uma instrução N vezes, pode-se microprogramar uma instrução de máquina para fazer uma multiplicação usando a soma e a repetição. Para o programador, a instrução de multiplicação tem o mesmo status da instrução de adição: ambas são instruções do microprocessador. A única diferença é que enquanto a soma é executada diretamente em um ciclo de máquina, a multiplicação ocuparia vários ciclos de máquina, pois o hardware não a executa diretamente.

Outra limitação dos processadores de 8 bits era que eles permitiam endereçar uma memória de apenas 64 kB usando um endereçamento baseado em duas palavras de 8 bits. No final dos anos 1970, pouquíssimos programadores de microcomputador poderiam imaginar aplicações que usassem mais do que 64 kB, mas elas rapidamente vieram e, juntamente, a necessidade de espaços de endereçamento maiores.

Uma técnica que por algum tempo foi usada nos processadores de 8 bits foi a paginação, com a qual se adicionava mais um registrador de endereço que continha o número de uma “página” de memória de 64 kB dentre um conjunto disponível. Essa técnica, porém, tinha como desvantagem o fato de que só se podia acessar posições de uma página de cada vez. Você não poderia, por exemplo, somar diretamente uma posição de memória em uma página com uma posição em outra página. Entretanto, o 68000 teria um modo de endereçamento linear de 32 bits, ou seja, permitiria acessar livremente uma memória de até 4 GB ( $2^{32}$ ), ou 4 milhões de bytes, algo inimaginável na época, mas que logo se tornou necessário.

O 68000 se tornou o padrão da indústria para implementação de sistemas baseados em Unix. Ele foi o processador das estações de trabalho Sun e Apollo/Domain. Também foi usado nos microcomputadores Apple Lisa e Macintosh, Atari ST e Amiga. Após mais de 35 anos desde sua invenção, a série 68000 ainda está em produção.

## 8.54 SQL – 1979

Já vimos que os bancos de dados relacionais, ou seja, bancos de dados baseados em estruturas de tabelas, foram definidos no início dos anos 1970 por Edward Codd.

Ainda no início dessa década, Donald D. Chamberlin (Estados Unidos, 1944) e Raymond F. Boyce (Estados Unidos, 1947-1974), da IBM, trabalharam na definição de uma linguagem de propósito específico, que serviria tanto para definir essas estruturas de tabelas

quanto para consultar dados nelas. A linguagem foi originalmente chamada de SEQUEL (Structured English QUERy Language), mas, devido ao fato de que o termo já era marca registrada de uma empresa britânica da área de aviação, a linguagem foi rebatizada como SQL. Em inglês, porém seu nome ainda é pronunciado como “sequel” por muitas pessoas.

A linguagem SQL original seria usada para acesso aos dados do Sistema R, desenvolvido pela IBM. Ele não era totalmente relacional, ou seja, não obedecia totalmente às restrições estabelecidas por Codd para sistemas relacionais, mas era uma boa aproximação.

Em 1979, uma empresa chamada Relational Software Inc. percebeu o potencial da linguagem definida por Chamberlin e Boyce a partir dos conceitos de Codd e desenvolveu sua própria linguagem SQL para o sistema gerenciador de banco de dados que ela estava desenvolvendo. Em junho desse ano, a Relational Software lança a primeira versão comercial da linguagem SQL para computadores VAX. O sistema que incluía a linguagem chamava-se Oracle V2. Mais tarde, a própria empresa trocou seu nome para Oracle Corporation.

Até os dias de hoje, várias empresas implementam suas próprias versões de SQL e embora exista um padrão definido (ANSI 1986 e ISO 1987), a maioria das implementações não é compatível com as demais.

Apesar disso, a grande maioria das aplicações atuais que trabalham com bancos de dados utilizam uma ou outra forma de SQL, e não parece haver outra linguagem candidata a ocupar esse papel no futuro próximo. Ocorre que ela baseia-se no cálculo relacional, uma teoria que permite realizar otimizações em consultas e operações, de maneira que simplesmente ainda não se conhece forma mais eficiente para armazenar informações em computadores.

Por outro lado, a orientação a objetos de Alan Kay, um paradigma distinto de representação de informação, estabeleceu-se como

padrão para o design de sistemas de informação. Felizmente os dois padrões são compatíveis, pois estruturas orientadas a objetos podem ser convertidas em estruturas relacionais através de um modelo denominado ORM (Object-Relational Mapping)<sup>61</sup>. Assim, embora outros tipos de bancos de dados tenham surgido ao longo dos anos, como os bancos orientados a objetos, eles não resistiram e praticamente todas as iniciativas de representação de informação acabam caindo no modelo relacional mais cedo ou mais tarde – pelo menos no que se refere a sistemas de informação, visto que sistemas de inteligência artificial, por exemplo, seguem padrões completamente distintos para representação de informação.

## 8.55 Até Aqui...

Vimos que os anos 1970 testemunharam o surgimento de computadores pessoais realmente úteis e com aplicativos como editores de texto e planilhas eletrônicas que os tornavam, inclusive, uma opção bem mais barata para uso em empresas. Isso foi possível a partir da concepção de microprocessadores como o 4004, 8008 e 8080 da Intel. Estes ainda eram relativamente caros, mas com o lançamento do MOS 6502 a meros 25 dólares, microcomputadores suficientemente baratos e, até consoles de videogames, tornaram-se possíveis.

Porém, raramente estes computadores se ligavam em rede. Isso foi algo que só começou a se tornar usual na década seguinte. A ARPANET e outras redes ao redor do mundo seguiram crescendo, interligando principalmente órgãos governamentais e universidades. A invenção do TCP/IP nos anos 1970 permitiu que essas redes se interligassem, dando início ao fenômeno da Internet. O usuário doméstico ainda não tinha acesso a ela, entretanto. A invenção do modem para microcomputador, porém, deu início a essa possibilidade.

Em termos de interfaces, as ideias de Vannevar Bush e Ivan Sutherland continuavam atraentes e projetos de computadores com

interfaces gráficas começaram a engatinhar, mas o custo e a baixa tecnologia da época ainda não permitiam que fossem universalizados. Já havia computadores com capacidade gráfica e cores, inclusive, mas essa potencialidade usualmente só era aplicada em jogos, a maioria dos quais muito simples. Um ambiente se destacava, Smalltalk do Xerox PARC, uma linguagem de programação orientada a objetos totalmente baseada em ambiente gráfico. Porém, seu uso era ainda muito restrito.

No Brasil, os anos 1970 viram o surgimento do Pato Feio e da COBRA, a primeira indústria com tecnologia nacional. Surge também a SBC, Sociedade Brasileira de Computação, a partir da consolidação de eventos científicos que já se realizavam no país.

---

<sup>1</sup> Nathan Zeldes. Cortesia. Disponível em:  
<http://www.nzeldes.com/HOC/IntelFirsts.htm>

<sup>2</sup> Edição brasileira: *Gödel, Escher, Bach: um entrelaçamento de gênios brilhantes*. Brasília: Editora UnB, 2001.

<sup>3</sup> Tradução: “Um modelo relacional de dados para grandes bancos de dados compartilhados.”

<sup>4</sup> “Digital PDP11-IMG 1498 cropped” by Digital\_PDP11-IMG\_1498.jpg: Rama & Musée Boloderivative work: Morn - This file was derived from Digital PDP11-IMG 1498.jpg:. Licenced under CC BY-SA 2.0 fr via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Digital\\_PDP11-IMG\\_1498\\_cropped.jpg#/media/File:Digital\\_PDP11-IMG\\_1498\\_cropped.jpg](https://commons.wikimedia.org/wiki/File:Digital_PDP11-IMG_1498_cropped.jpg#/media/File:Digital_PDP11-IMG_1498_cropped.jpg)

<sup>5</sup> “Ken Thompson (sitting) and Dennis Ritchie at PDP-11 (2876612463)” by Peter Hamer - Ken Thompson (sitting)-&-Bjarne Stroustrup at PDP-11 Uploaded by Magnus Manske. Licensed under CC BY-SA 2.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Ken\\_Thompson\\_\(sitting\)\\_and\\_Dennis\\_Ritchie\\_at\\_PDP-11\\_\(2876612463\).jpg#/media/File:Ken\\_Thompson\\_\(sitting\)\\_and\\_Dennis\\_Ritchie\\_at\\_PDP-11\\_\(2876612463\).jpg](https://commons.wikimedia.org/wiki/File:Ken_Thompson_(sitting)_and_Dennis_Ritchie_at_PDP-11_(2876612463).jpg#/media/File:Ken_Thompson_(sitting)_and_Dennis_Ritchie_at_PDP-11_(2876612463).jpg)

<sup>6</sup> Tradução: “Sem barba e sem barriga não é guru.”

<sup>7</sup> © Intel Free Press. Licença CC BY 2.0. Disponível em:  
<https://www.flickr.com/photos/intelfreepress/6351075653/>

<sup>8</sup> “Man Who Designed the World’s First Microprocessor” by Intel Free Press - Disponível em:  
<http://www.flickr.com/photos/intelfreepress/6254699845/sizes/o/in/photostream/>. Licensed under CC BY-SA 2.0 via Wikimedia Commons.  
[https://commons.wikimedia.org/wiki/File:Man\\_Who\\_Designed\\_the\\_World%27s\\_First\\_Microprocessor.jpg#](https://commons.wikimedia.org/wiki/File:Man_Who_Designed_the_World%27s_First_Microprocessor.jpg#/)  
[https://media/File:Man\\_Who\\_Designed\\_the\\_World%27s\\_First\\_Microprocessor.jpg](https://media/File:Man_Who_Designed_the_World%27s_First_Microprocessor.jpg)

<sup>9</sup> Tradução: O Rastejador.

<sup>10</sup> Tradução: “Eu sou o rastejador. Pegue-me se for capaz!”

<sup>11</sup> Tradução: O Ceifador.

<sup>12</sup> Tradução: Coelho.

<sup>13</sup> Referência à forma como Hortelino Trocaletra (Elmer Fudd) se referia ao coelho Pernalonga em inglês, em função de seu problema de dicção característico.

<sup>14</sup> Tradução: Impregnado.

<sup>15</sup> “Kenbak1” by Kathryn Greenhill - Kenbak 1 1973. *World’s first commercial personal computer*, Computing History Museum, Mountain View, California. Uploaded by liftarn. Licensed under CC BY-SA 2.0 via Commons. Disponível em:

<https://commons.wikimedia.org/wiki/File:Kenbak1.jpg#/media/File:Kenbak1.jpg>

<sup>16</sup> “Floppy disk 2009 G1” by George Chernilevsky - Own work. Licensed under Public Domain via Commons. Disponível em:  
[https://commons.wikimedia.org/wiki/File:Floppy\\_disk\\_2009\\_G1.jpg#/media/File:Floppy\\_disk\\_2009\\_G1.jpg](https://commons.wikimedia.org/wiki/File:Floppy_disk_2009_G1.jpg#/media/File:Floppy_disk_2009_G1.jpg)

<sup>17</sup> Tradução: Na mesma hora.

<sup>18</sup> Tradução: “C++ roda mais rápido, mas frequentemente na direção errada.”

<sup>19</sup> Tradução: Código C Ofuscado.

<sup>20</sup> © Copyright 1984-2015, Leo Broukhis, Simon Cooper, Landon Curt Noll - All rights reserved

*This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.* Disponível em: <http://www0.us.ioccc.org/1998/banks.c>

<sup>21</sup> “Patinho feio”. Licenciado sob CC BY 3.0, via Wikimedia Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:Patinho\\_feio.jpg#/media/File:Patinho\\_feio.jpg](https://commons.wikimedia.org/wiki/File:Patinho_feio.jpg#/media/File:Patinho_feio.jpg)

<sup>22</sup> “Magnavox-Odyssey-Console-Set” by Evan-Amos - Own work. Licensed under Public Domain via Commons. Disponível em:

<https://commons.wikimedia.org/wiki/File:Magnavox-Odyssey-Console-Set.jpg#/media/File:Magnavox-Odyssey-Console-Set.jpg>

<sup>23</sup> “HP9830A-HP9866” by Hydrargyrum - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em:

<https://commons.wikimedia.org/wiki/File:HP9830A-HP9866.png#/media/File:HP9830A-HP9866.png>

<sup>24</sup> By Chris Rand - Own work, CC BY-SA 3.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=28364913>

<sup>25</sup> “EthernetCableBlue2” by Raysonho @ Open Grid Scheduler / Grid Engine - Own work. Licensed under CC0 via Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:EthernetCableBlue2.jpg#/media/File:EthernetCableBlue2.jpg>

<sup>26</sup> “Xerox Alto mit Rechner” by Joho345 - Own work. Licensed under Public Domain via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Xerox\\_Alto\\_mit\\_Rechner.JPG#/media/File:Xerox\\_Alto\\_mit\\_Rechner.JPG](https://commons.wikimedia.org/wiki/File:Xerox_Alto_mit_Rechner.JPG#/media/File:Xerox_Alto_mit_Rechner.JPG)

<sup>27</sup> “Micral MGR Lyon-IMG 9895” by Photograph by Rama, Wikimedia Commons, Cc-by-sa-2.0-fr. Licensed under CC BY-SA 2.0 fr via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Micral\\_MGR\\_Lyon-IMG\\_9895.JPG#/media/File:Micral\\_MGR\\_Lyon-IMG\\_9895.JPG](https://commons.wikimedia.org/wiki/File:Micral_MGR_Lyon-IMG_9895.JPG#/media/File:Micral_MGR_Lyon-IMG_9895.JPG)

<sup>28</sup> By 2007Computex\_e21Forum-MartinCooper.jpg: Rico Shenderivative work: PowellS (talk) - 2007Computex\_e21Forum-MartinCooper.jpg, CC BY-SA 3.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=17589802>

<sup>29</sup> Tradução: “Joel, eu estou ligando para você de um telefone celular ‘verdadeiro’. Um telefone portátil de mão.”

<sup>30</sup> “Community Memory Project, 1975, Computer History Museum, Mountain View, California” by Kathryn Greenhill - Flickr: Community Memory Project, 1975 , Computing History Museum, Mountain View, California. Licenced under CC BY-SA 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Community\\_Memory\\_Project,\\_1975,\\_Computer\\_History\\_Museum,\\_Mountain\\_View,\\_California.jpg#/media/File:Community\\_Memory\\_Project,\\_1975,\\_Computer\\_History\\_Museum,\\_Mountain\\_View,\\_California.jpg](https://commons.wikimedia.org/wiki/File:Community_Memory_Project,_1975,_Computer_History_Museum,_Mountain_View,_California.jpg#/media/File:Community_Memory_Project,_1975,_Computer_History_Museum,_Mountain_View,_California.jpg)

<sup>31</sup> “Intel 8080 Advertisement May 1974” by Intel Corporation, Santa Clara, CA. Advertising agency, Regis McKenna, Palo Alto, CA. - Scanned from pages 22 and 23 of the May 2, 1974 issue of Electronics magazine by Michael Holley Swtpc6800. Licenced under Public Domain via Wikimedia Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:Intel\\_8080\\_Advertisement\\_May\\_1974.jpg#/media/File:Intel\\_8080\\_Advertisement\\_May\\_1974.jpg](https://commons.wikimedia.org/wiki/File:Intel_8080_Advertisement_May_1974.jpg#/media/File:Intel_8080_Advertisement_May_1974.jpg)

<sup>32</sup> “Altair 8800 Computer” by Swtpc6800 en:User:Swtpc6800 Michael Holley - Transferred from en.wikipedia. Licensed under Public Domain via Wikimedia Commons.

Disponível em:

[https://commons.wikimedia.org/wiki/File:Altair\\_8800\\_Computer.jpg#/media/File:Altair\\_8800\\_Computer.jpg](https://commons.wikimedia.org/wiki/File:Altair_8800_Computer.jpg#/media/File:Altair_8800_Computer.jpg)

<sup>33</sup> Tradução: Protocolo de Controle de Transmissão / Protocolo Inter-Redes.

<sup>34</sup> Não confundir com a WWW, ou World Wide Web, que só foi criada em 1989 e é um dos muitos serviços da Internet.

<sup>35</sup> © Dionata Costa - Agência de Notícias São Joaquim Online. Cortesia. Disponível em: <http://saojoaquimonline.net/2013/07/16/primeiro-computador-de-sao-joaquim-esta-exposto-em-museu-na-infoneve-informatica/>

<sup>36</sup> Edição brasileira: *O mítico homem-mês: ensaios sobre engenharia de software*. Rio de Janeiro: Editora Campus Elsevier, 2009.

<sup>37</sup> “IBM 5100 - MfK Bern” by Sandstein - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:IBM\\_5100\\_-MfK\\_Bern.jpg#/media/File:IBM\\_5100\\_-MfK\\_Bern.jpg](https://commons.wikimedia.org/wiki/File:IBM_5100_-MfK_Bern.jpg#/media/File:IBM_5100_-MfK_Bern.jpg)

<sup>38</sup> “Sphere Personal Computer Ad January 1976” by Sphere Corporation - Scanned from page 23 of the January 1976 Byte magazine by Michael Holley Swtpc6800. Licensed under Public Domain via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Sphere\\_Personal\\_Computer\\_Ad\\_January\\_1976.jpg#/media/File:Sphere\\_Personal\\_Computer\\_Ad\\_January\\_1976.jpg](https://commons.wikimedia.org/wiki/File:Sphere_Personal_Computer_Ad_January_1976.jpg#/media/File:Sphere_Personal_Computer_Ad_January_1976.jpg)

<sup>39</sup> “Cray-1-p1010221” by Rama, carol for small repairs - Image:Cray-1-p1010221.jpg rotated and cropped.. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:Cray-1-p1010221.png#/media/File:Cray-1-p1010221.png>

<sup>40</sup> “CPM Ad, Dec 11, 1978” by Digital Research - Scanned from the December 11, 1978 InfoWorld magazine. Licensed under Public Domain via Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:CP%81%84M\\_Ad,\\_Dec\\_11,\\_1978.png#/media/File:CP%81%84M\\_Ad,\\_Dec\\_11,\\_1978.png](https://commons.wikimedia.org/wiki/File:CP%81%84M_Ad,_Dec_11,_1978.png#/media/File:CP%81%84M_Ad,_Dec_11,_1978.png)

<sup>41</sup> Tradução: Clube do Computador Feito em Casa.

<sup>42</sup> “Apple first logo” by Unknown - Transferred from en.wikipedia to Commons by Sreejithk2000 using CommonsHelper. Licensed under Public Domain via Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:Apple\\_first\\_logo.png#/media/File:Apple\\_first\\_logo.png](https://commons.wikimedia.org/wiki/File:Apple_first_logo.png#/media/File:Apple_first_logo.png)

<sup>43</sup> “Apple I Computer” by Ed Uthman - originally posted to Flickr as Apple I Computer. Licensed under CC BY-SA 2.0 via Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:Apple\\_I\\_Computer.jpg#/media/File:Apple\\_I\\_Computer.jpg](https://commons.wikimedia.org/wiki/File:Apple_I_Computer.jpg#/media/File:Apple_I_Computer.jpg)

<sup>44</sup> “Zilog Z80” by *The original uploader was Damicatz at English Wikipedia - Transferred from en.wikipedia to Commons.. Licensed under CC BY 2.5 via Commons.* Disponível em:

[https://commons.wikimedia.org/wiki/File:Zilog\\_Z80.jpg#/media/File:Zilog\\_Z80.jpg](https://commons.wikimedia.org/wiki/File:Zilog_Z80.jpg#/media/File:Zilog_Z80.jpg)

<sup>45</sup> “Processor Technology SOL 20 Computer” by *Swtpc6800 en:User:Swtpc6800 Michael Holley - Own work. Licensed under Public Domain via Commons.* Disponível em:

[https://commons.wikimedia.org/wiki/File:Processor\\_Technology\\_SOL\\_20\\_Computer.jpg#/media/File:Processor\\_Technology\\_SOL\\_20\\_Computer.jpg](https://commons.wikimedia.org/wiki/File:Processor_Technology_SOL_20_Computer.jpg#/media/File:Processor_Technology_SOL_20_Computer.jpg)

<sup>46</sup> Jim Battle. Cortesia. Fonte: <http://www.sol20.org/>

<sup>47</sup> Steen Stengel. Cortesia. Disponível em:  
<http://www.oldcomputers.net/computecolor-8001.html>

<sup>48</sup> “Dale Heatherington with 80-103” by *User Maury Markowitz on en.wikipedia - Image taken on Dale Heatherington’s own camera: “The first two were shot with my camera so there is no question about publishing rights.”. Licensed under Copyrighted free use via Commons.* Disponível em:

[https://commons.wikimedia.org/wiki/File:Dale\\_Heatherington\\_with\\_80-103.jpg#/media/File:Dale\\_Heatherington\\_with\\_80-103.jpg](https://commons.wikimedia.org/wiki/File:Dale_Heatherington_with_80-103.jpg#/media/File:Dale_Heatherington_with_80-103.jpg)

<sup>49</sup> “Hayes 300 Baud Smartmodem 02” by *Michael Pereckas from Milwaukee, WI, USA - Smartmodem. Licensed under CC BY-SA 2.0 via Commons.* Disponível em:  
[https://commons.wikimedia.org/wiki/File:Hayes\\_300\\_Baud\\_Smartmodem\\_02.jpg#/media/File:Hayes\\_300\\_Baud\\_Smartmodem\\_02.jpg](https://commons.wikimedia.org/wiki/File:Hayes_300_Baud_Smartmodem_02.jpg#/media/File:Hayes_300_Baud_Smartmodem_02.jpg)

<sup>50</sup> “VAX 11-780 intero” by *Emiliano Russo, Associazione Culturale VerdeBinario. Licensed under Public Domain via Commons.* Disponível em:  
[https://commons.wikimedia.org/wiki/File:VAX\\_11-780\\_intero.jpg#/media/File:VAX\\_11-780\\_intero.jpg](https://commons.wikimedia.org/wiki/File:VAX_11-780_intero.jpg#/media/File:VAX_11-780_intero.jpg)

<sup>51</sup> By Rama, CC BY-SA 2.0 fr. Disponível em:  
<https://commons.wikimedia.org/w/index.php?curid=11276454>

<sup>52</sup> “RadioShack TRS80-IMG 7206” by *Rama & Musée Bolo - Own work. Licensed under CC BY-SA 2.0 fr via Commons.* Disponível em:  
[https://commons.wikimedia.org/wiki/File:RadioShackTRS80-IMG\\_7206.jpg#/media/File:RadioShackTRS80-IMG\\_7206.jpg](https://commons.wikimedia.org/wiki/File:RadioShackTRS80-IMG_7206.jpg#/media/File:RadioShackTRS80-IMG_7206.jpg)

<sup>53</sup> “Commodore PET2001” by *Photographer: Tomislav Medak from Flickr / Editing: Bill Bertram (Pixel8) -* Disponível em:  
<http://www.flickr.com/photos/tomislavmedak/3803230853/>. Licensed under CC BY-SA 3.0 via Commons.

[https://commons.wikimedia.org/wiki/File:Commodore\\_PET2001.jpg#/media/File:Commodore\\_PET2001.jpg](https://commons.wikimedia.org/wiki/File:Commodore_PET2001.jpg#/media/File:Commodore_PET2001.jpg)

<sup>54</sup> By Georges Seguin (Okki) - Own work, CC BY-SA 3.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=4884157>

<sup>55</sup> By Jarekt (Own work), Public domain, via Wikimedia Commons.

<sup>56</sup> "Intel 8086 CPU Die" by Pdesousa359 - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Intel\\_8086\\_CPU\\_Die.JPG#/media/File:Intel\\_8086\\_CPU\\_Die.JPG](https://commons.wikimedia.org/wiki/File:Intel_8086_CPU_Die.JPG#/media/File:Intel_8086_CPU_Die.JPG)

<sup>57</sup> "Visicalc" by User:Gortu - apple2history.org. Licensed under Public Domain via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:Visicalc.png#/media/File:Visicalc.png>

<sup>58</sup> Tradução: O que Você Vê É o que Você Obtém.

<sup>59</sup> "WordStar" by --Plenz 20:42, 19. Jul 2004 (CEST) - eigenhändig erstellt, und zwar mit einem originalen WordStar 3.0 auf einem CP/M-Emulator unter Windows. Licensed under GFDL via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:WordStar.png#/media/File:WordStar.png>

<sup>60</sup> "XC68000.agr". Licensed under CC BY-SA 3.0 via Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:XC68000.agr.jpg#/media/File:XC68000.agr.jpg>

<sup>61</sup> Tradução: Mapeamento Objeto-Relacional.

## **PARTE IX**

### **Evolução dos Computadores Pessoais e sua Interconexão**

Os anos 1980 viram duas grandes mudanças na área da computação, as quais moldaram a nova sociedade tecnológica na qual vivemos. O primeiro movimento apareceu no início da década com a popularização do computador pessoal. Circuitos integrados cada vez mais poderosos e baratos possibilitaram a construção de computadores suficientemente potentes para fazer coisas úteis e ainda assim a um preço acessível para famílias e pequenas empresas.

O segundo grande movimento foi no sentido da interligação desses computadores em rede. Foi nessa década que a Internet, que já consolidava várias redes de computadores ao redor do mundo, começou a se tornar acessível aos usuários domésticos e pequenas empresas. No finalzinho da década, foi criada a World Wide Web, que remodelou a maneira como a sociedade humana funciona.

Os anos 1980 também viram a ascensão meteórica e a queda ainda mais meteórica da indústria de videogames. Surgiram nesse período o computador padrão PC, os processadores RISC, os notebooks e até o computador de bolso.

Em termos de software, várias linguagens como Ada, C++, Turbo Pascal, Eiffel e Perl surgiram, além de novos sistemas operacionais como o MS-DOS, o Mac OS e o GNU. Foram anos de grandes e incríveis avanços, como veremos.

## 9.1 dBASE II – 1980

Em 1980, a empresa Ashton-Tate começou a distribuir um sistema de banco de dados chamado dBASE II. Seu nome era uma contração de “data base” (banco de dados), e o “II” foi usado por razões comerciais: nunca houve um dBASE I, a primeira versão já foi numerada com II. Eles consideravam que isso dava uma impressão de maior maturidade ao sistema. Para potenciais clientes, um software na segunda versão certamente já teria seus principais bugs eliminados.

Assim, o dBASE II, inicialmente desenvolvido para o sistema operacional CP/M por Wayne Ratliff (Estados Unidos, 1946), rapidamente se tornou muito popular, e até hoje ainda se encontram sistemas desenvolvidos nessa ferramenta, embora a maioria já tenha sido substituída por bancos de dados mais modernos.

Ele foi, de fato, construído a partir da experiência de Ratliff no desenvolvimento do sistema Vulcan (nome dado em homenagem ao planeta natal do Sr. Spock da série *Jornada nas Estrelas*). Ratliff trabalhava na época na NASA, no programa Viking, que enviava sondas a Marte. Porém, Vulcan não foi criado para armazenar dados espaciais, e, sim, para organizar as apostas dos funcionários da NASA em jogos de futebol.

Em parte, o sucesso do dBASE II deve-se primeiro à inexistência de produtos semelhantes para microcomputadores e em segundo lugar porque era um ambiente completo para construção de aplicações. Ele tinha um banco de dados, um sistema de consultas, um gerador de formulários e uma linguagem de programação.

Quando a IBM lançou o PC em 1981, a Ashton-Tate rapidamente portou o dBASE II para esse novo computador, fazendo dele um dos poucos produtos comerciais disponíveis para o PC no início, o que só aumentou ainda mais sua popularidade.

Pela sua facilidade em permitir a criação de aplicações comerciais, dBASE II é considerado como a primeira linguagem/ambiente a possibilitar ao usuário leigo que produzisse sistemas informatizados.

Entre outras características, ele tinha uma linha de comando que permitia que consultas sobre o banco de dados fossem digitadas e seu resultado imediatamente visualizado.

Em 1984 foi lançado o dBASE III, escrito em C e não em Assembly como seu antecessor. Isso permitiu que o sistema fosse portado para Unix e VMS. Em 1986 foi lançado o dBASE III+, com novas características, como a capacidade de exibir menus na tela. Com essa versão, apenas em 1987 a empresa faturou 300 milhões de dólares.

Finalmente, em 1988 é lançado o dBASE IV, o qual ficou bem abaixo das expectativas, pois não possuía todas as características prometidas pela empresa e, além disso, era repleto de bugs, o que fez com que os usuários acabassem abandonando essa tecnologia em detrimento de outras. E, assim, a Ashton-Tate viu seu mercado de sistemas de banco de dados para PCs, que era de 63% em 1988, cair para 43% no ano seguinte e continuar em queda livre.

## 9.2 Linguagem Ada – 1980

O Departamento de Defesa (DoD) dos Estados Unidos sempre teve papel muito ativo na história da computação, conforme já vimos. Em 1970 havia uma preocupação deste departamento em relação ao número de diferentes linguagens de programação usadas em seus projetos – na época, mais de 450, segundo consta.

Foi assim criado em 1975 o HOLWG (High Order Language Working Group) com o objetivo de estabelecer uma linguagem de alto nível única para o DoD. O grupo era coordenado pelo coronel William Whitaker (Estados Unidos, 1936-2010) e fortemente baseado nos Estados Unidos, mas a cooperação com países da OTAN também foi grande, até porque nos anos 1970 a pesquisa em linguagens de programação foi muito mais intensa na Europa do que nos Estados Unidos. Alguns pesquisadores de renome que participaram desse grupo foram Edgster Dijkstra (Holanda, 1930-2001), Niklaus Wirth (Suiça, 1934) e Charles Hoare (Ceilão, 1934).

Os objetivos do grupo eram estabelecer os requisitos de uma linguagem única para o DoD, verificar se alguma existente já atendia a estes requisitos e propor o uso desta linguagem ou a criação de uma nova na medida do necessário.

Após bastante trabalho, produziram uma proposta denominada “Straw-Man Proposal” (proposta “homem de palha”, ou “espantalho”), que é um jargão comum em inglês para indicar os resultados de uma tempestade cerebral. Essa proposta foi depois consolidada em outra, conhecida como “Steelman Language Requirements” (requisitos de linguagem “homem de aço”). Assim, em 1977 eles concluíram que nenhuma linguagem existente atendia a todos os requisitos elaborados pela equipe.

O DoD lançou, então, uma licitação para a definição e construção da nova linguagem, e quatro grupos se qualificaram na primeira fase:

- Grupo vermelho (Intermetrics).
- Grupo verde (Honeywell Bull).
- Grupo azul (SofTech).
- Grupo amarelo (SRI International).

Em maio de 1979, o grupo verde ganhou o contrato. A nova linguagem, baseada na LIS, que eles haviam desenvolvido nos anos 1970, foi batizada de Ada, em homenagem à Condessa de Lovelace. O manual da linguagem foi aprovado pelos militares no dia 10 de dezembro de 1980 (dia no aniversário de Ada Lovelace) e foi numerado como MIL-STD-1815, sendo 1815 o ano de nascimento de Ada Lovelace.

Ada era uma linguagem que visava, acima de tudo, segurança. Era ao estilo de ALGOL, mas com várias características únicas. Ela obteve muita atenção da comunidade em seus primeiros anos, e acreditava-se que se tornaria uma linguagem dominante. Seu criador, Jean Ichbiah (França, 1940-2007), chegou mesmo a afirmar que em 10 anos haveria apenas duas linguagens de programação em uso no mundo: Ada e LISP.

Entretanto, ela se demonstrou muito difícil de implementar. Era vasta e complexa; em função disso, o tempo de compilação era alto e a eficiência do código executável gerado, baixa. Dessa forma, a linguagem acabou não sendo muito popular fora do DoD. Apesar disso, entre 1991 e 1997, o DoD exigia que qualquer sistema desenvolvido em seus projetos fosse feito em Ada, embora muitas vezes exceções a essa regra fossem consideradas.

Atualmente, devido às suas características de segurança, a linguagem Ada ainda é bastante usada em sistemas que requerem essa particularidade, como sistemas militares e sistemas de controle de aeronaves. O sistema de voo automatizado do Boeing 777, por exemplo, foi todo escrito em Ada.

Na tabela TIOBE, em 1985 e 1995 Ada chegou a aparecer em terceiro lugar entre as linguagens de programação, mas em fevereiro de 2016 havia caído para a 31<sup>a</sup> posição, abaixo até do velho FORTRAN, que está na posição 25.

### 9.3 VIC-20 – 1980

O Commodore VIC-20 ([Figura 9.1](#)) foi o segundo modelo lançado pela Commodore, em 1980, logo depois do PET. Ele é digno de nota aqui porque foi o primeiro computador no mundo a vender mais de 1 milhão de unidades, sendo que nos períodos de pico até 9 mil unidades eram produzidas por dia.



**Figura 9.1: Commodore VIC-20.<sup>1</sup>**

A sigla VIC vem de Vídeo Interface Chip, um chip usado para produção de gráficos coloridos e som que fazia parte do seu projeto. Quanto à escolha do número 20, foi arbitrária. O gerente do projeto, Michael Tomczyk, teria dito: “VIC soava como o nome de um motorista de caminhão, então eu insisti em adicionar um número. Eu escolhi ‘20’ e quando Jack Tramiel perguntou ‘por que 20?’, eu respondi ‘porque é um número amigável e este vai ser um computador amigável.’” Tramiel, presidente da Commodore, concordou.

O VIC-20 foi lançado meio que por acaso, pois a Commodore havia desenvolvido, dois anos antes do seu lançamento, o chip VIC para uso em videogames; mas eles não conseguiram colocá-lo no mercado e, assim, resolveram lançar seu próprio computador com capacidade gráfica igual à de um videogame.

O VIC-20 foi lançado primeiramente no Japão em 1980, já adaptado para a escrita japonesa, e em 1981 nos Estados Unidos. As primeiras máquinas tinham apenas 4 kB de memória, mas podiam ser estendidas com o uso de cartuchos semelhantes aos dos antigos videogames.

Sua memória era muito pequena para rodar, por exemplo, o BASIC, mas isso era resolvido pelo uso dos cartuchos. Um contendo BASIC, por exemplo, já vinha com o programa instalado na sua própria memória ROM. Assim, o compilador BASIC não precisava ser carregado na pequena memória do VIC, que ficava totalmente disponível para programas e dados.

A escolha por usar chips de 1 kB para construir a memória do VIC veio do fato de que a Commodore tinha comprado uma quantidade excessiva destes e não tinha mais onde usá-los. O VIC foi assim, para a empresa, um meio se livrar de muitos circuitos encalhados, desde processadores gráficos até memórias.

Seu sucesso deveu-se principalmente ao baixo custo, pois foi o primeiro computador em cores a custar menos de 300 dólares. A empresa também contratou William Shatner (o ator que encarnou o Capitão Kirk em *Jornada nas Estrelas*) para fazer o comercial do computador, no qual ele perguntava à audiência: “Por que comprar um simples videogame?”, aludindo ao fato de que pelo mesmo preço você poderia ter um computador que rodava jogos com a mesma qualidade de um videogame.

A empresa também contratou Scott Adams, o autor das tirinhas *Dilbert*, para portar para o VIC seus jogos em estilo “Adventure”. Na época, estes jogos eram baseados apenas em texto, mas mesmo assim, criavam ambientes bastante interessantes para os jogadores. Inicialmente, o computador descrevia o ambiente em que o usuário se encontrava e indicava algumas palavras-chave que podiam ser usadas como comandos, como “olhar”, “pegar”, “abrir” etc. Na sequência, o jogador podia escrever frases usando estes comandos, como “olhe para cima”, e o computador respondia descrevendo

novas partes da cena ou mudando o estado do jogo, por exemplo, levando virtualmente o jogador para outros cenários ou alterando o estado dos objetos.

O VIC foi descontinuado em 1985 após a Commodore lançar o modelo 64 com 64 kB de memória, além de gráficos e sons bastante superiores aos do VIC. O último acessório para o computador, porém, foi lançado em 1986 e tratava-se de um sintetizador de voz. A revista *Ahoy!* escreveu sobre ele: “Acredite ou não, um novo acessório para o VIC... Estamos tão surpresos quanto você!”

## 9.4 Apple III – 1980

Nem só de sucessos vive a Apple. Um de seus maiores fracassos foi o lançamento do Apple III em 1980, que devia ser uma versão melhorada do Apple II, mas ainda seria um computador de 8 bits. O fracasso do Apple III deveu-se principalmente a dois motivos. O primeiro deles foi que o produto foi lançado com muitos defeitos e um *recall* precisou ser feito para as primeiras 14 mil unidades vendidas. Isso manchou a imagem do Apple III de forma que mesmo com o lançamento de uma versão mais estável em 1981 as vendas permaneceram baixas. O outro motivo foi o lançamento do IBM PC em 1981, um microcomputador de 16 bits. Todo o mercado se voltou para esse novo modelo que, além de ser de 16 bits, era produto de uma empresa muito mais sólida no mercado, e tinha um custo relativamente baixo. O Apple III foi projetado para ser um computador para empresas e neste segmento a competição com o IBM PC foi cruel.

A [Figura 9.2](#) mostra um Apple III+, modelo de 1984, um dos últimos a serem lançados.



**Figura 9.2: Apple III+.**<sup>2</sup>

Steve Wozniak teria afirmado em uma oportunidade que a razão do fracasso do Apple III deveu-se ao fato de ele ter sido projetado pelo departamento de marketing da Apple e não pelo departamento de engenharia. De acordo com ele, a quantidade de falhas de hardware do computador era de 100%, ou seja, todas as unidades produzidas falharam de início.

Acredita-se que um dos principais problemas do Apple III fosse o superaquecimento. Há relatos, inclusive, de que os chips chegavam a pular de seus slots de tão quentes que ficavam.

A ideia da Apple era segmentar seu mercado. O Apple II deveria permanecer como um computador para hobistas, como computador pessoal e educacional com apenas 10% do mercado. Já o Apple III, com sua tela de 80 colunas e aplicativos comerciais, seria o

computador para o mercado empresarial, representando cerca de 90% do mercado. Acreditava-se, porém, que quando o III fosse lançado as vendas do II parariam imediatamente. Wozniak lamentava essa possibilidade pois sempre se sentira mais próximo dos hobistas do que dos empresários.

E, de fato, pouco antes do lançamento do III a Apple cancelou todos os projetos de engenharia envolvendo o Apple II. Porém, com o fracasso do III a empresa voltou a desenvolver novas versões do Apple II. Assim, foi lançado em 1983 o Apple Ile e em 1984 o IIC. Outras versões do Apple II continuaram sendo produzidas até 1996. Paralelamente, a companhia iniciou em 1983 o projeto do Lisa e, em 1984, do Macintosh.

## 9.5 Sinclair ZX80 – 1980

Em 1980, uma empresa do Reino Unido chamada Science of Cambridge Ltda. lançou um pequeno computador pessoal construído com base do processador Z80, o Sinclair ZX80 ([Figura 9.3](#)). Posteriormente, a empresa mudou seu nome para Sinclair, que era o sobrenome de seu criador, Sir Clive Sinclair (Reino Unido, 1940). Em relação ao nome do computador, Z80 é obviamente referência ao microprocessador usado e o “X” refere-se ao “ingrediente secreto”.



Figura 9.3: Sinclair ZX80.<sup>3</sup>

O computador impressionava por seu pequeno tamanho: cabia em uma bolsa pequena e pesava apenas 340 gramas, mas ainda assim era um computador. Seu manual tinha 128 páginas, incluindo instruções sobre programação em BASIC. Programas podiam ser lidos e gravados em unidades de fita K7 (não incluídas) e qualquer aparelho de TV servia como monitor. A tela tinha 24 linhas de 32 colunas, com letras pretas em fundo branco, o que não era comum na época pois, por questões de luminosidade, usualmente computadores trabalhavam com fundo escuro e letras luminosas.

Como outros microcomputadores de seu tempo, ele provocava interferências em aparelhos de rádio, mas isso era minimizado pelo fato de sua caixa ser pintada internamente com tinta metálica.

Ele tinha apenas 1 kB de memória RAM e 4 kB de memória ROM, na qual havia um Integer BASIC residente e um gerador de

caracteres. Era um computador muito simples, e o microprocessador Z80 executava praticamente todas as funções da máquina; ele não tinha coprocessadores para tratar entrada e saída de dados, ou controle de vídeo, por exemplo, coisa que já era comum em computadores maiores. Devido a essa limitação, cada vez que o usuário digitava uma tecla em seu teclado de membrana, a tela piscava, pois enquanto o pobre Z80 processava a entrada de dados e sua interpretação, ele não conseguia manter o texto que estava sendo exibido na tela.

Uma das coisas boas desse modelo era que as instruções do BASIC não precisavam ser totalmente digitadas. Bastava pressionar uma tecla, por exemplo, “G”, para que o comando “GOTO” aparecesse na linha de comando. Isso dependia de contexto, ou seja, o GOTO apareceria somente se a linha de comando estivesse em um estado no qual um comando como esse fosse esperando, digamos, no início da linha ou após um THEN. Em outras situações, a tecla “G” simplesmente geraria uma letra “G”. Essa característica de representar comandos com uma única letra também tinha implicações no uso da memória, pois assim bastaria um byte para guardar um comando na exígua memória do ZX80.

No Brasil, vários clones do ZX80 e seus sucessores foram fabricados. Entre eles, a Microdigital produziu os TK82 e TK83, com 2 kB de memória RAM e o TK85 com 16 kB. Já a Prológica produziu o CP 200 praticamente idêntico ao TK 85.

## 9.6 UseNet – 1980

Um dos antecessores dos fóruns da Internet surgiu em 1979 e se chamava USENET. USENET basicamente é um serviço de publicação de notícias semelhante aos BBS, já mencionados. Talvez a principal diferença é que no caso da USENET as publicações são separadas por categorias, conhecidas como newsgroups ou grupos de notícias.

A USENET, apelidada de “ARPANET dos pobres”, surgiu por iniciativa de dois estudantes da Universidade Duke, Tom Truskott e Jim Ellis (Estados Unidos, 1956-2001) na virada de 1979 para 1980. Nessa época, ocorreu uma atualização de sistema operacional no computador da universidade e o sistema de notícias que era usado parou de funcionar. Motivados por isso, pelo fato de que os preços dos modems vinham caindo e também pelo surgimento do protocolo UUCP (Unix to Unix Copy Protocol), que facilitava bastante a cópia de arquivos entre computadores usando Unix, eles criaram um software que permitiria não apenas aos membros da sua universidade publicarem notícias internas, mas também possibilitava a cooperação com outras universidades, o que na época só era feito pessoalmente ou através de telefone ou correio.

O nome USENET vem do fato de que toda a informação ali depositada é criada pelos próprios usuários (“USER NET”, ou “rede dos usuários”). Outra versão menciona que a escolha do nome foi inspirada na USENIX, a associação de usuários do Unix, fundada em 1975, que se esperava tivesse um papel fundamental na sua operação.

Como ainda não havia uma Web organizada na época do lançamento da USENET, os artigos tinham que ser copiados de um computador para o outro através de modems, que eram conectados eventualmente. Assim, uma postagem de um usuário originalmente era feita apenas no servidor que ele acessasse. Porém, de tempos em tempos os servidores se comunicavam uns com os outros e enviavam as novas publicações de um para o outro. Ao final de um certo período, todas as publicações acabavam disponíveis em todos os servidores. Isso era razoável em uma época em que as redes eram lentas e nem sempre disponíveis.

A USENET ainda existe e pode ser usada através da Internet, mas sua utilização foi progressivamente substituída por fóruns, blogs e redes sociais.

Com a ajuda de Steve Bellovin, da Universidade da Carolina do Norte, eles implementaram a primeira versão do sistema, chamada “netnews” e, assim, as duas universidades foram as primeiras a terem servidores da USENET. O sistema passou a ser divulgado e distribuído a outras universidades e foi logo rebatizado como “A News”.

Como era uma forma muito barata e efetiva de comunicação, seu uso rapidamente cresceu e em 1981 o sistema foi reprogramado. A nova versão foi batizada como “B News”. Mais tarde, com a continuidade do crescimento da rede, foi criada a versão “C News”.

A USENET iniciou com apenas dois servidores: Duke e UNC. Porém, no final de 1980 já eram 15; em 1981, 150; em 1982, 400; e em 2003, vários milhões.

O atrativo da USENET sempre foi reunir pessoas interessadas em discutir um determinado assunto. Truskott, por exemplo, era altamente motivado por sua paixão pelo xadrez. Assim, o primeiro grupo de discussão da USENET foi sobre xadrez.

Posteriormente, os grupos de discussão foram organizados em oito grandes diretórios, os quais são controlados por regras de administração bem definidas:

- comp.\* - discussões relacionadas a computação.
- humanities.\* - ciências humanas.
- misc.\* - tópicos variados, como por exemplo, misc.education, misc.kids etc.
- news.\* - discussões sobre a USENET em si.
- rec.\* - recreação e entretenimento.
- sci.\* - ciência.
- soc.\* - questões sociais.
- talk.\* - tópicos controversos como religião, política etc.

Além desses oito grandes grupos, há um nono, o alt.\*, que não segue as mesmas regras de organização. Ele foi responsável por muitos problemas enfrentados pelos servidores da USENET,

especialmente a disponibilização de software pirata e pornografia infantil pelos seus usuários. Em função disso, muitos servidores cancelaram o serviço de USENET enquanto outros baniram apenas os diretórios alt.\*. Apesar disso, o tráfego da USENET, mesmo sendo cada vez menor em relação ao total do tráfego na Web, ainda tem crescido ano após ano até os dias de hoje.

## 9.7 MS-DOS – 1980

Em meados de 1980, a IBM estava procurando desesperadamente um sistema operacional para seu novo computador pessoal, o IBM-PC, que estava para ser lançado no ano seguinte. Eles já tinham um entendimento com Bill Gates para que a Microsoft desenvolvesse a linguagem BASIC, que seria usada nesse novo computador. Quando pediram a Gates alguma referência sobre sistema operacional, ele indicou a Digital Research (DR), que havia desenvolvido o CP/M – padrão de mercado para processadores de 8 bits –, e com certeza teria condições de produzir uma nova versão para o processador 8086 de 16 bits a ser usado no PC.

Tivesse esse acordo dado certo, provavelmente a Microsoft não seria a gigante que é hoje no mundo da informática. O fato é que não houve concordância entre a DR e a IBM. Há muito folclore e muitas histórias a respeito desse fato, mas provavelmente a grande maioria é falsa. O que parece fazer sentido é que realmente não houve acordo porque não se chegou a um valor que agradasse às duas empresas. Consta que a IBM teria oferecido 250 mil dólares por um número indeterminado de cópias do CP/M para 8086, mas a DR queria um acordo baseado em *royalties*, ou seja, um pequeno valor por cópia que a IBM vendesse, como era usual nesses casos.

Assim, em outubro, a IBM retorna a Gates solicitando novamente uma referência ou o desenvolvimento de um sistema operacional. Gates então menciona uma empresa de Seattle, a Seattle Computer Products (SCP), que havia desenvolvido um sistema operacional de 16 bits para o 8086, visto que vendiam placas com este processador

e, juntamente com o sistema, cópias do BASIC da Microsoft. O sistema, na época, era chamado QDOS (Quick and Dirty Operating System<sup>4</sup>); ele recebeu esse nome porque foi feito de forma muito rápida, em apenas dois meses. O fato é que a SCP precisava de um sistema operacional para o 8086 e esperava que a DR fizesse uma versão do CP/M para o novo processador. Porém, como essa versão demorou demais, um de seus funcionários, Tim Paterson (Estados Unidos, 1956), foi solicitado a escrever um sistema operacional para as placas 8086 da SCP.

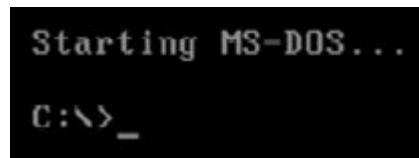
Paterson fez praticamente um clone do CP/M, como ele mesmo recorda. O primeiro passo era escrever código para fazer tudo o que o CP/M fazia, e o segundo seria projetar um sistema de arquivos que fosse rápido e eficiente.

Em julho de 1980 ele já tinha terminado cerca de 50% do código e o batizou de QDOS 0.10. Logo em seguida, achou um bug no sistema que precisou ser corrigido e imediatamente lançou a versão 0.11.

No final do ano, uma versão bastante melhorada, a 0.33 foi lançada e o sistema rebatizado como 86-DOS. Nessa mesma época, Gates adquiriu uma licença não exclusiva do 86-DOS por 25 mil dólares e seis meses depois adquiriu a licença exclusiva por apenas 50 mil dólares, tornando-se assim único proprietário do 86-DOS. O produto foi então rebatizado como MS-DOS.

A Microsoft fez melhorias no sistema e o entregou à IBM em outubro de 1981 como MS-DOS versão 1.10. A IBM efetuou uma análise de qualidade no produto e reportou mais de 300 erros, que tiveram que ser corrigidos. Dessa forma, o produto final acabou tendo seu copyright compartilhado entre Microsoft e IBM. Entretanto, a IBM concordou em assinar com a Microsoft um contrato que previa *royalties*, ou seja, pagaria à Microsoft por cópia vendida. Além disso, o contrato não exigia exclusividade e a Microsoft poderia, se quisesse, vender o sistema diretamente a outras empresas. Ao fazer isso, a IBM mal sabia que assinava sua morte no mercado de PCs anos depois.

O DOS era um sistema baseado em linha de comando e funcionava em tela de texto, sem gráficos. Os primeiros sistemas Windows, lançados anos mais tarde, eram programas executados a partir do MS-DOS. Assim, para rodar uma versão do Windows 3.1, por exemplo, mais de uma década mais tarde, o usuário ligava o computador, carregava o sistema operacional MS-DOS e, a partir da tela escura característica dessa interface, mostrada na [Figura 9.4](#), digitava “win”, o que fazia rodar o sistema Windows, baseado em interface gráfica.



**Figura 9.4:** Tela do MS-DOS.<sup>5</sup>

Versões mais modernas do Windows eliminaram esse passo e passaram a ser carregadas diretamente quando a máquina era ligada. Invariavelmente, contudo, todas elas disponibilizam um aplicativo que simula a antiga tela do MS-DOS, usualmente chamado “Command Prompt”, que pode ser executado mesmo que o computador esteja em modo gráfico.

## 9.8 Xerox 8010 Star – 1981

Pensa que computadores pessoais com interfaces gráficas iniciaram com o Windows? Não! Então, foi Macintosh ou o Lisa da Apple? Também não! O primeiro computador comercial com um sistema operacional com interface gráfica como conhecemos hoje foi o Xerox Star ou Xerox 8010 Information System ([Figura 9.5](#)).



Figura 9.5: Tela do Xerox Star.<sup>6</sup>

O Star era considerado uma estação de trabalho, ou seja, um computador que funcionaria bem em um ambiente corporativo em conexão com outros computadores e dispositivos como servidores de arquivo, impressão e comunicação. Ele foi inspirado no Xerox Alto, e se este não era um produto comercial, o Star era. Tratou-se do primeiro computador a utilizar o conceito de ícones em uma área de trabalho. Sua propaganda afirmava que era o primeiro computador no qual você poderia usar texto e gráficos juntos em um mesmo documento.

O Star foi um dos pioneiros a ter seu sistema desenvolvido a partir de análise de tarefa, uma técnica para desenvolvimento de sistemas

ergonômicos que considera em primeiro lugar quem é o usuário e que tipo de trabalho ele vai fazer com o equipamento; então a melhor interface é definida para facilitar esse trabalho.

A máquina inaugurou o conceito de “desktop”, ou “área de trabalho”, que seria a visão inicial que o usuário tem do sistema. Ela deveria lembrar a visão de uma mesa de escritório, na qual estariam dispostos os documentos e ferramentas de trabalho, que passaram a ser representados por ícones – ou seja, desenhos simplificados, mas fáceis de lembrar, que transmitem ao usuário algum significado ou lembrança.

O mouse seria usado nessa área de trabalho, facilitando a movimentação dos itens e sua “abertura”, o que faria com que o ícone se expandisse em uma janela maior na qual seu conteúdo poderia ser trabalhado. Várias janelas poderiam ser abertas umas sobre as outras e todas estariam sempre acessíveis ao comando do usuário. Até então, a noção de “tela” dos sistemas de informática implicava em apenas uma tela disponível a cada momento. Para acessar outra seria necessário “fechar” a tela corrente, o que com frequência dificultava a interação.

O Star vinha com um sistema de e-mail incorporado e podia ser ligado à Arpanet a partir de uma placa Ethernet. Ele já incorporava os conceitos de “caixa de entrada” e “caixa de saída” que eram representadas por ícones próprios.

O paradigma de interação com o uso do mouse e ícones entrava em contraste com as formas de interação da época baseadas no uso do teclado. O WordStar, por exemplo, exigia combinações da tecla “Control” e letras para realizar operações. Assim, ou o usuário tinha que lembrar que combinações eram essas; ou consultava um manual; ou, como o WordStar fazia, os comandos deveriam ser exibidos na tela, facilitando a lembrança, mas ocupando espaço que poderia ser usado para outras finalidades.

O Star tinha um poderoso sistema de edição de texto e imagens. Letras poderiam ser representadas em diferentes fontes (até então

os editores trabalhavam com uma única fonte, normalmente), em itálico, negrito ou sublinhado. O texto também poderia representar subscritos e sobrescritos em até dois níveis, o que facilitava a edição de fórmulas matemáticas. Havia também uma função para pesquisar e substituir texto considerando, inclusive, as propriedades deste, ou seja, o tipo de formatação aplicado. Diferentes tamanhos de letra também podiam ser utilizados no mesmo texto.

A edição de textos e imagens em tela gráfica levou o Star a ser o primeiro computador totalmente WYSIWYG (“*What You See Is What You Get*”), ou seja, a relação entre o que se via na tela e o que seria impresso era exata.

O sistema de desenvolvimento da Star chamava-se Tajo e era uma versão estendida do Smalltalk-80, recém-lançado.

Porém, apesar de todas as suas características inovadoras e determinantes em relação à maneira como nos comunicamos com nossos computadores hoje, o Star passou praticamente despercebido pela história. Isso provavelmente deveu-se ao fato de ele não ter sido exatamente um sucesso de vendas. Uma das razões era seu alto custo. O Star não foi projetado para ser um computador pessoal, era uma estação de trabalho e, se você quisesse automatizar seu escritório, teria que comprar uns dois ou três dele, e mais um servidor de arquivos. A instalação completa custaria até 100 mil dólares. Considerando que o salário de uma secretária na época era de 12 mil dólares ao ano, não se tratava de uma vantagem econômica muito clara para as empresas. Além disso, havia computadores pessoais bem mais baratos no mercado.

Comenta-se também que o sistema era muito lento. O hardware disponível na época simplesmente não estava à altura de todas as características inovadoras do Star. Salvar um arquivo, por exemplo, poderia levar vários minutos e se algum problema acontecesse (o que não era incomum) haveria um processo de recuperação que poderia levar horas, deixando o computador paralisado durante muito tempo, o que era inaceitável.

Porém, o legado do Star não se perdeu. Alguns membros da equipe do projeto Apple Lisa conheceram o Star em 1981 e isso provocou uma reviravolta no Lisa, que passou a considerar o uso de uma interface com usuário semelhante à do Star.

## 9.9 IBM PC – 1981

Como já foi visto, no final dos anos 1970 a IBM começou a se interessar pelo mercado de microcomputadores. Ela dominava o negócio de computadores médios e grandes, mas perdia feio no que se referia aos micros. A primeira iniciativa, como sabemos, foi o IBM 5100, lançado praticamente junto com microcomputadores de outras empresas. Porém, essa máquina ainda era pesada e cara. Seu custo de cerca de 20 mil dólares não o tornava competitivo frente aos computadores da Apple, Tandy e Commodore do final dos anos 1970. Assim, a IBM não emplacava neste mercado.

A empresa resolveu iniciar, então, um projeto secreto de um microcomputador que inicialmente se chamaria “Acorn”, referenciado internamente como Projeto Chess. Ele foi realizado na fábrica de Boca Ratón, na Flórida, por uma equipe de 12 engenheiros liderada por William C. Lowe (Estados Unidos, 1941-2013).

Para o processador de 16 bits havia três escolhas: o Motorola 68000, e os chips da Intel 8086 e 8088. Foi escolhido o processador menos potente, o 8088, mas que em termos de código de máquina era totalmente compatível com o 8086.

O PC foi inicialmente denominado de IBM 5150, para colocá-lo na série de computadores portáteis iniciada pelo 5100. Entretanto, como seu projeto e arquitetura eram bem diferentes de tudo o que a IBM já tinha feito, quando foi anunciado em agosto de 1981 foi rebatizado como “IBM PC” (Personal Computer ou Computador Pessoal). A [Figura 9.6](#) mostra um IBM 5150 ou PC, com monitor CGA de 16 cores (modelo 5153) e uma impressora de coluna matricial IBM 5152.



**Figura 9.6: IBM PC.<sup>7</sup>**

Houve, porém, muitas objeções ao projeto dentro da própria IBM. Alguns acreditavam que microcomputadores não tinham nada a ver com automação de escritórios e que a IBM entrando nesta seara só colheria desilusão e vergonha. Outros diziam que a empresa, sendo uma gigante, não poderia se envolver nesse tipo de projeto porque ensinar a IBM a fazer computadores pessoais seria “como ensinar um elefante a dançar sapateado”.

Além disso, até o momento, o seu computador mais barato custava 15 mil dólares e todas as vendas da IBM eram feitas pelos seus próprios vendedores, que atendiam diretamente aos clientes corporativos. A IBM não tinha qualquer experiência ou estrutura para fazer vendas no varejo.

Porém, o projeto foi levado adiante. O pequeno IBM PC tinha apenas 16 kB de memória inicial a um custo de 1.500 dólares. Era uma máquina robusta, especialmente se considerarmos seu teclado e monitor de alta qualidade. Porém, a principal razão do seu sucesso não se discute: foi a marca IBM. Uma revista da época chegou a anunciar que havia chegado o “IBM dos

microcomputadores". Um ditado popular nas empresas na época era que "ninguém nunca foi demitido por comprar um IBM"; isso caracterizava bem o respeito que a marca tinha. Foi registrada uma marca de 40 mil unidades do PC vendidas apenas no primeiro dia.

O PC foi lançado com o sistema operacional MS-DOS e BASIC desenvolvidos pela Microsoft. Entre os aplicativos iniciais também havia uma versão do VisiCalc e o jogo *Adventure*, ou *Colossal Cave Adventure* (o primeiro game para PC).

A máquina, porém, ainda não contava com unidade de disco rígido interna. Ela poderia ter até dois drives de disquete de 5½ opcionais, que a encareciam um pouco. Era possível trabalhar com gravadores de fita K7 ou discos rígidos externos de 10 MB com fonte de alimentação própria.

Entretanto, o fator preço ainda pesava contra o PC, pois ele era significativamente mais caro que seus concorrentes, apesar de contar com uma configuração inicial que não incluía monitor nem drives de disquete. Porém, como a IBM divulgou amplamente as características internas da máquina, muitas empresas ao redor do mundo passaram a construir clones mais baratos do PC, o que contribuiu para a dominação do padrão PC no mercado. Assim, pessoas ou empresas com menos recursos optavam pelos clones, enquanto que aqueles que adquiriam uma marca pela sua confiabilidade compravam IBM.

## 9.10 Placa de Video CGA – 1981

Juntamente com o IBM PC foi lançada em 1981 uma placa para vídeo chamada CGA, ou Color Graphics Adapter. Ela era o adaptador de vídeo padrão para o PC e acabou sendo a ancestral de muitos outros modelos como a EGA, VGA e SVGA.

A CGA foi, de fato, o primeiro padrão para display de vídeo colorido de computadores. A placa, que funcionava independentemente da CPU embora fosse conectada a ela, tinha um microprocessador Motorola MC6845, dedicado ao processamento de vídeo e uma

memória de vídeo de 16 kB. Sua saída podia tanto ser ligada a um monitor colorido RGBI (Red Green Blue Intensifier), quanto a uma televisão NTSC via saída RCA analógica, mas neste caso necessitava de um adaptador.

A [Figura 9.7](#) mostra uma das primeiras placas CGA na qual se pode ver na extremidade direita os conectores RCA no alto (imagem e som) e RGB mais abaixo. Na extremidade inferior da placa, no lado direito, pode-se ver o barramento de contatos elétricos que era encaixado em um *slot* adequado na placa-mãe do PC. O maior dos circuitos integrados no centro inferior da placa é o processador de vídeo Motorola 6845.

A resolução máxima da CGA era de 640 colunas por 200 linhas, mas nessa resolução ela só exibia duas cores. Outro modo trabalhava com resolução de 320 x 200 e, com ele, era possível mostrar quatro cores de uma vez. Isso não quer dizer que eram sempre as mesmas cores porque se podia escolher uma dentre quatro *palettes*, ou seja, podia-se mostrar até 16 cores, mas apenas quatro de cada vez na tela. Um modo gráfico estendido aumentava o número de cores simultâneas para 16, mas neste caso a resolução caia para 160 colunas por 100 linhas.

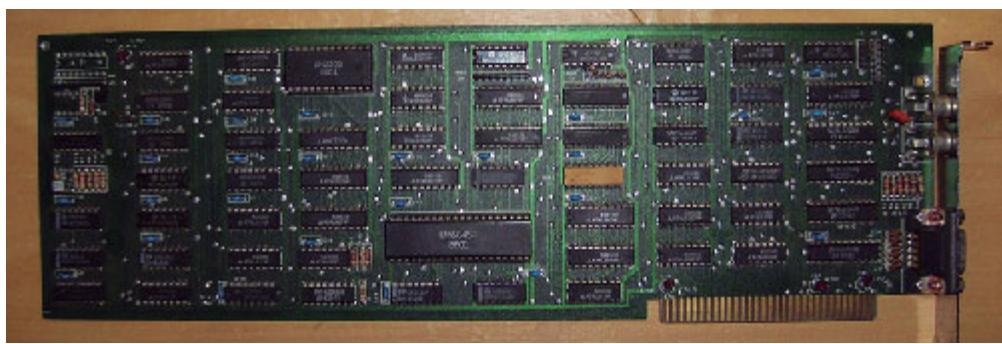


Figura 9.7: Placa CGA.<sup>8</sup>

As 16 cores eram obtidas pelas combinações binárias de 4 bits, um para vermelho, um para verde e um para azul, além do bit de intensificação que permitia gerar cores mais claras ou mais escuras.

A intensificação agia, portanto, sobre todas as cores combinadas. Cada cor então tinha uma configuração escura e uma clara, respectivamente:

- Preto e cinza.
- Azul e azul claro.
- Verde e verde claro.
- Ciano (azul esverdeado) e ciano claro.
- Vermelho e vermelho claro.
- Magenta (uma espécie de lilás) e magenta claro.
- Marrom e amarelo.
- Cinza claro e branco.

Observa-se que, pela lógica, deveria haver um amarelo escuro e um claro, mas como o amarelo escuro não é muito agradável à vista e o marrom é uma cor muito mais usada e que estaria fora de todas as *palettes*, os projetistas das placas CGA optaram por implementar uma condição. Quando a cor tivesse o código 6, que corresponderia a amarelo escuro, a quantidade de verde seria diminuída pela metade, gerando marrom em vez de amarelo escuro.

O padrão CGA foi bastante usado no início dos anos 1980 até ser sucedido por EGA e VGA no final dessa década. Havia poucas opções de placas gráficas durante este tempo, uma delas, que chegou a ser usada no Brasil, foi a placa Hércules, que não era colorida, mas permitia uma resolução mais alta com 720 colunas por 348 linhas em modo monocromático.

## 9.11 Revista Micro Sistemas – 1981

A *Micro Sistemas* foi a primeira revista de microinformática editada no Brasil. Sua história começou quando no final de 1980 Aldenor Campos compreendeu a importância do mercado de microcomputadores e resolveu que ia explorá-lo no Brasil. Ele já tinha uma empresa que funcionava como bureau de programação e cujo principal cliente era o DNER (Departamento Nacional de

Estradas de Rodagem). Resolveu criar mais duas empresas: uma fábrica de clones do Apple chamada Del e uma loja de computadores, a Computique.

Porém, as vendas de microcomputadores não foram bem como Campos esperava. Assim, ele chamou sua filha, Alda Campos, na época recém-formada em jornalismo, e deu a ela uma pilha de revistas de computação norte-americanas. Ele lhe disse que o Brasil não tinha nada assim e que para vender microcomputadores precisavam criar uma cultura; dessa forma, a publicação de uma revista de microcomputadores em português se fazia necessária, e como Alda era formada em jornalismo esse seria um trabalho perfeito para ela.

De início, ela procurou ler as revistas americanas e conversou com jornaleiros sobre a demanda por esse tipo de publicação. Depois, convidou Wayne Green (1922-2013), editor de várias revistas de computação, para vir ao Brasil e trabalhar como consultor nos primeiros meses do periódico.

Em agosto de 1981 foi lançada a primeira edição da revista em uma feira da Associação dos Usuários de Informática e Telecomunicações (SUCESU) em São Paulo. Alda conta que nem ela nem os demais editores entendiam muita coisa de microcomputadores, porque quando se formaram não havia tais dispositivos disponíveis no Brasil. Assim, foram aprendendo por tentativa e erro.

A revista incluía seções que apresentavam notícias e programas, especialmente jogos, para as principais linhas de microcomputadores, como Apple, TRS-80 e Sinclair. Recebia muitas cartas de leitores e procurava responder a todas. Outra coisa que fizeram foi criar um laboratório para testar equipamentos. No início foi difícil, pois as empresas não cediam equipamentos facilmente nem mesmo por publicidade. Além disso, Alda era muito jovem e comenta que muitas vezes era difícil ser levada a sério pelos empresários ou mesmo escapar de paqueras indesejadas. Assim,

após comprarem alguns equipamentos, o laboratório começou a funcionar e depois de algum tempo, com a popularização da revista, fabricantes passaram a disponibilizar seus equipamentos para avaliação.

No final de 1986, após se desentender com um de seus colaboradores, Renato Degiovani, que depois se tornaria o novo editor da revista, Alda se despede da *Micro Sistemas* para trabalhar em uma editora concorrente, que havia lançado a revista *Micro Mundo* e a *Datanews*, hoje chamada de *Computerworld*. Em sua despedida, escreve o editorial “Bye, Byte”, de grande repercussão, fazendo com que ela se impressionasse com o impacto que a publicação teve na escolha de carreira de muitos brasileiros. De fato, o autor deste livro recorda-se de ter sido assíduo leitor da *Micro Sistemas* entre 1983 até 1985, quando iniciou seu Bacharelado em Ciência da Computação. Com um TK 83 e depois um CP 200 para trabalhar em casa, a *Micro Sistemas* foi certamente uma agradável inspiração e determinante na escolha da carreira em computação. O maior choque foi chegar à universidade e descobrir que ali praticamente só existia a cultura mainframe IBM e, para dificultar ainda mais, os primeiros PCs só chegaram ao final do curso, por volta de 1987-88.

Renato e Alda acabaram fazendo as pazes, mas seguiram em empresas diferentes. Com Renato à frente da *Micro Sistemas*, a revista começou a se especializar em jogos do tipo Adventure, que Renato gostava e que costumava agradar também aos jovens hobistas da época pela sua semelhança com os RPG (*Role Playing Games*), que estavam na moda.

## 9.12 Osborne 1 – 1981

A indústria de computadores portáteis andava apenas um pouco atrás da indústria de computadores pessoais. Para deixar claro, um computador pessoal é um que você pode ter em casa, já um

computador portátil é um que você pode carregar com facilidade para todo lado.

O Osborne 1 ([Figura 9.8](#)), foi um destes computadores portáteis. Possivelmente o primeiro portátil completo a ser vendido comercialmente. Havia computadores menores e mais leves, mas eles não incluíam o monitor nem as unidades de disquete. O Osborne 1 tinha tudo isso: ele não era exatamente um laptop, mas caberia facilmente debaixo de um assento de avião. Por este motivo, era classificado como “*luggable*”, que não tem tradução exata em português, mas se formos criar um neologismo seria algo como “bagageável”, ou seja, podia ser carregado como bagagem de mão.



Figura 9.8: Osborne 1.<sup>9</sup>

Seu criador, Adam Osborne (Tailândia, 1939-2003), autor de livros e software, fundador da Osborne Computer Company, é descrito pelo historiador Georgi Dalakov como “a pessoa mais charmosa,

persuasiva, egoísta, e absolutamente confiante de toda a indústria da computação, aliás, de toda a indústria”.

O projeto foi claramente inspirado no Xerox NoteTaker. O Osborne era um computador completo, bastava ligá-lo à tomada para funcionar. Bem, ele não tinha bateria, mas tinha teclado, monitor de 5 polegadas e duas unidades de disquete, além das conexões usuais para outros periféricos.

Seu peso era de 10 quilos, ou seja, tratava-se de um portátil pesado, e seu gabinete podia ser facilmente fechado no formato de uma maleta, com alça e tudo. O próprio gabinete reforçado o protegia contra choques. Admite-se que o computador poderia sobreviver mesmo se caísse de uma mesa ou da mão de seu transportador.

O preço inicial era de 1.795 dólares, considerado barato, porque a máquina já vinha com cerca de 1.500 dólares em software, incluindo o WordStar, dBASE II e outros produtos para CP/M. O computador foi um sucesso por alguns anos, chegando a vender 10 mil unidades por mês em um período. Seu principal competidor era o Kaypro, outro portátil, mas com uma tela de 9 polegadas, quase o dobro do Osborne.

Seu declínio ocorreu abruptamente por um erro de marketing que até hoje é conhecido na indústria de computadores como “Efeito Osborne”. Esse erro consistiu em anunciar publicamente o lançamento de um sucessor superior na mesma faixa de preço muito tempo antes de ser capaz de lançá-lo efetivamente. No caso, a empresa anunciou o Osborne Executive, sucessor do Osborne 1, cerca de um ano antes de estar pronta para lançá-lo. O efeito foi que as vendas do Osborne 1 imediatamente caíram para praticamente zero, e no final de 1983 a empresa, sem recursos para bancar o desenvolvimento do Executive, declarou falência.

A Osborne ainda conseguiu se recuperar pouco depois e lançar não só o Executive como também o Vixen, mas seus produtos a essa altura já tinham fama de “vaporware” e nunca mais atingiram os

mesmos níveis de antes. Assim, não conseguiram levantar a empresa da falência.

Adam Osborne retornou para a Índia, onde tinha vivido em sua juventude, nos anos 1990 e lá iniciou uma empresa de software.

## 9.13 Rational Machines – 1981

A empresa Rational Machines teve um papel muito importante no desenvolvimento de notações, métodos e ferramentas para desenvolvimento de software. A partir dela, nos anos 1990, surgiram a notação UML (Unified Modeling Language) e o processo UP (Unified Process). Porém, a empresa foi fundada muito antes disso, em 1981. Seus fundadores foram Paul Levy e Mike Devlin (Estados Unidos, 1969).

Em 1985, a empresa lançou o Rational Environment, um conjunto de ferramentas de produtividade para desenvolvimento de sistemas em Ada, mas ainda sem capacidade de modelagem visual. O ambiente foi projetado para rodar em uma estação de trabalho especialmente projetada para ele, a R1000, que também era vendida pela empresa, ou seja, a ideia era você comprar estações de trabalho já preparadas para desenvolvimento eficiente de software em Ada. Ela tinha arquitetura otimizada tanto para rodar as ferramentas de produtividade, quanto os programas em Ada.

Em 1990, a Rational lançou três projetos:

- Reimplementar o Rational Environment para Ada em Unix, o qual foi lançado em 1993.
- Implementar o Rational Environment para C++ também em Unix, o qual foi lançado em 1994.
- Desenvolver uma ferramenta para modelagem visual de sistemas usando a notação de objetos proposta por Grady Booch (Estados Unidos, 1955). A versão 1.0 dessa ferramenta, chamada Rational Rose foi lançada durante a conferência da

ACM sobre orientação a objetos, a OOPSLA (Object-Oriented Programming, Systems, Languages & Applications) em 1992.

Infelizmente, a performance do Rose era muito limitada de várias maneiras e ela não foi mantida como produto comercial por muito tempo. A empresa então se dedicou a produzir uma nova versão, a Rational Rose 2.0, que foi construída a partir de uma ferramenta de modelagem chamada Object System Designer, adquirida da empresa Palladio. Essa ferramenta também empregava a notação de Booch, que representava classes como pequenas nuvens tracejadas. O Rose 2.0 foi um sucesso maior, pois entre outras coisas permitia a realização de engenharia reversa, ou seja, o programador podia alimentar a ferramenta com um programa desenvolvido em C++ e ela construía o diagrama de classes que documentaria a situação atual do programa. Esse diagrama poderia então ser alterado na ferramenta pelo uso de técnicas de modelagem visual, e então o código C++ poderia ser novamente gerado, agora refletindo a nova organização.

Em 1994, a Rational Machines fundiu-se com a Verdix Corporation, uma empresa pública que produzia um grande conjunto de ferramentas para Ada. Nessa ocasião, ela foi rebatizada como Rational Software Corporation, passando a dar mais ênfase ao desenvolvimento de software do que de hardware.

Em 1995, James Rumbaugh (Estados Unidos, 1947), outro metodologista já famoso pela notação e processo OMT (Object Modeling Technique), juntou-se à Rational. Grady Booch (Estados Unidos, 1955) já estava na empresa, e em 1996 eles adquiriram da Ericsson a companhia Objectory AB, fundada por Ivar Jacobson (Suécia, 1939). Foi neste momento que historicamente a notação de representação de requisitos conhecida como “casos de uso” juntou-se às notações de classes de objetos propostas por Booch e Rumbaugh. Jacobson já tinha uma notação para representar classes e objetos que usava círculos estereotipados. A notação de Booch, como vimos, utilizava pequenas nuvens. Porém, no final, a notação para classes e objetos escolhida para compor a notação

“unificada” dos três metodologistas foi a de Rumbaugh: simples retângulos, mais fáceis de desenhar e de representar. Os três especialistas passaram, então, a desenvolver a UML, ou Unified Modeling Language, uma linguagem de notação gráfica que hoje, com 13 diferentes diagramas, é considerada um padrão de direito e de fato para modelagem visual de sistemas. Eles também passaram a ser conhecidos como “*The Three Amigos*”<sup>10</sup>, em referência ao filme homônimo de faroeste lançado na época (*Três amigos!*, no Brasil).

Após o lançamento da UML 1.0 em 1997, a linguagem foi oferecida à OMG, Object Management Group. A OMG é uma associação fundada em 1989 por 11 empresas e seu lema é “*We Set the Standard*”.<sup>11</sup> Hoje em dia, UML é um dos principais padrões da OMG.

Ainda em 1996, quando a Rational adquiriu a empresa Objectory de Jacobson, eles tomaram o processo de desenvolvimento de software proposto por ele, substancialmente mais eficiente do que os de Rumbaugh e Booch, e o rebatizaram como ROP (Rational Objectory Process). Posteriormente, seu nome foi alterado para RUP (Rational Unified Process) quando características de outros processos foram adicionadas, além da experiência de 15 anos da própria Rational em desenvolvimento de sistemas e consultoria a clientes.

A criação do RUP esteve a cargo de Philippe Kruchten (Canadá, 1952), engenheiro da Rational. O RUP foi definido na forma de um hipertexto no qual se pode navegar e obter informações sobre como realizar processos, quais documentos gerar e que perfis de pessoas devem ser encarregados de cada atividade. O RUP é um processo prescritivo, ou seja, ele apresenta instruções sobre como fazer o desenvolvimento de software. Posteriormente, foi conceitualizado como um conjunto de ideias chamado UP (Unified Process) a partir do qual vários outros modelos se derivaram, como EUP (Enterprise Unified Process) e OUM (Oracle Unified Method). Inclusive versões

ágeis, não prescritivas, foram criadas no início do século XXI, como AUP (Agile Unified Process) e OpenUP, um projeto da Fundação Eclipse.

A Rational era uma empresa influente e lucrativa. Ela estava de fato criando as bases para o desenvolvimento de software no século XXI. Em seu auge, chegou a ter 4 mil funcionários e faturar 850 milhões de dólares por ano, mas posteriormente foi comprada pela IBM entre 2002 e 2003 por 2,1 bilhões de dólares. Hoje os produtos Rational, inclusive o RUP e suas muitas ferramentas de produtividade, são mantidos e comercializados pela IBM.

## **9.14 Sun Microsystems Inc. – 1982**

A Sun Microsystems Inc., fundada no início de 1982, foi uma empresa de características singulares. Em primeiro lugar, ela focava em computação em rede, sendo seus principais produtos estações de trabalho com alta capacidade gráfica que operavam em rede. Isso era uma noção bem diferente da ideia de terminais de mainframes ou de computadores pessoais isolados.

Em segundo lugar, a Sun fomentou uma nova forma de trabalhar com produtos de software. No início dos anos 1980, a maioria das empresas de software procurava lucrar com a venda de licenças de seus produtos, mas a Sun resolveu distribuí-los gratuitamente, a começar pelo seu sistema operacional, o SunOS, e depois Solaris, ambos baseados em Unix.

Ela também foi responsável nos anos 1990 pela criação do Java, uma linguagem de programação gratuita que contrastava com a maioria das outras da época, pelas quais você precisava pagar para ter uma licença de uso do compilador e ambiente de programação. Finalmente, a Sun também teve um papel importante no enfrentamento do monopólio da Microsoft em automação de escritório ao adquirir, no final dos anos 1990, a empresa que fabricava o StarOffice, concorrente direto do Microsoft Office. Logo

em seguida, a Sun lançou esse software gratuitamente no mercado com o nome de OpenOffice.

Na área de banco de dados, a empresa adquiriu por 1 bilhão de dólares a desenvolvedora do MySQL, para em seguida liberar o produto gratuitamente. Não é à toa que a Sun e a Microsoft eram consideradas arqui-inimigas.

A Sun nasceu a partir de um projeto de Andreas Bechtolsheim (Alemanha, 1955), que fazia pós-graduação na Universidade de Stanford. Ele projetou aquilo que seria uma estação de trabalho, ou seja, mais do que um computador pessoal ou terminal, um computador que pudesse usufruir intensamente de recursos de rede como servidores de arquivos e impressoras, e que também tivesse grandes monitores com alta capacidade gráfica, de forma a atender às necessidades de projetistas e engenheiros. O computador não seria construído com tecnologia própria, como a maioria das empresas de então fazia, mas com partes que pudesse ser compradas no mercado, o que tornaria seus custos mais baixos.

As ideias de Bechtolsheim atraíram a atenção de outros colegas de Berkeley e eles acabaram fundando a Sun Microsystems Inc. em 1982. O nome SUN era a sigla de “Stanford University Network”. Em função das características de suas estações de trabalho e do seu preço, a empresa já começou vendendo muito e em apenas seis meses dava lucro. As primeiras estações, Sun-1 ([Figura 9.9](#)) e Sun-2 foram vendidas principalmente para universidades e faturaram 8 milhões de dólares já no primeiro ano.



**Figura 9.9: Estação de trabalho Sun-1.<sup>12</sup>**

No segundo ano de operação, a Sun ficou sabendo que a ComputerVision, uma empresa líder no desenvolvimento de sistemas CAD (Computer Aided Design<sup>13</sup>), estava disposta a desistir de seu hardware próprio em função de outro fornecedor. Tratava-se de uma grande oportunidade para a Sun, mas a ComputerVision já havia assinado um contrato com outro fornecedor, a Apollo. Isso não fez os executivos da Sun desistirem e, após várias abordagens intensas, a ComputerVision desfez o contrato e optou pela Sun

como plataforma para seu produto. Esse gesto fez com que o mercado visse na Sun uma empresa agressiva, a qual não se poderia menosprezar.

Em 1984, ela introduziu e licenciou gratuitamente a tecnologia NFS (Network File System<sup>14</sup>) para arquivos em rede de computadores, que se transformou em padrão da indústria. Nos anos 1980, a Sun foi a empresa de informática que mais cresceu no mundo, seu faturamento chegando à casa dos bilhões em 1988.

Porém, no início do século XXI, com o estouro da bolha das “.com”, a Sun foi fortemente afetada e viu suas vendas caírem. Em 2004, após uma série de processos fracassados contra a Microsoft, as duas empresas resolvem fazer uma trégua. A Microsoft pagou à Sun 2 bilhões para acabar com a briga e as duas companhias passaram a trabalhar juntas para tornar suas tecnologias mais compatíveis.

Em 2009, a Sun Microsystems finalmente foi vendida para a Oracle. Um dos seus parques industriais, com 1 milhão de metros quadrados em Menlo Park, na Califórnia, foi vendido no início de 2011 para se tornar o quartel general do Facebook.

## 9.15 Commodore 64 – 1982

O ano era 1982 e a Commodore não sossegava. Dois anos após lançar o VIC 20, ela introduz o Commodore 64, que às vezes também era chamado de VIC-64. Tratava-se de um computador com processador de 8 bits MOS 6510 e 64 kB de memória (daí seu nome). Tudo bem, mas por que é digno de nota falar sobre mais um computador da Commodore? Bem, porque este entrou para o *Livro Guinness dos Recordes* como o computador mais vendido na história. Estima-se que até 17 milhões de unidades tenham sido vendidas. Entre 1983 e 1986, ele dominou o mercado de microcomputadores. Talvez você nunca tenha visto um porque na época o Brasil ainda estava sob a Lei de Reserva de Mercado e a importação de microcomputadores como este era proibida.

Ele era produzido em linha. Até 400 mil unidades podiam ser fabricadas por mês e isso ocorreu por um período de dois anos. Além disso, havia mais de 10 mil títulos de software disponíveis para o Commodore 64.

O projeto dessa máquina foi uma grande aposta da empresa, pois quando eles se comprometeram a desenvolver um microcomputador com 64 kB de memória, o preço destas ainda era muito alto, cerca de 100 dólares. Porém, eles apostaram que no tempo necessário para desenvolver o projeto e até a data de lançamento do produto o preço delas já teria baixado, o que de fato aconteceu.



**Figura 9.10: Commodore 64.<sup>15</sup>**

Comenta-se que quando o 64 foi lançado em uma feira de eletrônicos o pessoal da Atari ficou de queixo caído, pois não entendiam como conseguiam vender o produto por apenas 595 dólares. A verdade era que a Commodore poucos anos antes tinha comprado a MOS e, logo, ela própria produzia os

microprocessadores. Assim, o preço de produção de um 64 não passava de 135 dólares.

A produção do Commodore 64 só foi descontinuada em 1995, após 13 anos de sucesso absoluto.

## 9.16 Arquitetura RISC – 1982

No ano de 1982, uma nova vertente na construção de microprocessadores foi lançada, a RISC, ou Reduced Instruction Set Computing. A ideia era criar microprocessadores a partir de uma filosofia diferente. Ao invés de usar um grande conjunto de instruções implementadas em hardware, o processador usaria um conjunto muito pequeno de instruções básicas. Instruções mais complexas, em vez de serem implementadas em hardware, seriam executadas como combinações das instruções básicas. Dessa forma, as instruções mais complexas seriam executadas mais lentamente do que se fossem implementadas em hardware. Por outro lado, e essa era a grande sacada, a simplificação do circuito permitiria que se criasse um processador tão rápido que essa desvantagem acabaria desaparecendo e sendo compensada, resultando em um processamento ainda mais veloz.

Essas ideias, claro, não surgiram do dia para a noite e tampouco em um único lugar. Foi na Universidade de Berkeley que o termo foi cunhado a partir de um artigo sobre um curso de Carlo H. Séquin (Suíça, 1941) e David Patterson (Estados Unidos, 1947), publicado como relatório técnico em 1982. O projeto de um novo processador com essa filosofia, chamado RISC I, foi desenvolvido por alunos de pós-graduação em uma disciplina na qual avaliavam ideias arquiteturais e projetavam novos chips VLSI (Very-Large-Scale Integration).

A ideia era que os novos circuitos VLSI não deviam apenas replicar em menor escala o projeto arquitetural dos antigos computadores. Eles deveriam ser totalmente repensados. Um dos maiores problemas ao se colocar todo um circuito processador em um chip é

a distância entre contatos, pois isso implica em perdas de tempo e aquecimento. Assim, circuitos mais simples com conexões mais curtas podem ser mais rápidos do que circuitos complexos com conexões longas. O conjunto reduzido de instruções de RISC ajudou a minimizar a distância entre as conexões.

Uma das características dessa arquitetura era que a grande maioria das instruções eram implementadas apenas no processador, envolvendo unicamente seus registradores. Instruções de acesso a memória (mais lentas) seriam apenas LOAD e STORE. Dessa forma, com um conjunto grande de registradores internos (128, por exemplo), o programa poderia carregar seus dados da memória para os registradores apenas uma vez, processá-los internamente e armazenar os resultados na memória somente ao final do processamento. Se a cada instrução o processador tivesse que pegar dados na memória ou armazená-los lá novamente, ele perderia muito tempo.

Outra consideração que aponta a vantagem desse tipo de arquitetura foi a constatação de que nos processadores CISC (Complex Instruction Set Computing), anteriores, cerca de 70% das instruções eram pouco ou nada usadas, mas estavam no circuito, ocupando espaço e tornando todo o processo de computação mais lento. Assim, era chegada a hora de fazer essa limpeza.

O grupo de Stanford criou dois protótipos: o primeiro, RISC I, foi chamado de “Gold”. Com as lições aprendidas, um novo design foi feito, resultando no RISC II, ou “Blue”, o qual acabou virando o padrão da indústria e praticamente sinônimo de RISC.

Paralelamente, a IBM investia em uma ideia similar com o IBM 801, mas como o projeto foi um fracasso comercial, teve pouco impacto a não ser pelo fato de que inspirou anos mais tarde a criação dos processadores PowerPC.

Por outro lado, a Universidade de Stanford também vinha desenvolvendo um projeto semelhante, denominado MIPS. O MIPS foi conduzido por John L. Hennessy (Estados Unidos, 1952) da

Universidade de Stanford, que ficou famoso por seu livro de arquitetura de computadores, considerado um clássico. O MIPS foi finalizado em 1983 e rodou os primeiros programas em 1984.

Comercialmente, o uso de processadores com arquitetura RISC só iniciou de fato em 1986, quando a HP o utilizou em alguns de seus computadores. Mais importante, porém, foi que em 1987 a SUN começou a produzir em massa estações de trabalho com seu processador RISC chamado SPARC. O processador das SPARC Station era, portanto, fortemente baseado no projeto do RISC II de Berkeley. O sucesso das SPARC reavivou o interesse da IBM na arquitetura RISC, o que resultou nos PowerPC da década de 1990.

## 9.17 BBC Micro – 1982

A ideia de que computadores fossem instrumentos de aprendizagem nas escolas é relativamente antiga. O advento dos microcomputadores possibilitou colocar em prática essa ideia a um preço acessível para as escolas. Com esse objetivo em mente, várias empresas ao redor do mundo criaram modelos de computadores específicos para uso no ensino.

Merece destaque uma das primeiras iniciativas de amplo sucesso realizada no Reino Unido, que foi o BBC Micro ([Figura 9.11](#)). A BBC (sigla para British Broadcasting Corporation) é uma empresa de televisão britânica, que encomendou à companhia Acorn a produção de um computador que teria como objetivo criar uma cultura de computação (*computer literacy*) nas escolas do Reino Unido.



Figura 9.11: BBC Micro.<sup>16</sup>

A Acorn foi fundada em Cambridge em 1979 por dois ex-funcionários da Sinclair. Inicialmente, a empresa funcionou em um espaço emprestado pela Sinclair, mas ironicamente, pouco depois da criação daquela, esta resolveu entrar no mercado de microcomputadores também, tornando-se a sua maior competidora. A Acorn usava em seu projeto o MOS 6502, enquanto que a Sinclair, como já vimos, apostou no Z80.

O primeiro produto da Acorn foi um kit de computador com apenas 512 bytes de memória, o System-1. Em 1980, lançaram o System-2, com mais memória, pensado para ser de fato um computador doméstico. Após venderem 10 mil unidades, puderam pagar o desenvolvimento de uma máquina ainda melhor, o System-3 e, paralelamente, o Acorn Atom, uma simplificação do System-3 que usava fita K7, por exemplo, no lugar dos bem mais caros drives de disquete. Cerca de 20 mil Atom foram vendidos.

Em 1981, entretanto, perceberam que os dias do computador de 8 bits estavam contados. Além disso, a MOS havia sido adquirida pela Commodore, competidora direta da Acorn.

Como não havia disponibilidade fácil de processadores melhores e baratos, eles decidiram lançar o novo modelo, o Proton, basicamente igual ao System-2, mas com gráficos e som superiores.

Nessa época, a BBC começou a se interessar pelo mercado de microcomputadores após ter televisionado um documentário bastante impactante, *The Mighty Micro*,<sup>17</sup> com o Dr. Christopher Evans, o qual previa o crescimento e o domínio do microcomputador na sociedade, empresas e escolas. A BBC resolveu então atuar diretamente neste sentido, lançando o BBC Computer Literacy Project, cujo logo era uma corujinha que pode ser vista do lado direito do console na [Figura 9.11](#). O projeto incluiria vários programas de TV e o lançamento de um computador especial com a finalidade de educar crianças e adolescentes em informática, preparando a nova geração, que seria a primeira a ter amplo acesso a esse tipo de tecnologia.

A BBC pediu para a Sinclair especificar os requisitos desse computador, ao que esta descreveu requisitos idênticos aos de um de seus modelos, o NewBrain, recém-lançado. Ele era um computador bastante primitivo se comparado ao Atom – e ainda mais, se comparado ao Proton –, mas era barato.

A BBC lançou uma chamada pública para a produção do BBC Micro e, ironicamente, a Sinclair não conseguiu se qualificar, pois não tinha como dar atendimento à alta demanda prevista.

Chris Curry (Reino Unido, 1946), da Acorn, resolveu então construir vários protótipos do Proton para que a BBC pudesse avaliar. Ele teria pedido até ajuda dos alunos de Cambridge para ajudar a montar os computadores.

No final, o Proton competiu com um modelo da Sinclair e outro da empresa Dragon, que produzia um clone do Tandy TRS-80. Porém,

o Proton era muito superior aos demais, especialmente em relação aos gráficos e à capacidade de comunicação por rede – característica que os competidores não tinham. Assim, ele foi escolhido como o computador da BBC e rebatizado como BBC Micro.

Escolas primárias, secundárias e universidades britânicas receberam vários destes computadores, que eram interligados em rede. Milhares de crianças e adolescentes aprenderam BASIC nessa época e acabaram se interessando depois por carreiras na área de computação e tecnologia. Uma enorme comunidade de praticantes surgiu, erguendo a indústria de informática do Reino Unido a patamares nunca antes vistos.

Apenas no primeiro ano foram vendidos 24 mil BBC Micros, o dobro do planejado. Até o final do projeto, 1 milhão de micros tinham sido produzidos e comercializados. Dizem que os escritórios da Acorn ficavam sotterrados pelos crescentes pedidos que, com dificuldade, eram atendidos.

## **9.18 GRiD Compass 1101, o Primeiro Notebook – 1982**

Se o Osborne 1 foi o primeiro computador portátil comercial de sucesso, o primeiro notebook com o formato que hoje conhecemos – ou seja, que abre e fecha como uma ostra – foi lançado um ano depois, em 1982. Seu nome era GRiD Compass.

Ele foi projetado a partir de 1979 por William Moggridge (Reino Unido, 1943) para a GRiD Systems Corporation, de Mountain View, na Califórnia. O modelo anunciado foi o 1100, mas ele nunca existiu e acabou sendo lançado o 1101. O projeto foi claramente inspirado no Dynabook de Alan Kay, embora o design original de Kay mais se assemelhasse a um tablet. O importante, porém, era que o 1101 tinha muitas das funções previstas por Kay. Aliás, Moggridge também ficou conhecido por suas contribuições à área de IHC (Interface Humano-Computador) nos anos seguintes.

O preço era salgado, mais de 8 mil dólares. Seu mercado-alvo eram altos executivos, militares e governo. Consta que ele teria sido usado como notebook em algumas missões dos ônibus espaciais da NASA. A [Figura 9.12](#) mostra um desses computadores flutuando no espaço dentro da Discovery e o astronauta John O. Creighton (Estados Unidos, 1943) mostrando um desenho do Sr. Spock no pequeno monitor de 6 polegadas. O monitor, como se pode ver, era monocromático eletroluminescente em cor âmbar. Esse tipo de monitor era construído colocando-se uma camada de material que emite luz quando atravessado por corrente elétrica entre duas camadas de condutores. Tubos CRT certamente não funcionariam muito bem com notebooks, porque não poderiam ser tão finos.

O computador tinha um processador 8086 com coprocessador matemático 8087, 256 a 512 kB de memória RAM e mais 384 kB de memória magnética interna não volátil, ou seja, que não se apagava quando a máquina era desligada.

Ele não tinha bateria e para funcionar precisava estar ligado na tomada, o que, infelizmente, ainda não o tornava muito portátil. Além disso, não era recomendável usá-lo no colo, pois o aparelho esquentava muito e podia queimar suas calças.

Outro de seus problemas relatados era o fato de que usava um sistema operacional próprio em vez dos populares CP/M ou DOS.



Figura 9.12: Um GRiD Compass em um ônibus espacial.<sup>18</sup>

## 9.19 80286 – 1982

Em 1982 a Intel lançou um processador que seria sucessor do 8086. Ele foi chamado de 80286, talvez porque sua performance fosse aproximadamente o dobro do 8086. Então nunca houve um 80186? Sim, de fato ele existiu, mas era um microprocessador projetado para uso em sistemas embarcados e microcontroladores, e não era totalmente compatível com o hardware do PC. Portanto, o verdadeiro sucessor do 8086 e 8088 foi o 80286.

Ele era um processador de 16 bits formado a partir de 134 mil transistores. Uma de suas características distintivas era o endereçamento de memória de 24 bits, que permitia a ele gerenciar uma memória de até 16 MB de RAM, enquanto seus antecessores se limitavam a 1 MB. Porém, isso veio um pouco antes do tempo porque naquela época raríssimos sistemas de software usavam mais de 1 MB de memória e, assim, a maioria das máquinas com 80286 eram lançadas com 1 MB ou menos.

Uma das características que permitiam a ele ser mais rápido era o fato de que tinha um circuito especial para calcular endereços de memória nos modos e endereçamento mais complexos. Os antecessores dele tinham apenas uma unidade de lógica e aritmética e ela era usada para fazer quaisquer cálculos, inclusive de endereços. Assim, com um circuito dedicado a esta tarefa, o 286 conseguia ser mais eficiente que os processadores anteriores.

Os primeiros processadores desse tipo lançados tinham velocidade de clock de 6 e 8 MHz, que a Intel conseguiu elevar até 12,5 MHz. A AMD e a Harris, mais tarde, conseguiram lançar clones com velocidade de 20 e 25 MHz. A [Figura 9.13](#) mostra um processador 80286 fabricado pela AMD e instalado em sua placa-mãe.

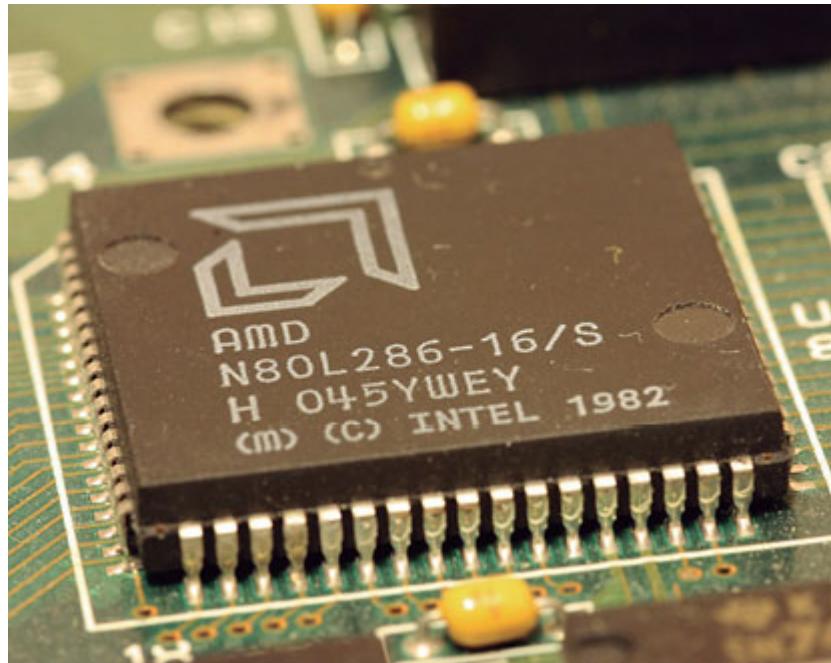


Figura 9.13: 80286.<sup>19</sup>

O 80286 foi projetado para ser totalmente compatível com seus antecessores. Ou seja, programas em linguagem de máquina do 8086 e 8088 rodavam no 80286, o que foi fundamental para seu uso

continuado na série de novos PCs que seriam lançados ao longo das décadas de 1980 e 1990.

O 80286 foi projetado para rodar vários programas em paralelo. O problema era que o DOS, sistema operacional dos PCs, não suportava essa característica. Apenas em 1985 foi lançado pela DRI, em cooperação com a Intel, o Concurrent DOS 286. Esse sistema, porém, revelou algumas falhas no projeto do 80286 que tiveram que ser corrigidas pela Intel.

Esses problemas com o 80286 fizeram com que Bill Gates se referisse a ele como tecnologia morta, pois ficou claro que o novo Windows que a Microsoft estava produzindo não conseguiria rodar versões concorrentes do DOS com o 80286.

## 9.20 Compac Portable – 1982

O Compac Portable ([Figura 9.14](#)) não era um notebook. Seu estilo foi mais inspirado no Xerox NoteTaker e Osborne 1, que se transformavam em uma maleta usando o teclado para tampar o painel com a tela e unidades de disco. Porém, esse computador, o primeiro produto da empresa Compaq Computer Corporation, era quase totalmente compatível com o IBM PC; e podia ser carregado para todo lado.



**Figura 9.14: Compaq Portable.<sup>20</sup>**

Seu sistema operacional era o MS-DOS, visto que a Microsoft manteve o direito de licenciar o sistema para outras empresas além da IBM. Em relação a BIOS, ou seja, o programa interno do PC que inicializava o computador, esse código era propriedade da IBM e ela havia publicado todo o código mantendo o direito de copyright – possuindo, portanto, o direito de processar qualquer empresa que o usasse sem permissão. A Compaq, entretanto, conseguiu reescrever o código todo do zero, sem copiar nenhuma linha e assim escapar de um processo. Essa técnica hoje é conhecida como “*clean room design*”,<sup>21</sup> mas na época foi novidade. A Compaq trabalhou com dois grupos de programadores, um que conhecia o código da IBM e outro que nunca tinha visto esse código. O primeiro grupo descrevia as funções que deviam ser implementadas sem

mencionar nenhum código e o segundo o criava a partir do zero apenas obedecendo aos requisitos indicados pelo primeiro grupo.

O Compaq vinha com um drive de disquete de 5½ com uma característica inovadora: como a máquina seria transportada, quando ela era desligada, os cabeçotes de leitura e escrita do drive ficavam travados de forma que seu movimento involuntário com a máquina desligada não danificasse os disquetes eventualmente mantidos dentro dele. Além disso, era capaz de armazenar o dobro de informações que o disquete do IBM-PC.

Um IBM-PC com configuração semelhante ao Compaq custava na época 3.735 dólares, enquanto que o Compaq custava 2.995. Além disso, com essa configuração, o IBM-PC ficava com apenas um *slot* livre para expansões, enquanto o Compaq manteria três *slots* livres.

O Compaq tinha uma tela de fósforo verde de 9 polegadas que podia exibir tanto texto, quanto gráficos. Era possível também conectá-lo a um monitor externo para obter gráficos coloridos.

Por muitos anos, a Compaq foi a líder entre as empresas que fabricavam clones do IBM-PC.

## 9.21 ZX Spectrum – 1982

O ZX Spectrum foi um computador da Sinclair lançado em 1982. Conforme vemos na [Figura 9.15](#), o layout de seu teclado procurava deixar bem claro que era um computador que gerava gráficos coloridos. Aliás, a figura mostra não apenas o teclado, mas o computador todo. O Spectrum era um computador baseado em Z80 muito leve e podia ser ligado a qualquer aparelho de TV como terminal. Ele era tão leve que às vezes seus donos o enviam pelo correio para a Sinclair para que fosse feita a atualização de 16 para 32 kB de memória.



Figura 9.15: ZX Spectrum.<sup>22</sup>

Cerca de 60 mil unidades da primeira versão com teclas cinza foram produzidas e vendidas. Uma segunda versão com teclas cinza azuladas foi lançada depois.

Em 1987 já havia modelos com 128 kB de memória e possibilidade de uso de disquetes. Contando todas as suas versões produzidas, ele vendeu mais de 5 milhões de unidades, isso sem contar seus clones, que no Brasil foram representados pelo TK 90X. O ZX Spectrum estava para os microcomputadores da Inglaterra mais ou menos como o Commodore 64 estava para os norte-americanos em termos de popularidade.

Seu sistema gráfico era *sui generis*. A resolução da tela gráfica era de 256 x 192 pixels e 16 cores estavam disponíveis. Já a tela de texto tinha 32 colunas por 24 linhas. Porém, para economizar memória, a tela gráfica não guardava a cor de cada pixel. O projeto fez com que a colorização fosse proporcional à tela de texto, ou seja, cada matriz de 8 x 8 pixels na tela tinha que compartilhar a

mesma cor de fundo e cor de frente. Dentro da matriz de 8 x 8 pixels podia-se definir quais pixels teriam a cor de fundo e quais teriam a cor de frente. Assim, o tipo de gráficos gerados pela ZX era bem característico.

O Spectrum foi um computador que teve muitos hobistas e uma grande comunidade de usuários, incluindo várias revistas especializadas na publicação de programas e novidades sobre o computador. Ele foi produzido até 1992, porém o número de usuários ainda é significativo, especialmente hobistas. Algumas máquinas ainda funcionam e, para quem não as têm, existem simuladores, ou seja, programas que rodam em computadores modernos e que imitam o *modus operandi* do Spectrum, permitindo que seus antigos programas sejam executados e novos sejam desenvolvidos.

## 9.22 Computadores de Quinta Geração – 1982

A história da computação nem sempre andou para frente. Alguns reveses aconteceram no caminho. Um deles foi o projeto do computador de quinta geração japonês. O Japão até o início da década de 1980 era conhecido por ser um grande copiador de tecnologia alheia, que não produzia virtualmente muita coisa nova. Na esperança de mudar essa visão e tomar a dianteira no desenvolvimento de computadores, o MITI (Ministério do Comércio Internacional e da Indústria) do Japão iniciou em 1982 um projeto de 10 anos para desenvolver o chamado “computador de quinta geração”, o qual seria uma máquina fortemente baseada em arquitetura e linguagens paralelas, além de usar fortemente os conceitos de inteligência artificial e tendo a linguagem Prolog como sua linguagem de máquina. A nova máquina deveria ser uma arquitetura totalmente nova, não baseada na de von Neumann.

Porém, após gastar o equivalente a mais de 400 milhões de dólares americanos no projeto, o governo japonês anunciou em meados de 1992 que ia desistir de sua proposta.

Quando o Japão anunciou este projeto, no entanto, houve um fenômeno conhecido como “Efeito Sputnik”:<sup>23</sup> com receio de que o programa japonês pudesse dar certo e eles desenvolvessem uma verdadeira inteligência artificial, muitos outros países também criaram seus projetos de desenvolvimento de computadores inteligentes e altamente paralelos. Há quem diga, inclusive, que os cientistas americanos exageraram um pouco a ameaça do computador de quinta geração japonês com o objetivo de receber mais fundos do governo para suas pesquisas.

Porém, o projeto japonês estava muito à frente de seu tempo. Naquela época, apesar da vontade dos pesquisadores, havia ainda muito pouco conhecimento sobre a forma como realmente funciona o raciocínio humano, o qual é muito menos simbólico do que se suspeitava na época. Além disso, o desenvolvimento de arquiteturas paralelas em computadores tem um tipo de aplicação muito restrita. Nem todos os processos de computação se tornam mais rápidos por rodar em uma arquitetura paralela. O que acabou acontecendo também com este projeto foi que os novos processadores RISC e mesmo a família x86, que foi desenvolvida ao longo dessa década, acabaram se tornando bem mais rápidos do que os computadores paralelos japoneses, mesmo em relação a problemas para os quais o paralelismo poderia ser uma vantagem. Em relação a inteligência artificial, no início da década de 1990 computadores de propósito geral conseguiam trabalhar melhor do que os computadores de propósito específico. O mesmo fenômeno já havia acontecido anos antes quando computadores especialmente dedicados a rodar a linguagem LISP foram construídos, as assim chamadas “máquinas LISP”. Em pouco tempo, com o aumento da capacidade dos microprocessadores de propósito geral, eles conseguiam rodar LISP mais rapidamente do que as próprias máquinas LISP.

O projeto deixou como legado toda uma nova geração de cientistas japoneses e um incremento nas pesquisas em inteligência artificial ao redor do mundo. Infelizmente, seu fracasso também desacreditou a área de inteligência artificial por alguns anos. O entusiasmo nessa

área só foi retomado no finalzinho do século XX, quando novos métodos baseados em probabilidade, em vez de processamento simbólico, passaram a ser desenvolvidos, dando origem a toda uma nova área dentro da computação, chamada “inteligência computacional”.

## 9.23 C++ – 1983

C++ é uma linguagem de programação ainda muito usada hoje. Ela foi criada como uma evolução da linguagem C por volta de 1979, quando Bjarne Stroustrup (Dinamarca, 1950) fazia seu doutorado em Cambridge. Nessa época, ele conheceu a Simula 67 e achou suas estruturas de programação muito interessantes, mas Simula era muito lenta para a maioria das aplicações práticas. Assim, ele teve a ideia de adicionar à linguagem C, que era muito eficiente, estruturas semelhantes às de Simula. Dessa forma, a nova linguagem teria a eficiência de C e as facilidades de programação de Simula.

A primeira versão da nova linguagem foi chamada de “C com classes” e seu compilador era o CFront. Ele na verdade poderia ser chamado de pré-compilador porque não gerava linguagem de máquina diretamente, mas código C puro. Assim, o processo consistia em escrever os programas em C com classes, pré-compilar esse código gerando código C e, finalmente, compilar o código C gerando código de máquina.

Stroustrup trabalhava originalmente na construção de um simulador e, naturalmente, usava Simula 67 para tal. Porém, ele percebeu que à medida que o tamanho do sistema crescia, a linguagem e sua implementação apresentavam problemas. Simula não dava conta de grandes sistemas: funcionava bem com pequenos, mas com grandes sistemas o tempo de compilação e a velocidade do código gerado eram inadmissíveis.

Quando se certificou de que continuar com Simula seria um beco sem saída, ele optou por reescrever seu simulador com BCPL. Sua

experiência foi então diametralmente oposta, o código era extremamente eficiente, mas o processo de programação era muito difícil pois a linguagem era muito mais parecida com uma linguagem de máquina do que uma linguagem de alto nível. Segundo ele, BCPL fazia C parecer uma linguagem de muito alto nível. Quando deixou a universidade, jurou nunca mais se empenhar em uma proposta tão mal atendida por ferramentas. Assim, ele iniciou um projeto pessoal para definir a ferramenta adequada para escrever simuladores. Ela deveria ter o suporte de Simula para a organização de programas, como classes, hierarquias, concorrência e tipagem forte, e também ser capaz de gerar programas tão eficientes quanto BCPL, além de ter a característica dessa linguagem de combinar códigos já compilados (*link-edição*) sem ter que recompilar todo o código novamente. Mais do que isso, Stroustrup queria que a nova ferramenta pudesse mesmo *linkar* código gerado por diferentes linguagens de programação como FORTRAN, ALGOL etc.

Em abril de 1979, enquanto Stroustrup trabalhava para o Bell Labs, ele foi envolvido em um projeto para analisar o kernel (núcleo) do Unix de modo a determinar quanto ele poderia ser distribuído sobre uma rede local de computadores. Isso envolvia analisar o tráfego de rede que resultaria dessa distribuição e como seria feita a modularização do kernel para distribuição. Esse era exatamente o tipo de problema que ele tinha jurado nunca mais se envolver sem uma ferramenta apropriada e, portanto, ele se determinou a criá-la.

Em outubro do mesmo ano ele tinha um pré-compilador chamado Cpre, ao qual adicionou a capacidade de compilar estruturas de classes como as de Simula. Em março do ano seguinte ele já podia compilar alguns projetos usando a linguagem C com Classes, a qual o pré-compilador traduzia para C. Por volta dessa época, Stroustrup convenceu-se de que o que ele realmente precisava não era de uma ferramenta, mas de uma nova linguagem de programação.

C com classes seria portanto uma linguagem para organizar programas, ou seja, fazer *programming-in-the-large*, mas a programação de fato, ou *programming-in-the-small*, seria feita em C.

Nesse sentido, os comandos mais básicos de C eram praticamente idênticos aos de C com classes.

A nova linguagem, porém, não deveria perder a eficiência de código que C tinha. Houve muitas sugestões para que algumas inseguranças de C fossem eliminadas em C com classes, mas Stroustrup argumentava que colocar mecanismos de segurança, que fariam determinadas checagens enquanto o código estava rodando, reduziria a performance da linguagem. Ele até considerava essas verificações aceitáveis enquanto o código estava sendo testado e depurado, mas não quando o código final estivesse pronto.

Em 1983, o nome da linguagem foi mudado para “C++” por basicamente dois motivos: o primeiro é que “++” é uma das operações da nova linguagem que incrementa uma variável inteira e o segundo é que “+” (plus) já era usado na área de tecnologia para indicar algo superior; assim “++” ou “plus-plus” deveria ser algo muito superior.

Quando perguntado “Por que C?”, ou seja, porque não usar outra linguagem como base, visto que C não era exatamente a linguagem mais limpa e organizada que existia, ele apontava as seguintes vantagens: C era flexível, eficiente, disponível e portável. C podia ser usado em praticamente qualquer área de aplicação, sua semântica era de baixo nível, isto é, a linguagem era muito próxima da própria linguagem de máquina. Qualquer computador, grande ou pequeno, poderia ter um compilador C aceitável implementado para ele. Um programa escrito para um computador não era necessariamente portável automaticamente para outro, nem necessariamente seria fácil fazer isso, mas usualmente era possível.

Na tabela TIOBE de fevereiro de 2016, C++ aparecia na terceira posição em popularidade, atrás apenas de Java e C.

## 9.24 Turbo Pascal – 1983

Em 1983 havia várias implementações do compilador Pascal disponíveis no mercado. Talvez a mais usada fosse a UCSD Pascal, da Universidade da Califórnia em San Diego. Até a Microsoft tinha seu próprio Pascal, que não era tão usado quanto seu BASIC. Porém, nesse ano uma empresa recém-criada, a Borland, produziu algo de novo nessa linha. Primeiro, o compilador desenvolvido por eles era absurdamente rápido se comparado com os outros que existiam. Compilações que levavam minutos passaram a ser feitas em segundos. Além disso, eles integraram várias ferramentas criando algo que pode ser considerado um dos primeiros IDEs (Integrated Development Environment). O sistema vinha com um editor de texto para escrever os programas, o qual era muito semelhante ao WordStar, que na época definia os padrões de interface e comandos em editores de texto. Nesse mesmo editor, com um toque de teclas era possível mandar compilar o programa. Se houvesse algum erro no código, o editor de texto abria o programa exatamente na posição na qual o erro tinha sido detectado e já o indicava. Por outro lado, se o programa compilasse sem erros ele seria imediatamente executado e o usuário poderia ver seus resultados.

O sistema foi chamado de Turbo Pascal, em referência à velocidade com que os programas eram compilados e executados. Até então, a maioria dos compiladores eram programas separados. Você tinha que trabalhar com o editor de texto, salvar o programa, sair do editor de texto, executar o compilador, indicar qual o arquivo a ser compilado, em caso de erro, anotar em um papel a linha onde este ocorreu, abrir novamente o editor de texto, carregar o texto do programa, procurar a dita linha, corrigir e repetir o processo para ver se deu certo. Se o programa fosse dividido em vários arquivos, ainda haveria o processo de link-edição. O ambiente de programação da Borland simplificou bastante esse panorama.

Além disso, o ambiente era vendido barato. A maioria dos compiladores na época custava centenas de dólares, mas o Turbo Pascal completo era vendido por apenas 49,95 dólares. Mais ainda,

os disquetes não eram protegidos contra cópia como a maioria dos sistemas da época. Sua licença era semelhante à de um livro: o dono do disquete podia usar à vontade onde e quando quisesse. Ele apenas tinha que cuidar para que não fosse usado em mais de um lugar ao mesmo tempo, pois isso seria ilegal.

Para que um programador trabalhando com Turbo Pascal pudesse comercializar os programas compilados com este sistema a Borland cobrava mais 100 dólares de taxa. Mesmo assim, o sistema completo com autorização para venda de produtos por um preço de 149,95 era considerado muito barato.

Comenta-se que Bill Gates teria ficado muito irritado com o sucesso do Turbo Pascal em detrimento do MS-Pascal. Como uma empresa recém-criada poderia bater a Microsoft, na época já uma companhia grande e respeitável?

Apenas em 1995 a Borland encerrou o projeto do Turbo Pascal ou Borland Pascal, como era então conhecido, para o lançamento de sua sucessora, a linguagem Object Pascal, integrada ao ambiente de desenvolvimento Delphi. Tanto o Turbo Pascal quanto o Delphi tiveram grande importância como linguagens de programação, não apenas com fins educacionais, mas também no desenvolvimento de sistemas, especialmente no Brasil onde a linguagem foi dominante nos anos 2000. Delphi ainda aparecia na 10<sup>a</sup> posição da tabela Tiobe de fevereiro de 2016.

Atualmente, as versões do Turbo Pascal até a 5.5 são distribuídas gratuitamente e consideradas “abandonware”, pois a empresa já não mais mantém o produto e embora ainda seja possivelmente protegido por direitos autorais, ela nada faz para garantir esse direito.

## 9.25 Apple IIe – 1983

Após o fracasso do Apple III, a empresa resolveu lançar uma versão melhorada do Apple II para tentar ganhar uma fatia maior do mercado de pequenos negócios. Essa nova versão foi chamada de

Apple IIe, no qual “e” indica “enhanced”, ou “melhorado”. Ele também chegou a ser chamado de “Diana” e “Super II”, mas o nome oficial acabou sendo Apple IIe ou, por vezes, Apple ][e ou Apple //e. Na [Figura 9.16](#) pode-se ver um Apple IIe sem a tampa superior, o que permite visualizar sua placa-mãe e os vários *slots* na parte superior, nos quais placas de expansão e periféricos podiam ser conectados.



Figura 9.16: Apple IIe sem a tampa superior.<sup>24</sup>

O Apple IIe não chegou a ser um marco tecnológico, mas suas novas características foram importantes para manter a Apple no mercado e competir especialmente com a IBM, que havia lançado o PC dois anos antes, especialmente considerando-se que o Apple II original era tecnologia dos anos 1970 e, portanto, já defasada.

Entre outras coisas, o novo Apple IIe já vinha com 64 kB de memória de fábrica, mas usava apenas  $\frac{1}{4}$  da quantidade de circuitos integrados que o seu antecessor.

Além disso, ele podia representar letras minúsculas, coisa que o II não fazia, além de texto em 80 colunas em vez de meras 40. Outra melhoria, que pode parecer enganosamente pequena, foi a existência de quatro teclas de navegação (setas), visto que a versão anterior só tinha direita e esquerda. Lembram como isso complicava a navegação do VisiCalc?

O Apple IIe foi produzido até o final de 1993, mas há sobreviventes usados hoje em dia em alguns lugares. Pequenas revisões foram produzidas neste meio-tempo, sendo a primeira delas o (redundante) “Enhanced Apple IIe”, lançado em 1985 e o “Apple IIe Platinum” de 1987.

## 9.26 Apple Lisa – 1983

Se o Apple IIe não foi um marco tecnológico, pelo menos foi um sucesso de vendas. O inverso aconteceu com outro projeto da Apple lançado no mesmo ano, o Apple Lisa ([Figura 9.17](#)), que foi um marco tecnológico, mas fracasso de vendas, visto que apenas 100 mil unidades foram comercializadas.

Quando Steve Jobs visitou o Xerox PARC e conheceu o Xerox Alto em 1979, ele ficou tão impressionado com a interface gráfica com janelas e com o uso do mouse que resolveu investir nessa direção também. No entanto, apenas em 1983 a Apple conseguiu lançar um computador com essas qualidades e isso após investir 50 milhões de dólares no projeto.

O Apple Lisa era um modelo altamente avançado. Ele tinha características que somente vários anos depois apareceram em outras máquinas, como o Macintosh da própria Apple. Toda essa sofisticação tinha um custo, porém: 9.995 dólares. O preço, aliado ao fato de que a Apple deixara escapar que tinha em produção um novo modelo mais barato e de capacidade equivalente, pode ter feito as vendas serem mais tímidas do que a empresa esperava (Efeito Osborne).

O Lisa usava um processador poderoso para a época, o Motorola 68000 de 5 MHz e sua memória atingia, então impressionantes, 1 MB. O Lisa original tinha dois drives de disquete de 5¼ embutidos de dupla capacidade, chamados de Twiggy, e que permitiam armazenar 871 kB cada, mas necessitavam de disquetes especiais. Infelizmente, esses drives ganharam uma reputação de não serem confiáveis e acabaram sendo substituídos a partir de 1984 por um modelo Sony de 400 kB e 3½ polegadas. Um disco rígido externo de até 10 MB, que tinha sido originalmente desenvolvido para o Apple III, também estava disponível.



Figura 9.17: Apple Lisa.<sup>25</sup>

Ainda em 1984, uma revisão do Lisa, denominada Lisa II foi lançada. O computador passou então a custar entre 3.500 e 5.500 dólares.

Em relação ao nome do computador, Jobs deixou muito claro em sua biografia que se tratava de uma homenagem à sua filha, Lisa Nicole Brennan, nascida em 1978, mas existem várias interpretações apócrifas de LISA como um acrônimo. A mais comum é “Local Integrated System Architecture”.<sup>26</sup> Outras mais espirituosas são “Lisa: Invented Stupid Acronym”<sup>27</sup> ou “Let’s Invent Some Acronym”.<sup>28</sup>

Em 1980, Jobs teve que deixar o projeto do Lisa em detrimento do Macintosh, que, ao contrário do que muitos pensam, não foi seu sucessor, embora tenha sido lançado depois; tratavam-se de projetos paralelos, mas que usaram algumas tecnologias em comum.

O lançamento do Macintosh em 1984 com características semelhantes, embora inferiores, ao Lisa, mas a um custo muito mais baixo e maior investimento em marketing, acabou levando ao fim da produção do Lisa em abril de 1985. Logo depois, a Apple ofereceu aos donos dessa máquina a oportunidade de, por um pequeno investimento, trocar seu computador por um Macintosh.

## 9.27 IBM PC-XT – 1983

Em 1983 a IBM já havia se dado conta de que a configuração do seu IBM PC não era suficiente para atender a muitas das aplicações comerciais. A principal forma de entrada e saída de dados era através de fita K7, com drives de disquetes opcionais e sem possibilidade de disco rígido interno. Além disso, 16 kB de memória expansível para 64 kB era muito pouco.

Foi lançado assim o modelo IBM 5160, denominado XT (eXtended Technology), com um disco rígido interno de 10 MB, uma unidade de disquete de 5½ com capacidade para discos de 360 kB e uma memória principal inicial de 128 kB expansível para 640 kB ([Figura 9.18](#)). Fora isso, era praticamente a mesma máquina. Seu processador ainda era o Intel 8088 a 4,77 MHz, com um coprocessador aritmético opcional, o 8087.



**Figura 9.18: IBM PC-XT.<sup>29</sup>**

Até ser substituído por outro modelo em 1987, o XT vinha de fábrica com o PC DOS e o IBM BASIC, ambos fabricados pela Microsoft. Um acordo entre as duas empresas obrigava a IBM a colocar estes produtos em todos os seus XTs.

Em 1986 foi ainda lançado o XT-286, ou IBM 5162, um modelo com o processador 80286 em vez do 8088. Ele não deve ser confundido com o IBM PC AT (Advanced Technology), também conhecido como IBM 5170, baseado no mesmo processador. O AT, lançado em 1984, é considerado uma geração mais recente, por ser baseado em várias tecnologias não disponíveis para o PC XT.

## **9.28 IBM PCjr – 1983**

Não foi só a Apple que colecionou fracassos em meio a sucessos nos anos 1980. A IBM também teve o seu grande fiasco e ele foi o IBM PCjr, ou IBM 4860, também conhecido como Peanut, ou Amendoim ([Figura 9.19](#)).



Figura 9.19: IBM PCjr.<sup>30</sup>

Ninguém discutia que a IBM era a gigante dos computadores. Para muitas pessoas, “IBM” e “computador” eram sinônimos. O lançamento do PC no segmento de mercado dos microcomputadores, competindo com as recém fundadas Apple, Commodore e Tandy, foi um sucesso, e logo a IBM se tornou líder de vendas também no segmento de microcomputadores, mas com uma fatia bem menor do que ela tinha no mercado de mainframes. No mercado de micros a IBM liderava com 26%, mas no mercado de mainframes ela tinha 70%.

Não poucas pessoas imaginavam que a IBM queria mais. Assim, por volta de 1982 surgiram rumores de que a empresa ia lançar um novo computador pessoal que seria compatível com o IBM PC, mas custaria menos do que um Apple IIe. Pessoas iam às lojas tentando comprar essa máquina que não existia e não se conformavam com a informação de que ela não estava disponível. Por outro lado, muitos diziam que quando tal computador fosse lançado, ele iria acabar com as demais empresas, como a Apple, Tandy e Commodore. A IBM sistematicamente negava tais rumores. Esse movimento ficou conhecido como “*peanut panic*” ou “pânico do amendoim”.

Em 1983, contudo, a IBM fez o anúncio de um novo computador pessoal. Houve muita propaganda, um verdadeiro megaevento. A revista *Time* chamou esse momento de o Dia D do microcomputador, aludindo à histórica data do desembarque dos Aliados na Normandia, que provocou uma reviravolta na Segunda Guerra.

A expectativa era enorme, tanto por parte dos usuários que desejavam a nova máquina, quanto por parte dos especialistas que previam que o PCjr passaria a ser o novo padrão da indústria, fazendo com os computadores caseiros aquilo que o PC havia feito com os computadores de escritório. Desenvolvedores de software começaram a adaptar e criar produtos para a nova máquina antes mesmo de ela estar disponível porque tinham certeza de que viria para ficar.

Apesar de todo o entusiasmo, quando a máquina realmente chegou às lojas e começou a ser testada e demonstrada, as vendas não foram nada do que se esperava. O *Wall Street Journal* chegou a publicar que a máquina tinha três grandes vantagens: as letras “I”, “B” e “M”. Fora isso, era decepcionante para a maioria dos compradores.

A primeira grande deceção foi o teclado estilo “chiclete”. Todos sabiam pela experiência com computadores mais antigos que esse

modelo de teclado era pouco resistente e não se prestava para trabalhos mais intensos. A *PC Magazine* chegou a publicar em uma avaliação que digitar no PCjr era como massagear um bolo de frutas. Comenta-se que quando viram o teclado do PCjr a Atari e a Coleco, que fabricavam máquinas para o mesmo segmento de mercado, imediatamente elevaram seus preços.

Empresas de software como a Sierra, que investiram pesado desenvolvendo produtos para o PCjr usando apenas protótipos, ficaram horrorizadas com o teclado, considerado de muito má qualidade. Mesmo os vendedores da IBM aconselhavam eventuais compradores a investir em um teclado novo.

Em relação ao custo, que variava de 800 a 1.600 dólares, ficava a sensação de que o PCjr era caro para um computador pessoal e pouco potente para um computador caro. A Texas Instruments, por exemplo, pouco tempo antes vendia um computador pessoal a 99 dólares. Embora ela o fizesse amargando prejuízos em função de uma guerra de preços com outras empresas, e tenha mesmo deixado o mercado de microcomputadores logo depois, isso era uma amostra de que o valor do PCjr estava realmente longe do orçamento da maioria das famílias. Jeanne A. Dietsch, presidente de uma empresa de análise de mercado, comentou que a IBM escolheu a pior faixa de preço possível para trabalhar: demais para uma residência e muito pouco para uma empresa. Já Peter Norton disse que a máquina foi perfeitamente projetada para um mercado que simplesmente não existia.

O PCjr custava o dobro de um Commodore 64 e era inferior a ele em gráficos. As máquinas da Atari também eram superiores e mais baratas. Para piorar, logo depois do lançamento do PCjr a Apple introduziu o Macintosh, uma máquina muito mais interessante e poderosa do que um PC e custando apenas um pouco mais do que o PCjr. A Apple reagiu duplamente ao lançar também o Apple IIc, uma máquina com poder semelhante ao PCjr, mas com um teclado muito superior, além de ser muito mais barata.

Outro erro que a IBM cometeu com esse lançamento foi apostar que a grande maioria dos compradores seriam novatos, leigos – mas ocorreu justamente o oposto, cerca de 75% dos potenciais compradores já tinham experiência com computadores e queriam máquinas que rodassem programas como Lotus 1-2-3, WordStar e outros produtos de automatização de escritório. Infelizmente para eles e para a IBM, apesar de uma compatibilidade esperada, ela não ocorreu na prática, porque cerca de 60% das aplicações do PC falhavam ao ser rodadas no PCjr e isso foi a gota d'água para os clientes.

A Tandy lançou em 1984 o Tandy 1000, e sua propaganda dizia que ele era o que o PCjr devia ter sido: dentre outras coisas, 100% compatível com o IBM PC.

Neste mesmo ano, a IBM ofereceu-se para trocar os teclados chiclete por teclados normais e muitos viram nisso um ato de extrema generosidade enquanto outros enxergaram a confissão de um erro de projeto. Porém, a máquina tinha outro problema: pouco espaço para expansões; você teria que escolher entre mais memória, drive de disco, modem, impressora etc., mas não poderia ter todas as funções pois não havia lugar para ligar tudo (isso era mais ou menos como ter um computador moderno com uma única porta USB).

A IBM fez um redesign da máquina e investiu pesadamente em propaganda. Além disso, ela baixou o preço, deixando um PCjr equivalente a um Apple IIc. Essa campanha foi chamada pelos observadores de “Salvem o Jr.”. Para espanto de muitos, ela deu certo e o PCjr foi, por alguns meses, líder de vendas no mercado. Porém, quando os descontos acabaram as vendas também baixaram. Assim, no início de 1985, a IBM chegou à conclusão de que o PCjr não era lucrativo e resolveu acabar com essa linha de computadores, para desespero da indústria de software que tinha investido muito nela.

## 9.29 Word – 1983

A primeira ferramenta a compor o atual Microsoft Office foi o Word. Ele surgiu timidamente em 1983 em uma versão para DOS e Xenix. O projeto foi chefiado por Charles Simonyi (Hungria, 1948), contratado pela Microsoft em 1981. Simonyi foi o responsável pelo desenvolvimento do primeiro editor de texto WYSIWYG, o Bravo, da Xerox Parc, que foi desenvolvido para o Xerox Alto em 1974.

Inicialmente o nome da ferramenta era Multi-Tool Word, mas logo foi simplificado para “Word”. Ele não se tornou popular no início, primeiro porque era bem diferente do campeão de vendas, o WordStar, e segundo porque preconizava o uso do mouse, um dispositivo que na época ainda não era muito comum entre os usuários.

Quando do seu lançamento em 1983, um disquete de demonstração do Word foi distribuído para os assinantes da revista *PC World*. Foi a primeira vez na história que uma revista era distribuída com um disquete contendo software.

Em 1988, centenas de usuários do Apple Macintosh se reuniram em um auditório para ver Bill Gates, o bilionário dono da Microsoft, apresentar o novo editor de texto, Word 3.0. Foi um tremendo golpe de marketing, pois Gates anunciou um produto que não estava nem perto de ser concluído e passou para o mercado a seguinte mensagem: “Parem de comprar outros editores de texto, porque a Microsoft vai lançar o Word 3.0 em breve, e ele será um sucesso.” Não só os compradores seguraram o fôlego, como os desenvolvedores, desanimados, cancelaram ou retardaram o desenvolvimento de novas versões de seus produtos até ver o que a Microsoft tinha preparado.

Porém, o Word 3.0 só chegou ao mercado quase seis meses depois. Durante a apresentação, porém, Gates demonstrou ser capaz de caminhar sobre um campo minado com pés de plumas. Ele foi avisado de que havia na versão de demonstração do Word seis grandes bugs que poderiam travar o sistema caso esbarrasse em qualquer um deles. Gates foi capaz de fazer a demonstração de

forma quase perfeita. Apenas em um momento ele provocou uma falha de sistema, mas nada que usuários de computadores, acostumados com esse tipo de problema, não pudessem relevar. O fato que fez com que Gates fosse mais tarde aclamado como um grande apresentador foi a descoberta de que o Word não tinha apenas seis falhas graves, mas 600. Gates esbarrara em apenas uma delas.

Em 1988, a Microsoft também lançou uma versão do Word para o computador Atari ST. Essa versão foi originalmente traduzida a partir daquela desenvolvida para o Macintosh. A versão para o Atari, porém, foi lançada sob o nome Microsoft Write. Posteriormente, essa versão foi embutida nos sistemas operacionais Windows.

A primeira variante para Windows, chamada WinWord, foi lançada em 1989, mas foi apenas com o lançamento do Windows 3.0 em 1991 que o Word se tornou um sucesso comercial. O Write continuou sendo oferecido com o sistema operacional, ao passo em que o Word era vendido separadamente. O Write continuou sendo um editor bastante simples e o Word foi se sofisticando com o passar do tempo. Os concorrentes do Word na época, como o WordStar e o WordPerfect, falharam ao não desenvolver uma versão para o Windows. Na medida em que o Windows se tornou amplamente usado, o WinWord foi ganhando terreno frente a estes outros aplicativos.

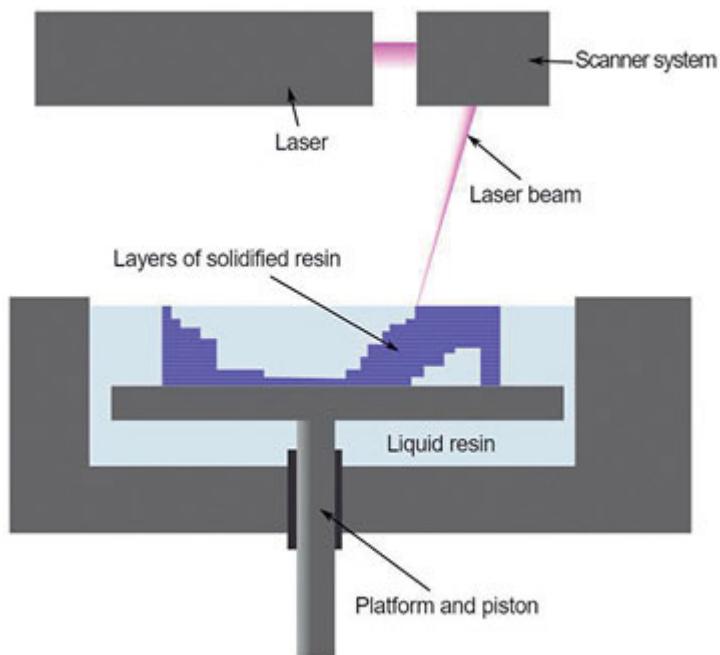
### **9.30 Impressora 3D – 1984**

A história da impressão em 3D remonta à invenção das impressoras jato de tinta incialmente nos anos 1950 e depois mais largamente desenvolvidas nos anos 1970. Se era possível controlar um cabeçote de impressão para soltar gotículas de tinta de diferentes tonalidades sobre uma folha de papel, por que não seria possível programar uma máquina dessas para soltar partículas de algum tipo de resina sobre uma superfície em camadas sucessivas, formando, assim, objetos tridimensionais?

A primeira patente sobre impressão 3D foi solicitada por Hideo Kodama do Instituto de Pesquisas Municipal de Nagoya, no Japão. Entretanto, ele, embora fosse advogado de patentes por profissão, não fez o registro definitivo da própria um ano depois da solicitação inicial, e com isso, perdeu o direito de ser oficialmente reconhecido como o inventor.

Nos anos 1980 a impressão 3D era conhecida principalmente como “prototipação rápida”, porque seu objetivo inicialmente era produzir peças mecânicas de maneira rápida para que fossem inspecionadas e aperfeiçoadas antes de entrar no processo usual de usinagem. Hoje, outro termo bastante usado na indústria para impressão 3D é “manufatura aditiva”, considerado mais genérico.

Três anos depois de Kodama, em 1984, Charles Hull (Estados Unidos, 1939) inventou, e em 1987 patenteou, a estereolitografia, que é uma técnica para construir figuras em 3D a partir de modelos computacionais. A técnica é baseada no uso de um fotopolímero, que é normalmente líquido, mas que se for aquecido por, digamos, um raio laser ultravioleta, se solidifica na forma de plástico. A impressão é feita camada por camada. O laser age na superfície do líquido formando a primeira camada da impressão. Depois, um pequeno elevador submerge essa camada no líquido apenas o suficiente para que a segunda camada a cubra, usualmente 0,05 mm até 0,015 mm. Então o processo de endurecimento do polímero é novamente feito pelo laser. O processo se repete com o objeto afundando cada vez mais no líquido até que as camadas mais do topo são finalmente adicionadas ([Figura 9.20](#)). Existem outras técnicas de impressão 3D: algumas são baseadas no uso de pó no lugar de polímero líquido e outras usam uma resina que endurece em contato com o ar. A caneta 3D é um exemplo dessa técnica.



**Figura 9.20: Processo de estereolitografia.<sup>31</sup>**

O interessante é que três semanas antes de Hull solicitar sua patente, os inventores franceses Alain Le Méhauté (França, 1947), Olivier de Witte e Jean Claude André haviam patenteado o mesmo processo, mas como não viram nenhuma aplicação comercial prática para a ideia, eles a abandonaram.

Em 1999, a primeira experiência de impressão 3D associada à reposição de órgãos humanos foi realizada quando cientistas do instituto Wake Forest de medicina regenerativa imprimiram em três dimensões a estrutura básica de uma bexiga humana e fizeram crescer sobre ela, usando células do próprio paciente, uma nova bexiga funcional. A nova bexiga foi implantada no paciente sem risco algum de rejeição.

A década de 2000 foi muito profícua para a medicina regenerativa, graças à impressão 3D. Entre outras conquistas, foram impressos em três dimensões rins funcionais e pernas mecânicas com

componentes complexos. Além disso, nessa época também foram impressos vasos sanguíneos a partir de células humanas.

Em relação à reprodução de máquinas, foi proposto em 2005 e realizado por Adrian Bowyer (Reino Unido, 1952) em 2008 o projeto de uma impressora 3D, a Darwin, que consegue imprimir uma cópia exata de si própria.

Atualmente, com o preço caindo e a precisão aumentando, esse tipo de impressora já é algo que se pode ter em uma residência. Há histórias conhecidas de pessoas que imprimiram por poucos dólares próteses humanas que, se compradas de fabricantes tradicionais, custariam dezenas de milhares de dólares. Por outro lado, a impressão 3D também permite a construção de armas de fogo caseiras plenamente funcionais.

A NASA é uma das maiores incentivadoras dessa tecnologia, com projetos que visam a impressão de comida e em gravidade zero.

## 9.31 GNU – 1984

Quando Richard M. Stallmann (Estados Unidos, 1953) começou a trabalhar no Laboratório de Inteligência Artificial do MIT em 1971, ele se tornou parte de uma comunidade que se autodenominava “hackers”. Essa palavra não tinha a conotação negativa que às vezes atribui-se hoje, ou seja, não eram necessariamente pessoas que quebravam a segurança de sistemas (ou seja, “crackers”), mas indivíduos muito criativos e apaixonados por programação.

Uma das características dessa comunidade era acreditar que software era algo para ser compartilhado, e não protegido. Eles frequentemente trocavam programas entre si, modificando-os e melhorando-os.

No laboratório, usavam um PDP-11 com um sistema operacional chamado ITS (The Incompatible Timesharing System<sup>32</sup>), criado pelos hackers em linguagem de máquina. Embora o termo “software livre” ainda não existisse, eles compartilhavam o sistema livremente com outras universidades ou quem quer que quisesse usá-lo.

A comunidade, porém, entrou em declínio a partir do início dos anos 1980, primeiro porque a DEC descontinuou o PDP-11 e os programas escritos em sua linguagem de máquina não eram facilmente portáveis para outros computadores mais modernos. Em segundo lugar, a universidade passou a usar sistemas operacionais proprietários em seus computadores. Stallman reclama de ter que assinar um termo de confidencialidade até para ter acesso ao código executável (que dirá os códigos-fonte). Ele considerava isso inaceitável: o primeiro passo para usar um computador era jurar que você não iria ajudar o seu próximo; uma comunidade de cooperação era proibida: “Se você ajudar seu próximo, você é um pirata e se você quiser modificações no sistema, deve nos implorar por elas!” Essas são as regras dos vendedores de software, dizia.

Stallman resolveu que não se entregaria à indústria do software proprietário, mas que iniciaria uma nova comunidade, produzindo programas que poderiam ser compartilhados por todos. A primeira coisa a fazer, pensou ele, era um sistema operacional, porque a grande maioria dos aplicativos necessita rodar sobre algum deles. Portanto, este seria o primeiro produto. Como ele era um desenvolvedor de sistemas operacionais, isso facilitou a escolha.

Stallmann resolveu que usaria como inspiração o sistema Unix porque ele era altamente portável e tinha já uma grande comunidade de usuários que eventualmente estariam dispostos a se juntar a ele nessa nova comunidade. Com alguma inspiração hacker, definiu como nome desse novo sistema operacional a sigla recursiva GNU, que significa: “GNU’s not Unix.”<sup>33</sup> A [Figura 9.21](#) apresenta o logotipo atual do projeto GNU.



Figura 9.21: Logotipo do projeto GNU.<sup>34</sup>

Entendia-se que o sistema deveria ser livre, e isso não significa necessariamente gratuito. Há muita confusão ainda hoje sobre software livre e software grátis. Software pode ser grátis, mas não ser livre; e também pode ser livre, mas comercializável.

Para ser livre, um sistema deve obedecer às seguintes regras:

- Você tem a liberdade de rodar o programa onde quiser e para qualquer finalidade, inclusive comercial.
- Você tem a liberdade de modificar o programa para atender às suas necessidades. Para que essa liberdade seja efetiva na prática, você deve ter acesso ao código fonte.
- Você tem a liberdade de redistribuir cópias, seja de graça, seja cobrando por elas.
- Você tem a liberdade de distribuir versões modificadas do programa de forma que outros possam se beneficiar delas.

Portanto “livre” se refere a essas liberdades e nada tem a ver com preço. Se fosse obrigatório redistribuir de graça, não seria livre.

Assim, em janeiro de 1984, Stallmann largou seu emprego no MIT e começou a trabalhar no projeto GNU. Ele temia que o MIT pudesse exigir o copyright do software desenvolvido por ele, visto que era empregado lá. Entretanto, o diretor do Laboratório de IA do MIT convidou Stallmann a continuar usando o espaço da universidade, mesmo que não mais como funcionário.

Para iniciar, ele se interessou em usar o Free University Compiler Kit,<sup>35</sup> e perguntou ao seu criador se poderia ser usado em um projeto como o GNU. O autor esclareceu que o que era “livre” era a universidade, mas não o software. Assim, Stallmann resolveu escrever um compilador multiplataforma antes de mais nada. Ele teve que escrever o compilador, hoje conhecido como GCC, a partir do zero, e talvez por isso só tenha ficado pronto anos mais tarde. O primeiro produto do projeto foi o GNU Emacs, um editor de texto.

No início de 1985, o Emacs já era usável e surgiu o problema de como distribui-lo. Stallmann resolveu usar o próprio computador que ele tinha no MIT e disponibilizou o sistema no endereço prep.ai.mit.edu através do serviço de FTP (File Transfer Protocol<sup>36</sup>).

A maioria das pessoas não tinha acesso à rede de computadores, porém. Então após ponderar sobre várias maneiras de disponibilizar o produto a essas pessoas, e porque ele precisava ganhar dinheiro para sobreviver, Stallman se dispôs a enviar o Emacs gravado em fita magnética para qualquer um que lhe pagasse uma taxa de 150 dólares. A partir de então a pessoa adquiria todas as liberdades citadas anteriormente e, assim, iniciavam os negócios com software livre.

Havia ainda o problema de alguma empresa ou individuo registrar os direitos sobre o GNU, tornando-o software proprietário. Para evitar isso, ele criou o conceito de *copyleft*, o qual usa a lei de copyright para o efeito exatamente oposto, ou seja, para garantir que o produto seja e permaneça livre.

Com o crescimento da comunidade foi criada em 1985 a Free Software Foundation (FSF), uma empresa filantrópica, isenta de impostos, para desenvolvimento de software livre. Ela assumiu a comercialização de fitas com o Emacs e logo começou a adicionar outros produtos e manuais livres. A empresa existe até hoje e continua vendendo manuais, mas sua principal fonte de renda são doações dos seus membros.

## 9.32 Apple Macintosh – 1984

O ano de 1984 ficou marcado por ser quando se passava a história do famoso livro *1984*, escrito por George Orwell em 1949. A narrativa acontece em um mundo no qual todas as pessoas são controladas pelo “Big Brother”,<sup>37</sup> que observa a todos o tempo todo e se comunica a partir de aparelhos semelhantes a televisores que estão por toda a parte. A Apple fez uma propaganda que ficou famosa neste ano, a qual mostrava uma mulher arremessando um martelo na tela central do Grande Irmão, liberando as pessoas de seu controle e dizendo no final: “No dia 24 de janeiro a Apple vai lançar o Macintosh e você vai ver porque 1984 não será como *1984*”. A propaganda faz referência indireta à IBM, claro, e seu controle sobre o mercado de computadores. O Macintosh (Mac) é apresentado como uma alternativa. O vídeo do comercial pode ser assistido em [https://www.youtube.com/watch?v=TvHrJ\\_S5jAQ](https://www.youtube.com/watch?v=TvHrJ_S5jAQ).

O projeto do Macintosh ([Figura 9.22](#)) iniciou em 1979 com Jef Raskin (Estados Unidos, 1943-2005), um empregado da Apple que pensou em desenvolver um computador fácil de usar e voltado para o grande público. Ele queria chamar o computador de McIntosh, porque era sua maçã favorita. Porém, a Apple não conseguiu registrar esse nome porque já havia um McIntosh Laboratory, Inc., um fabricante de equipamentos de áudio. Assim, resolveram mudar a grafia para Macintosh. Ainda assim a Apple teve que pagar direitos à McIntosh para usar o nome.

No dia do seu lançamento, Steve Jobs não se parecia nem um pouco com o apresentador elegante que fora nos anos mais recentes. Comenta-se que, com seu traje cinza muito grande para ele, a gravata verde e o cabelo comprido, ele lembrava mais um hippie que tinha se vestido para o casamento de um parente do que com o CEO de uma empresa importante. Jobs estava nervoso: ele comentou com um amigo que estava em pânico e que esse seria o dia mais importante da vida dele.



Figura 9.22: Macintosh.<sup>38</sup>

De fato, por essa época as empresas que fabricavam microcomputadores estavam fraquejando e algumas chegavam a quebrar. A Apple e a IBM vendiam em torno de 1 bilhão de dólares anualmente em microcomputadores e se alternavam na liderança do mercado. O mesmo mercado que recebeu a IBM de braços abertos, porém, agora temia que ela tomasse o controle total e passasse a ditar os rumos da área de computação pessoal. A única esperança para competir com a IBM nesse mercado era a Apple.

No ano de 1983 tanto a IBM quanto a Apple tiveram seus fiascos, a primeira com o lançamento desastrado do PCjr e a Apple com o LISA que, apesar de ser uma máquina que incorporava grandes

ideias, tinha um preço muito alto, pouco desempenho (a interface gráfica consumia quase todo o tempo de processador) e faltavam aplicativos para o mercado empresarial.

O projeto do Macintosh, porém, foi criado sobre outras bases. A máquina teria também uma interface gráfica semelhante à do LISA, mas era um computador que desde o início foi pensado para conquistar as massas. Seu preço deveria ser de 500 dólares, mas os gastos enormes na campanha de marketing fizeram a Apple elevar esse preço para cerca de 2.500 dólares. Ainda assim, era  $\frac{1}{4}$  do valor do Apple LISA. O investimento em propaganda foi pesado, a Apple chegou a comprar 39 páginas de anúncio na revista *Newsweek*.

As vendas iniciaram bem, mas depois que os nerds e hobistas compraram seus computadores elas despencaram. Os clientes empresariais pensavam mais ou menos assim: “O Macintosh é bem avançado e muito fácil de usar, mas o PC tem os aplicativos que nós precisamos, então...”. A Apple fez muitas campanhas de evangelização para tentar atrair desenvolvedores para construir os aplicativos que a empresa precisava para emplacar o Mac. Porém, a memória de apenas 128 kB era muito pequena para a maioria dos aplicativos. Apenas um ano depois foi lançado o Fat Mac, com 512 kB de memória, que resolveu esse problema.

Em 1987 a Apple vendeu 1 milhão de Macs, voltando a ameaçar a IBM. Mais do que a metade dos 2 mil dólares pagos por um Mac eram lucro para a Apple, o que levou seus executivos a se convencerem de que o povo está realmente disposto a pagar mais por tecnologia melhor. Foi com essa política, porém, que a Apple perdeu a chance de tornar o Mac um padrão da indústria. Se eles tivessem diminuído os lucros e incentivado a produção de clones, o Mac não teria sucumbido ao lançamento do Windows 3.0 em 1990, que fez o PC (não necessariamente da IBM) voltar ao topo e se tornar o padrão de mercado.

O mais importante de tudo foi que a Apple lançou com o Mac as bases para uma nova revolução na forma como as pessoas iriam interagir com os computadores. Longe do padrão IBM, o Mac tinha mais a ver com o Sketchpad, o Dynabook e o Smalltalk: interfaces baseadas em janelas, ícones e menus e controle sobre a máquina exercido com um mouse. O próprio Bill Gates teria afirmado na época ao então *BusinessWeek* (hoje *Bloomberg Businessweek*) que a próxima geração de software interessante seria escrita para o Mac e não para o PC.

### 9.33 Mac OS – 1984

Provavelmente a maior estrela do Macintosh foi seu sistema operacional gráfico. Hoje nós o conhecemos como Mac OS, mas na época do lançamento ele se chamava simplesmente System 1.0. Ele só foi renomeado quando a Apple começou a licenciar clones e precisava que o software tivesse um nome que o identificasse mais claramente.

O sistema operacional era totalmente gráfico. Ele sequer tinha a linha de comando característica do DOS que foi mantida em todas as versões do Windows na forma do aplicativo Command Prompt. O sistema era parcialmente baseado no sistema do Apple Lisa, sendo que ambos foram inspirados no sistema do Xerox Alto. Steve Jobs fez um acordo com a Xerox anos antes para que seus engenheiros pudessem estudar o sistema do Alto e usar ideias em troca da venda de ações da Apple para a Xerox a um preço bem mais baixo do que o usual.

Porém, a interface do Mac OS não era simplesmente uma cópia do modelo da Xerox: eles adicionaram vários conceitos novos que não existiam no Alto, como por exemplo, a barra de menus, os menus pop-up e a possibilidade de arrastar e soltar objetos na tela.

O executável do System 1.0 usava apenas 216 kB de memória. Os acessórios incluíam uma calculadora, um despertador, o painel de controle, o bloco de notas (Notepad) e a área de copiar-colar,

chamada Scrapbook. O interessante é que o Notepad permitia representar apenas oito páginas de texto e havia uma razão especial para isso: era tudo o que a memória do Macintosh permitia que o aplicativo usasse, visto que ele não trabalhava com memória virtual e a memória principal do Macintosh mais antigo era muito pequena.

O sistema operacional completo era incluso no pacote do Mac em um disquete de 400 kB. Havia também um outro disco com um manual que ensinava como usar o mouse, um dispositivo que a grande maioria dos usuários de computador na época ainda desconhecia. O manual chamava-se “Mousing Around” e mais tarde passou a ser conhecido como “Mac Basics”.

Havia, no entanto, alguns problemas com esse novo sistema. Por exemplo, não existia um comando para criar um novo diretório. Havia na área de trabalho um diretório chamado “*empty folder*”.<sup>39</sup> Se você o renomeasse, um novo *empty folder* iria aparecer na tela. Assim, a opção de criar um novo diretório existia, mas ela não era intuitiva. Além disso, se você tentasse abrir o disco com o tutorial no Mac, o sistema travava, pois havia uma incompatibilidade que precisou ser resolvida depois. Se o usuário resolvesse reiniciar a área de trabalho, haveria outra dor de cabeça, pois todos os diretórios do usuário seriam deletados e os arquivos contidos neles movidos para o diretório raiz. Também não havia comando para desligar a máquina; você simplesmente desligava ela fisicamente, o que podia causar estragos nos discos e circuitos.

O System 1.1, lançado apenas quatro meses depois do 1.0, resolia parcialmente um dos problemas da versão inicial, pois com o 1.0, copiar um disquete para outro podia levar até 20 minutos, uma vez que blocos muito pequenos do disco eram copiados de cada vez para minimizar o uso de memória. Como isso era muito irritante para os usuários, a versão 1.1 aumentou o tamanho dos blocos copiados, o que reduziu o tempo. Além disso essa versão criou o conceito de

“*startup*”, ou seja, podia-se determinar que um aplicativo fosse iniciado automaticamente quando se ligasse a máquina.

Um ano depois, em 1985, foi lançada a versão 2.0. Nela o “*finder*”, aplicativo para gerenciamento de arquivos, tornou-se muito mais rápido e intuitivo. Finalmente havia um comando para se criar um novo diretório! Além disso, foi introduzido um comando que permitia alternar entre a visualização dos ícones na forma de matriz ou na forma de lista. Foi também concebido o conceito de ejetar discos arrastando-os para a lixeira.

No início de 1986 foi lançado o System 3.0 para acompanhar o Mac Plus, nova versão do Macintosh. Uma das principais mudanças foi a possibilidade de criar diretórios dentro de diretórios, porque até a versão anterior só era possível criar diretórios na raiz; a partir de agora eles poderiam ser aninhados. Outra grande inovação foi o uso de cache de disco: arquivos ou programas frequentemente usados eram mantidos em memória para evitar a necessidade de carregá-los do disco com frequência, com desperdício de tempo.

Ainda em 1986 foram lançadas as versões 3.1, 3.1.1, 3.2 e 3.3, basicamente para corrigir bugs da versão 3.0. A 3.3 incluiu também o Apple Office e um novo sistema de acesso à rede chamado AppleShare. A partir do System 4.0, lançado em 1987, já não havia compatibilidade com o Mac 128, lançado em 1984.

O sistema Mac OS consolidou-se apenas em 1988, com o lançamento do System 6.0, considerado completo, estável e de longa duração. A atualização System 7.0 em 1991 era um upgrade com novos aplicativos e suporte à memória virtual.

A última versão do Mac OS foi lançada em 2002, a 9.19.2. Atualmente os computadores da Apple usam um novo sistema inspirado no Mac OS, mas com base tecnológica no Unix. O sistema é conhecido como Mac OS X.

## 9.34 IBM AT – 1984

O PC AT (Advanced Technology) ou IBM 5170, é considerado parte da segunda geração de computadores pessoais da IBM, sendo a primeira formada pelo PC e XT, ambos baseados no processador 8088. O AT utiliza o processador 80286 a 6 MHz. A configuração típica teria um disco rígido de 20 ou 30 MB e disco flexível de 5¼ com capacidade para 1,2 MB.

Comenta-se que o AT era uma máquina muito rápida para seu tempo, rápida até demais. Havia a possibilidade de se aumentar o clock de 6 para 16 MHz, mas a IBM não tinha interesse em permitir esse aumento, possivelmente para evitar que o computador competisse com outros modelos bem mais caros da própria empresa.

Com o PC AT todos os *slots* de expansão passam a ser de 16 bits e assim a compatibilidade com hardware de 8 bits não era mais um compromisso.

O sistema operacional passa a ser o MS DOS 3.0, capaz de gerenciar os novos drives de disquete de alta densidade e discos rígidos. Uma curiosidade do sistema operacional dessa época era que se você desconectasse o teclado antes de ligar o computador ele mostraria uma mensagem indicando que o teclado não foi localizado e pediria para o usuário teclar F1 para reinicializar!

O layout do teclado também foi modificado, aproximando-se assim do modelo usado em computadores nos dias de hoje, com teclado numérico separado.

Houve três modelos de placa-mãe para o AT. A primeira tinha algumas limitações e foi substituída por uma menor que era mais eficiente. Já a terceira era semelhante à segunda, mas permitia que o processador rodasse a 8 MHz em vez de 6.

## **9.35 Tetris – 1984**

Um dos jogos de computador mais populares, influentes e estudados no mundo foi criado na União Soviética em 1984 por Alexey Pajitnov (Rússia, 1956). Entre 1984 e 2009 há registro de

mais de 125 milhões de cópias vendidas, sem falar em um número indeterminado de versões distribuídas gratuitamente ou pirateadas.

Pajitnov trabalhava na Academia Soviética de Ciências com inteligência artificial e reconhecimento de fala, mas o que lhe interessava mesmo eram jogos psicológicos. Em 1984, comprou um jogo no estilo quebra-cabeças chamado pentaminó. As peças eram muito parecidas com as do *Tetris*, exceto pelo fato de que cada uma era formada por cinco quadrados, e não por quatro. Ele passou a imaginar formas de implementar esse jogo no computador. Inicialmente, reduziu o número de peças porque isso facilitava o jogo e melhorava a performance do processamento das imagens. Sua primeira versão consistia em um conjunto de peças que podiam ser movidas com o cursor para qualquer lado, como no caso do pentaminó. Esse jogo não obteve muito sucesso.

Isso só aconteceu quando ele resolveu adicionar a gravidade ao jogo, fazendo as peças caírem e terem que ser organizadas de forma a completar uma linha de quadrados, que então desaparecia, gerando espaço para mais peças serem encaixadas. O usuário agora só podia controlar a direção horizontal da queda e a orientação da peça, que podia ser girada.

No início de 1985 foi então implementada a primeira versão do *Tetris* em Pascal no computador Eletrônica 60, um clone russo do PDP-11. O jogo ocupava apenas 2,7 kB de memória. O nome *Tetris* veio da combinação de “tetraminó” (um pentaminó de quatro quadrados) e “tênis”, que, por puro acaso, era o esporte preferido de Pajitnov. A tela original do jogo russo pode ser visualizada em: <https://en.wikipedia.org/wiki/File:Tetris-VeryFirstVersion.png>.

O criador do *Tetris* atribui seu sucesso ao fato de que é um jogo fácil de aprender, mas difícil de se aperfeiçoar, o que o torna altamente viciante.

Após a primeira versão ter sido criada, um garoto de 16 anos, chamado Vadim Gerasimov, portou o jogo para o PC usando Turbo Pascal. O primeiro *Tetris* para MS-DOS aparece com copyright de

Pajitnov e Gerasimov. Os dois e mais um colega de Pajitnov, Dmitry Pavlovsky, criaram um grupo para desenvolver e tentar vender um conjunto de jogos nesse estilo. As vendas fracassaram, no entanto, e eles resolveram começar a distribuir o material gratuitamente. Assim, o *Tetris* acabou se tornando bastante conhecido e de longe era o game preferido dos usuários.

Ainda em 1986, o *Tetris* para PC foi portado por programadores húngaros para o Apple II e o Commodore 64. Essas versões chamaram a atenção de Robert Stein, presidente da Andromeda, uma empresa britânica de software. Ele planejava comprar os direitos do *Tetris* PC de Pajitnov e das outras versões dos programadores húngaros, mas acabou causando uma grande confusão, pois mesmo antes de ter um contrato com os russos, ele já havia vendido os direitos do jogo para outras empresas – direitos esses que ele não tinha!

Em novembro de 1986, Stein foi a Moscou com um contrato assinado por ele e buscou na Academia de Ciências quem do outro lado aceitaria o acordo. Voltou de mãos vazias, pois os soviéticos alegaram desconhecimento do mercado mundial de videogames.

Stein planejou, então, roubar o *Tetris* alegando que o jogo foi criado pelos programadores húngaros, com os quais ele já tinha entendimento. Enquanto isso, a Mirrorsoft, que tinha comprado os direitos de Stein, começava a vender o jogo, que logo se mostrou altamente viciante.

Em 1987, Stein consegue o direito de copyright do *Tetris*, mas sem nenhum contrato da parte dos soviéticos. No início de 1988, seu plano de roubar os direitos do *Tetris* é desmascarado quando Pajitnov é entrevistado pela CBS Evening News e apresentado como único criador do jogo. Neste meio tempo, o game já é o mais vendido nos Estados Unidos e Reino Unido.

A história se desdobra em muitas idas e vindas, encontros e desencontros e acabou se tornando um dos casos de litígio mais complexos e longos da história dos videogames, envolvendo um

grande número de empresas mundo afora. O caso acabou se tornando um processo judicial entre a Nintendo e a Atari, que só foi concluído em 1993. Robert Stein ganhou apenas 250 mil dólares com a venda dos direitos que ele alegava ter. Já na Rússia, ninguém ganhou dinheiro com o jogo exceto pelo governo soviético. Pajitnov recebeu um computador 286 e um apartamento melhor que os dos seus colegas como gratidão do governo comunista por ter inventado o game – e isso foi tudo.

Em 1991, ele se mudou para os Estados Unidos e fundou a Tetris Company, por meio da qual conseguiu passar a receber *royalties* pelo jogo, especialmente da Nintendo, que tinha no *Tetris* o carro-chefe do seu novo microconsole, o Game Boy. Entre 1991 e 2005 Pajitnov também trabalhou para a Microsoft na área de entretenimento.

## 9.36 Windows – 1985

Em novembro de 1985 um evento modesto em um hotel de Nova York testemunhou o lançamento do Windows. Essa primeira versão, a 1.0, era ainda bem diferente daquela que no início dos anos 1990 tomaria o mundo dos computadores pessoais de assalto, a 3.1. Uma das características mais marcantes do 1.0, como se pode ver na [Figura 9.23](#), é a impossibilidade de sobrepor janelas. Elas tinham que ficar necessariamente lado a lado. Porém, já havia a noção de *scrolling*, ícones e controle por mouse.



**Figura 9.23: Anúncio da Microsoft sobre o lançamento do Windows 1.0.<sup>40</sup>**

O Windows foi criado de forma até despretensiosa para ser uma “casca” sobre o MS-DOS. O aplicativo deveria se chamar “Interface Manager”, mas o departamento de marketing convenceu Gates que “Windows” seria um nome bem melhor.

A IBM, principal parceira da Microsoft, que usava o MS-DOS em todos os seus PCs não se interessou pelo projeto porque estava desenvolvendo seu próprio aplicativo, o Top View, o qual foi lançado como um aplicativo multitarefa sem interface gráfica. A IBM prometeu a Gates que no futuro a interface gráfica seria implementada, mas isso nunca ocorreu e dois anos depois o próprio aplicativo foi descontinuado pela empresa.

Porém, Gates sabia do potencial desse tipo de interface porque ele tinha visto o Apple Lisa e o Macintosh e acreditava que os PCs podiam chegar a este nível.

O Windows 1.0 era vendido a 99 dólares a cópia. A Apple imediatamente ameaçou processar a Microsoft por quebra de copyright e roubo de segredos industriais porque o Windows, embora fosse lento e tivesse muitos bugs, usava os mesmos menus pop-up e o controle por mouse inventados pela Apple. Gates ofereceu à Apple um acordo para pagar por essas características, que foi aceito.

O Windows deveria permitir que vários programas rodassem ao mesmo tempo, cada um em sua janela, mas o problema de rodar várias aplicações ao mesmo tempo sobre o DOS não era fácil de resolver na época. Para evitar problemas, o Windows permitia que aplicações “bem-comportadas”, ou seja, aquelas que usassem o computador apenas através das funções do DOS, pudessem rodar em janelas lado a lado. No entanto, as aplicações “malcomportadas”, ou seja, aquelas que usam características do computador sem passar pelo sistema operacional, eram obrigadas a rodar em modo de tela cheia. Dessa forma, ao contrário do Lisa e do Macintosh, o Windows já iniciava sua vida com uma infinidade de aplicativos que podiam ser usados, ou seja, quase tudo o que rodava no MS-DOS, enquanto os concorrentes tinham muito poucos aplicativos disponíveis. Isso veio a ser um diferencial competitivo enorme para a Microsoft.

Uma ameaça ao “pesado” Windows era o Sidekick da Borland, um sistema extremamente eficiente em termos de tempo e espaço, que ficava residente em memória e podia ser chamado a partir da tela do DOS, mesmo que houvesse uma aplicação rodando, com um simples pressionar de teclas Control-Alt. O Sidekick oferecia uma calculadora, um bloco de notas e outras utilidades de forma muito mais eficiente do que o sistema da Microsoft, mas talvez não tão atrativa no final, pois, embora usasse o conceito de janelas, o Sidekick era baseado em tela de texto e não em gráficos.

O *New York Times* criticou o consumo de memória pelo Windows, afirmando que a configuração mínima para rodá-lo era de 256 kB de memória principal, mas que a Microsoft avisava em letrelinhas miúdas

que, caso mais de uma aplicação fosse rodar ao mesmo tempo, mais memória seria necessária. O jornal pergunta, então, por que alguém compraria o Windows se não fosse para rodar várias aplicações ao mesmo tempo?

Em 1987 a Microsoft lança o Windows 2.0, uma versão já bem mais parecida com as interfaces do Lisa e Macintosh, bem como das versões mais recentes do próprio Windows, que permitia que as janelas ficassem parcialmente superpostas.

Essa versão também melhorou a operação do Windows com o teclado, visto que uma das maiores críticas ao Windows 1.0 era que ele tinha sido projetado especificamente para uso com o mouse; mas pouquíssimas pessoas na época realmente tinham um desses periféricos.

Essa versão também substituiu os termos “iconizar” e “zoom”, por “minimizar” e “maximizar”, nomes que até hoje são usados para operações sobre janelas.

O novo design e a escolha dos ícones rendeu para a Microsoft um novo questionamento por parte da Apple, que acabou em processo judicial no qual esta acusou a Microsoft e a HP de quebrar 189 de seus copyrights. No final do processo, o juiz reconheceu que a maioria das características mencionadas pela Apple já constavam no acordo do Windows 1.0 entre as duas empresas ou não poderiam ser protegidas – como, por exemplo, a ideia de interface gráfica ou a metáfora de mesa de escritório. Porém, 10 das 189 características realmente eram propriedade intelectual da Apple e não poderiam ser usadas pela Microsoft. O juiz entendeu que apenas a expressão das ideias podia ser protegida por copyright, como, digamos, o desenho específico do ícone da lixeira, mas não a ideia de uma lixeira.

## 9.37 Atari ST – 1985

A Atari Inc., que vendia consoles de videogames e microcomputadores de 8 bits, foi à falência no início da década de

1980 como uma das consequências do *crash* dos videogames – também conhecido como o “choque da Atari”. As vendas que iam na casa dos 3 bilhões por ano em 1983 caíram para 100 milhões em 1985, uma queda de 97%. Esse *crash* quase acabou com a indústria de consoles de videogame, a qual só se recuperou anos mais tarde graças à popularidade da nova Nintendo.

O *crash* teve várias causas, mas a mais importante é apontada como a saturação do mercado. Além disso, a Atari, que tinha iniciado como uma empresa praticamente familiar, acabou sendo vendida para a Warner e administrada por executivos que não entendiam nada de videogames. A companhia estava muito desagregada e mal administrada nessa época. Como as vendas de games cresciam exponencialmente, os executivos acharam que iam continuar crescendo para sempre, e a qualidade dos jogos começou a cair. Um executivo da Atari chegou a comentar uma vez que ele poderia defecar em uma caixa e ainda assim vendê-la e, de fato, parece que foi isso mesmo que eles fizeram.

Em 1982 foram lançados dois títulos altamente aguardados pelo mercado de consoles: *Pac-Man*, que já existia na versão Arcade, e *ET – The Extra-terrestrial* (*E.T., o Extraterrestre*), baseado no filme recém-lançado. A loucura dentro da Atari era tão grande que eles produziram 5 milhões de cartuchos do *ET*, o que consistia na mesma quantidade de consoles que existiam no mundo. Já o *Pac-Man* teve uma produção de 7 milhões de cartuchos, mais do que a quantidade de consoles em existência. A prova de que os novos executivos da Atari não tinham a menor ideia do que estavam fazendo é a justificativa dada por um deles para essa discrepância: ele disse que muitas pessoas iriam querer comprar dois cartuchos, um para sua residência e outro para a casa de campo!

Ambos os jogos foram uma verdadeira deceção e, apesar de *Pac-Man* ter sido um dos games mais vendidos da história dos cartuchos, as vendas ficaram muito abaixo daquilo que a Atari imaginava e tinha produzido. O *ET*, por sua vez, teve que ser desenvolvido e produzido em seis semanas por exigência dos

executivos. O resultado é que ele foi eleito o pior videogame de todos os tempos pois era repetitivo, desinteressante e nem um pouco divertido. Em 1983, a Atari literalmente enterrou 700 mil cartuchos não vendidos, especialmente de *ET* e *Pac-Man*. Aliás, esse é o número oficial que a Atari reconhece, pois há quem diga que tenham sido milhões de cartuchos e milhares de consoles. Muitos duvidaram desta história que, por vezes, foi citada como lenda urbana, mas em 2014 a Microsoft e outras empresas conseguiram autorização para escavar no local do alegado enterro e encontraram alguns milhares de cartuchos. Uma fração deles foi recuperada para ser exibida em museus.

Em 1984, Jack Tramiel, fundador da Commodore, que tinha acabado de sair da empresa, adquire a divisão de consoles e computadores da Atari Inc. de sua proprietária, a Warner Communications. Ele cria então a Tramel Technology Ltd., que em seguida se torna a Atari Corporation. Seu objetivo era investir na próxima geração de computadores pessoais.

A partir de então a Atari Corporation continuou a vender os últimos consoles de videogame ainda em estoque enquanto projetava o novo computador de 16 bits, o Atari ST ([Figura 9.24](#)) com processador Motorola 68000. Ele foi o primeiro computador pessoal a ter 1 MB de memória inicial e custo de menos de 1 dólar por kB de memória.



**Figura 9.24: Atari ST.<sup>41</sup>**

Ele entrou no mercado para competir com o Macintosh, o Commodore Amiga e, posteriormente, o PC com Windows. Utilizava uma interface gráfica chamada GEM (Graphics Environment Manager), muito semelhante à do Macintosh, tanto que chegou a ser apelidado de Jackintosh, em referência ao primeiro nome de Tramiel.

O ST não vendeu muito bem nos Estados Unidos, em parte devido à má reputação da antiga Atari, uma empresa com muitos problemas, e em parte também pela fama de Tramiel, conhecido por ser tão fácil de lidar quanto Átila, o Huno. Porém, o ST foi um grande sucesso na Europa, onde chegou a vender 5 milhões de unidades; afinal, apesar da reputação, ele era mais rápido e mais barato que os competidores. O computador foi muito usado por músicos porque tinha processadores especiais para som e MIDI (Musical Instrument Digital Interface<sup>42</sup>), inventado poucos anos antes e padronizado em 1983.

Quando o hardware estava quase pronto era necessário pensar em um sistema operacional. A Microsoft pediu dois anos de prazo para portar o Windows para o novo computador e Tramiel achou o tempo muito longo. Ele acabou fechando com a Digital Research, que desenvolveu o GEM como interface gráfica e o GEMDOS, baseado no CP/M como sistema operacional.

O ST foi produzido até 1993, quando foi substituído por um modelo superior, o Jaguar.

## 9.38 Commodore Amiga – 1985

O Commodore Amiga, lançado em 1985, era um computador fantástico e 10 anos à frente do seu tempo. Ele custava relativamente pouco e era capaz de processar som, gráficos e vídeo com a mesma facilidade que os outros computadores de sua época processavam texto. Ele só teve um grande problema: o mundo praticamente não ficou sabendo da sua existência.

Muitas pessoas contribuíram para a construção do Commodore Amiga, mas talvez a principal delas, o visionário, tenha sido Jay Miner (Estados Unidos, 1932-1994). Jay trabalhou para a Atari Inc. desde o seu início, tendo ficado envolvido com a criação do primeiro videogame Arcade, o *Pong*, e o famoso console de jogos Atari 2600.

O ambiente de trabalho na Atari foi bom até 1978, quando ela foi vendida para a Warner Communications e a nova administração, que entendia muito pouco dessa indústria, começou a criar problemas, por exemplo, negando aos desenvolvedores de jogos os *royalties* pelas suas criações. Jay, por sua vez, estava entusiasmado pelo processador Motorola 68000 e imaginava criar um incrível computador a partir dele. Como os executivos da Atari não queriam saber disso, desanimado ele deixou a empresa em 1982.



**Figura 9.25: Commodore Amiga.<sup>43</sup>**

Miner passou algum tempo em uma empresa de eletrônicos, mas logo recebeu um telefonema de seu amigo e antigo chefe na Atari, Larry Kaplan (Estados Unidos, 1949), que o convidou para abrir uma nova empresa na qual Larry faria os jogos e Jay o hardware. O nome escolhido para a empresa foi Hi-Toro, porque era *high-technology* e texana.

Porém, antes do final do ano, Larry voltou para a Atari quando ela lhe fez uma proposta salarial irrecusável. Jay então assumiu o controle da nova empresa, mas com uma condição: a máquina de jogos deveria ser construída com o processador Motorola 68000 e também deveria funcionar como um computador.

Um computador e um console de jogos são basicamente a mesma coisa, muda apenas a forma de interação com o usuário, visto que o console serve apenas para se jogar, enquanto o computador serve para muitas outras tarefas. Na época, entretanto, o mercado de videogames ultrapassava em muitas vezes o mercado de microcomputadores e os investidores queriam lucro. Assim, a opção por expandir o console de jogo para um computador deveria ser tratada com muita discrição pelos engenheiros.

Em 1983 houve o crash da Atari, sobre o qual já falamos. Em função da má qualidade dos jogos e das limitações dos consoles, as pessoas simplesmente começaram a achar que ter um videogame não era mais tão interessante. As vendas de consoles e cartuchos despencaram dramaticamente, mas a comercialização de microcomputadores continuava crescendo em um ritmo constante. As famílias compravam computadores que podiam ser usados com finalidade educacional e as crianças adoravam a máquina porque nela também se podia jogar videogames.

Imaginem a alegria de Jay Miner quando os investidores da Hi-Toro, nervosos, lhe pediram se poderia converter o projeto do novo videogame em um computador!

Porém, havia um problema: o nome da empresa, Hi-Toro, já estava registrado para uma empresa japonesa e precisava mudar. Jay queria um nome amigável e sexy, assim usou a palavra espanhola “Amiga”.

O processo de produção de um computador toma tempo, no entanto, e a empresa precisava de renda. Assim, eles desenvolveram inicialmente alguns acessórios e jogos para videogames. Um deles era o Joyboard, uma plataforma semelhante a uma balança de banheiro na qual você poderia sentar ou ficar em pé e pressionando para os lados teria um movimento semelhante ao do joystick. Um dos programadores do Amiga usou esse equipamento para criar um jogo chamado *Zen Meditation* (*Meditação Zen*). O objetivo era sentar sobre ele e conseguir ficar

absolutamente parado. O jogo virou uma espécie de piada entre os programadores e quando chegou a hora de escrever uma mensagem de erro para uma falha séria do computador, um deles propôs: “Erro de Meditação do Guru.” A mensagem foi de fato usada por algum tempo até que um executivo mais sério mandou substituir por um insosso “Falha de Software”.

Outra prática dos engenheiros era bater na cabeça do outro com um bastão de espuma caso um deles apresentasse ideias estúpidas ou desnecessárias. Não doía fisicamente, mas ninguém queria se humilhar dessa forma, e então eles caprichavam.

Jay era apaixonado por simuladores de voo e, após visitar um simulador militar de milhões de dólares, pensou que seu computador poderia rodar o melhor modelo possível. Para isso, resolveu usar um *blitter* – um circuito que move grandes blocos de gráficos pela tela sem ocupar o microprocessador, um conceito bem à frente de seu tempo.

Em relação aos gráficos, as poucas cores e *palettes* do PC não eram aceitáveis. Ele queria milhares de cores, mas a memória para guardar esses milhares de possíveis cores para cada pixel na tela seria absurdamente cara. Então, resolveu usar o conceito de HAM (Hold And Modify). Basicamente, a ideia era determinar a cor do primeiro bit e indicar para os bits seguintes apenas o que mudava em relação ao anterior em termos de saturação, luminosidade ou matiz. Assim, o Amiga foi lançado com a capacidade de exibir 4.096 cores, muito mais do que seus competidores.

Outra característica do Amiga foi o uso de sprites, ou seja, objetos gráficos que podiam ser movidos pela tela sem que o fundo tivesse que ser redesenhado. Com todas essas inovações, a máquina teria que ser bem diferente das antecessoras. Todos os microcomputadores até então eram montados com circuitos integrados que já existiam e podiam ser comprados facilmente, mas o Amiga, além do Motorola 68000 teria ainda outros circuitos especiais que sequer existiam e precisavam ser criados

especialmente para ele. Isso tornou o projeto um desafio ainda maior.

O sistema operacional deveria ser multitarefa, ou seja, capaz de executar vários programas ao mesmo tempo. Quando estava entrevistando um candidato a programador do sistema operacional, Jay perguntou qual seria seu trabalho dos sonhos; quando ele respondeu “construir um sistema operacional multitarefa” ele foi contratado na hora. Microcomputadores da época faziam multitarefa eventualmente e normalmente a um alto custo de memória e performance, como no caso do Windows. Depois do Amiga, a capacidade real de multitarefa só foi incluída no Windows em 1995 e no Macintosh em 2001 com o OS X.

A interface gráfica era outro problema. Antes de tudo, precisava ser construída uma API (Application Programming Interface<sup>44</sup>) que é o conjunto de objetos mais básico como os menus, janelas e ícones que vão ser usados depois pelos programas. Jay passou a tarefa a um de seus programadores, aquele do programa “Zen”, e ele se trancou em uma sala por três semanas e saiu de lá com uma API muito bem organizada, a Intuition. Já a API do Windows, chamada Win16, foi feita por uma grande equipe e acabou ficando uma colcha de retalhos no final, no estilo que programadores detestam.

Quando o primeiro protótipo do Amiga ficou pronto e resolveram apresentá-lo em uma feira, eles ficaram preocupados com a possibilidade de ele quebrar no transporte. Portanto, resolveram comprar uma terceira passagem de avião e foram dois engenheiros nas pontas e o protótipo no meio embrulhado em dois travesseiros para maior segurança. No ticket informaram o nome “Joe Pillow”<sup>45</sup> e desenharam na frente da caixa um rosto sorridente e uma gravata. Os atendentes do voo se recusaram a servir uma refeição a este passageiro.

Como não havia software disponível para a demonstração das capacidades do computador, os dois engenheiros passaram uma noite inteira criando um programa que mostrava uma bola xadrez

quicando pela tela e produzindo uma sombra em um plano quadriculado. Cada vez que a bola quicava produzia um som estéreo que fora criado com um dos bastões de espuma que os engenheiros usavam para bater na cabeça dos outros na porta da garagem. Essa imagem da bola acabou virando o símbolo do Amiga.

Por essa época, porém, a empresa estava ficando sem dinheiro e precisava pagar os credores. O principal era a própria Atari, que havia emprestado 500 mil dólares à Amiga, que tinha que devolver o valor até certa data ou então a Atari se tornaria dona dela. No último minuto, a Commodore fez uma proposta de comprar a empresa, pagou os 500 mil para a Atari e mais 24 milhões para se tornar proprietária da Amiga. Assim, o computador de Jay entrou na história como o Commodore Amiga.

Porém, Jack Tramiel, o fundador da Commodore, tinha sido removido de sua posição na empresa e comprado a Atari, como vimos anteriormente. Ele fez do combate à Commodore uma questão pessoal: não bastava competir, era necessário destruir os concorrentes, especialmente a Commodore. E, além de tudo, ela tinha tirado o Amiga de suas mãos.

Assim, os anos seguintes foram de uma guerra de preços entre a Commodore e a Atari, com seu modelo ST, criado especialmente para competir com o Amiga. No final, ambas as empresas pereceram e o IBM PC, muito menos avançado e muito mais caro, seguiu incólume como líder do mercado de microcomputadores.

## 9.39 PageMaker – 1985

Um programa inovador foi lançado em 1985. O Aldus PageMaker permitia que qualquer usuário de computador fosse capaz de criar folhetos, cartões, revistas, livros etc. com qualidade editorial. A recém-fundada empresa Aldus de Seattle era comandada por Paul Brainerd (Estados Unidos, 1947), graduado em Jornalismo. Em 1980, ele trabalhou para uma companhia que desenvolvia

equipamento computacional profissional para jornais, mas ela fechou em 1983. Assim, ele e outros ex-funcionários resolveram fundar a própria companhia. O nome, Aldus, foi uma homenagem a Aldus Pius Manutius (Itália, 1449-1515), um dos pioneiros da arte da publicação, tendo fundado a primeira editora, a Aldine, e estabelecido vários padrões editoriais e tipos de letra, inclusive o itálico.

O PageMaker aumentou ainda mais o mercado de microcomputadores com capacidades gráficas. A Apple rapidamente fez acordos com a Aldus para que ele fosse disponibilizado com o recém-lançado Macintosh e a Microsoft fez dele um dos aplicativos do Windows 1.0.

A base tecnológica do PageMaker era a linguagem de descrição de documentos PostScript, da empresa Adobe. O software era vendido a 495 dólares, um pouco caro para um usuário doméstico, mas nem tanto para um profissional das artes gráficas que com ele revolucionaria sua forma de trabalhar.

O aplicativo foi implementado em praticamente todos os computadores que existiam na época e foi líder absoluto de mercado por alguns anos. Porém, a partir de 1992 a Aldus começou a perder para um concorrente, o QuarkXPress, até ser praticamente retirada do mercado em 1998. Acabou sendo vendida para a Adobe, que relançou o produto como Adobe PageMaker. Em 2001 a Adobe o rebatizou como Adobe InDesign, mas ele continua sendo evoluído praticamente pela mesma equipe da Aldus em Seattle.

## 9.40 Acorn ARM – 1985

Já falamos sobre a Acorn, empresa britânica responsável pelo sucesso do BBC Micro em 1982. No ano seguinte a empresa se preocupava com a próxima geração de computadores e resolveu investir em um projeto para produzir um processador RISC, que era uma tecnologia que estava começando a se solidificar.

O projeto do ARM (originalmente Acorn RISC Machine, depois rebatizado como Advanced RISC Machine) foi realizado por Steve Furber (Reino Unido, 1953), professor de computação da Universidade de Manchester, Sophie Wilson<sup>46</sup> (Reino Unido, 1957), que trabalhou no projeto do BBC Micro, e Roberto Heaton, que dirigia o grupo de design VLSI dentro da Acorn.

O novo processador deveria conter a inteligência do MOS 6502, mas em uma versão RISC de 32 bits. Ele também deveria ser fácil de testar e usar e produzido a baixo custo.

O conjunto de instruções e modelo do hardware foram todos programados em BASIC e depois materializados na forma de um circuito VLSI. Um simulador de eventos para testar o circuito foi também desenvolvido em BASIC e depois reescrito em Modula 2 e finalmente em C. Esse sistema ainda hoje é usado pela Acorn e ARM Ltd. para projetar e testar chips.

O ARM-1 entrou em produção em abril de 1985, um dos primeiros microprocessadores RISC a existir. O número de transistores usados no processo era surpreendentemente baixo: apenas 25 mil. A experiência com o ARM-1 levou logo depois a um novo design mais eficiente e mais completo, o ARM-2.

Porém, em 1984 o mercado de microcomputadores sofreu uma forte queda na Inglaterra, e a Acorn estava em dificuldades financeiras. Ela acabou sendo vendida em 1985 para a italiana Olivetti, então uma das maiores fabricantes de computadores e equipamentos de escritório da Europa. No entanto, o projeto do ARM era tão secreto que a Olivetti sequer foi informada sobre sua existência.

O primeiro uso comercial do ARM em um computador ocorreu em 1987 no Archimedes, sobre o qual ainda falaremos. A evolução do ARM não parou, no entanto. Em 1989 foi lançado o ARM-3 com um aumento significativo de velocidade, funcionando a 25 MHz. Ele entrou em uso em computadores em 1990.

O crescente interesse por microprocessadores RISC levou a um acordo entre a Acorn, a VLSI Technology e a Apple, que culminou

na criação de uma nova empresa em 1990, a Advanced RISC Machines Ltd., ou ARM Ltd. A companhia cresceu e estabeleceu relações amigáveis com muitas outras empresas, inclusive potenciais competidores como a Intel, com a qual ela mantém negócios e projetos conjuntos. Ela sempre teve uma postura de cooperação ao invés de competição.

Possivelmente, essa postura levou a empresa a ter um crescimento sustentado e contínuo. Ela não parou de crescer até hoje. Em 2008 já havia produzido 10 bilhões de chips e nesse ano se estima que 98% dos smartphones vendidos no mundo tivessem pelo menos um chip ARM incluso. Até 2014 o número de chips produzidos alcançou a marca de 50 bilhões e continua crescendo.

Além disso, Steve Balmer da Microsoft anunciou recentemente que pretende basear as futuras gerações do Windows nos sistemas operacionais para microchips desenvolvidos pela companhia. O objetivo da ARM para 2015 foi ter um de seus chips em pelo menos 50% de todos os notebooks, tablets e microcomputadores vendidos no mundo.

## 9.41 CD-ROM – 1985

Até 1985 se você quisesse distribuir software de grande porte precisava usar fitas magnéticas. Na computação pessoal poucas pessoas tinham leitores para esse tipo de fita e isso ficava mais restrito ao mercado de mainframes. Como a Internet também não era muito acessível nesse mercado, ela também não era uma opção para distribuição de software e dados. Restavam os disquetes. Porém, a capacidade de um disquete era muito limitada; dificilmente passaria de 1,4 MB, ou seja, para gravar uma simples fotografia de uma máquina digital moderna, seriam necessários hoje uns quatro daqueles disquetes. Para uma foto! Imaginem, então, filmes. Era impossível. A indústria de jogos para computador sofria com isso.

Porém, em 1985 entrou em uso uma nova tecnologia que permitiu gravar em um único disco 650 MB, ou seja, um único disco teria a

mesma capacidade que mais de 450 disquetes. Nele caberia mais de 1 milhão de páginas de texto. E isso numa época em que a maioria dos discos rígidos de microcomputadores não passavam de 40 MB. Esse novo equipamento era o CD-ROM, ou Compact Disk – Read Only Memory. Trata-se de um tipo de mídia não magnética, mas ótica. Um CD-ROM tem sua superfície formada por espelhos de tamanho microscópico que são orientados de forma a refletir ou não um finíssimo raio laser, que ao ser corretamente refletido para um sensor, faz com que as informações gravadas no CD possam ser usadas em um computador eletrônico.



Figura 9.26: Um CD-R.<sup>47</sup>

Essa tecnologia não surgiu de repente. Ela tem uma longa história que resumiremos nas próximas linhas.

O raio laser foi inventado em 1961 por Theodore Maiman (Estados Unidos, 1927-2007), mas a ideia de codificar informações em uma superfície reflectora e ler essas informações com um raio laser é de James T. Russel (Estados Unidos, 1931), que requereu patente em 1966 e a obteve em 1970. Porém, não foi ele quem ficou com a patente, mas sim a empresa na qual ele trabalhava, a Battelle. Em

1973, ela constrói o primeiro protótipo do discolaser e passa a divulgar a ideia para gravadoras como a Sony e a Philips.

Em 1979, a Sony e a Philips iniciam juntas o desenvolvimento da tecnologia de discolaser para distribuição de música. As duas empresas chegam a um acordo sobre padrões para esse tipo de disco em 1980, que ficam registrados no chamado “Livro Vermelho” (*Red Book*). Em 1982, o discolaser se torna produto comercial e Russel nunca recebeu um centavo por ele.

Em 1983, o “Livro Amarelo” (*Yellow Book*) apresenta as bases para o desenvolvimento de CD-ROM, ou seja, discos que poderiam conter dados em formato que um computador poderia usar. Em 1985 é lançado o primeiro sistema de computador vendido em CD-ROM. Tratava-se da *Grolier’s Electronic Encyclopaedia*, que tinha 9 milhões de palavras e usava apenas 12% da capacidade do CD. Em 1987 a Microsoft também lança um CD-ROM, este com propósitos educacionais, o Bookshelf.

Em 1988 a Sony e a Philips publicam o “Livro Laranja” (*Orange Book*), no qual descrevem o novo CD-WO, ou Compact Disk-Write Once, um tipo de CD que podia ser vendido “virgem” e gravado por quem tivesse um drive especial capaz de efetuar essa gravação. Em 1990, são comercializadas as primeiras gravadoras de CD e a partir dessa época eles passam a ser chamados de CD-R, ou Compact Disk-Recordable. Porém, uma gravadora de CD nessa época tinha o tamanho de uma máquina de lavar-louça e custava até 35 mil dólares. Com o tempo, os preços baixaram e por volta de 1995 já estavam abaixo da casa dos 1.000 dólares.

Em 1997 um novo conceito surge: CDs que podiam ser gravados e regravados várias vezes. Surgia assim o CD-RW, ou Compact Disk – ReWritable. A partir desse momento já não fazia mais sentido falar em CD-ROM, porque eles não eram mais “*read only*”. Mesmo assim, o nome permaneceu em uso. No Brasil, às vezes doíam os ouvidos quando alguém falava em “CD-ROOM” ou “cedê-rum”, assassinando a pronúncia correta do acrônimo em inglês.

Outro problema comum na época é que alguns aparelhos de leitura foram construídos para a especificação dos CDs, gravados em indústrias, que tinham uma superfície bem mais refletiva do que os CD-R e muito mais do que os CD-RW. Por isso, algumas vezes, esses aparelhos não conseguiam ler CDs-R e, com muito mais frequência, não liam CDs-RW.

Em 1997 começa o declínio do CD com o lançamento da tecnologia de DVD, capaz de armazenar até 4,7 GB. O DVD-ROM foi criado pela Pioneer com a colaboração de empresas de computação como a Sun, Apple, Dell e outras. Logo drives para DVD-R e DVD-RW começaram a se tornar disponíveis nos computadores a preços cada vez mais baixos.

Porém, mesmo o DVD teve vida relativamente curta, pois no ano 2000 já iniciava a pesquisa dos DVR Blue, depois batizados como Blu-ray, com capacidades variando de 25 a 125 GB. O primeiro modelo comercial foi lançado no Japão pela Sony em 2003. O Blu-ray é hoje uma das principais formas de distribuição (além da internet) de jogos para os consoles PlayStation 3 e 4 e Xbox One.

Na indústria de informática, porém, esses dispositivos estão cada vez mais em desuso. Com o aumento da velocidade da Internet, filmes já podem ser assistidos online, sem necessidade de mídia. Além do que, para transportar dados fisicamente, a maioria das pessoas prefere pendrives, muito menores que discos, porém mais fáceis de perder.

## 9.42 Connection Machine – 1986

A Connection Machine, ou CM, foi resultado da tese de doutorado de W. Daniel Hillis (Estados Unidos, 1956) que buscava uma arquitetura de processamento altamente paralela que pudesse funcionar de uma forma diferente da maioria dos computadores que tinham processadores cada vez mais poderosos, mas que via de regra possuíam apenas um processador para executar os programas. O computador de Willis teria 1 milhão de processadores,

cada qual bem simples, mas todos se comunicando entre si e trabalhando ao mesmo tempo.

Um dia, no início da década de 1980, Hillis estava almoçando com Richard Feynman (Estados Unidos, 1918-1988), Prêmio Nobel de Física que trabalhou em Los Alamos no projeto da bomba atômica. Feynman também foi pioneiro da computação quântica e da nanotecnologia. Neste almoço, Hillis disse a Feynman que estava planejando abrir uma empresa para construir um computador com 1 milhão de processadores. Feynman respondeu que era de longe a ideia mais boba que já tinha ouvido em toda a sua vida, mas como para ele uma ideia louca era uma oportunidade interessante, ofereceu-se para trabalhar nessa empresa no verão.

Feynman viajou para Boston no dia seguinte ao início das operações da empresa em 1983. O grupo de recém-graduados do MIT estava há meses sem conseguir pensar em nada técnico; só tinham na cabeça problemas relacionados a local para trabalhar, compra de móveis e equipamentos, qual deveria ser o nome da empresa, sem falar no choque de ter alguns milhões de dólares de investidores na conta da companhia. Feynman teria simplesmente chegado e falado: “Richard Feynman se apresentando para o trabalho! Ok, chefe, qual a minha tarefa?”

Hillis e os outros não tinham a menor ideia sobre o que passar para ele e, após alguns cochichos entre o grupo, sugeriram que a tarefa dele seria dar orientações sobre a aplicação de processamento paralelo a problemas científicos. Feynman respondeu que isso não era uma tarefa de verdade e pediu algo real para fazer. Assim, mandaram-no comprar algumas canetas. Enquanto Feynman estava fora, eles se reuniram e chegaram à conclusão de que uma boa tarefa para ele seria analisar o roteador que controlaria o envio de mensagens de um processador para outro, pois não tinham certeza de que o seu design funcionava.

A máquina em si deveria ter 1 milhão de processadores, mas ligar todos entre si seria impraticável, pois formaria em torno de 500

bilhões de conexões. Assim, em vez de conectar cada processador a 999.999 processadores, resolveram ligar cada processador a 20 outros processadores, organizando a estrutura toda na forma de um hipercubo, ou seja, um cubo de 20 dimensões.

Dessa forma, enquanto Feynman se concentrava na tarefa, a equipe organizava a empresa. Ocasionalmente, Feynman também ajudava a instalar os cabos de rede, configurar a máquina de café, cumprimentar investidores, instalar telefones e lembrar a equipe de como eles eram malucos. Quando decidiram que o nome da empresa seria “Thinking Machines Corporation”<sup>48</sup> Feynman se mostrou contente e disse que agora finalmente não precisava mais explicar às pessoas que trabalhava com um bando de lunáticos – bastaria ele dizer o nome da companhia.

Em algum momento, uma série de seminários foi organizada e, entre outras pessoas, John Hopfield (Estados Unidos, 1933) da CalTech foi convidado para apresentar seu novo conceito de memória associativa baseada em rede neural artificial. A rede era capaz de aprender a reconhecer padrões (como, por exemplo, letras escritas à mão). Mesmo que cada imagem de uma letra fosse diferente da anterior, a rede seria capaz de saber que letra era a partir do padrão do desenho. Esse tipo de estrutura funcionaria maravilhosamente bem com a Connection Machine, e seria centenas de vezes mais rápida do que em qualquer outro computador da Terra.

Em relação à tarefa de Feynman, ele terminou a análise em 1983 e apresentou ao grupo um conjunto de equações diferenciais parciais, algo trivial para um físico, mas nem tanto para programadores. Após algum esclarecimento, foi entendido que cada chip deveria ter cinco buffers (uma memória temporária). Como uma análise intuitiva dos programadores indicava um mínimo de sete buffers, resolveram optar pela segurança e ignorar o resultado de Feynman. Porém, quando isso tornou o projeto demasiadamente complexo, voltaram

atrás e implementaram os chips com cinco buffers. Felizmente para todos, Feynman estava certo.

O último projeto em que Hillis e Feynman trabalharam juntos no CM foi uma simulação da evolução de uma população finita que se reproduzia sexualmente. Hillis ficou impressionado ao perceber que a população evoluía aos saltos e não de forma contínua e gradual como ele esperava. Eles foram aprofundando suas descobertas até Hillis descobrir que tudo isso já estava publicado nas páginas iniciais de um livro de biologia. Feynman então teria ditto: “*Hey, we got it right! Not bad for amateurs.*”<sup>49</sup>

Em relação à máquina em si, o primeiro protótipo, CM-1, tinha “apenas” 65 mil processadores, cada qual com 4 kb de memória e capaz de processar 1 bit por vez. Ele tinha a forma de um cubo de um metro e meio dividido internamente em oito cubos menores. Cada um dos cubos menores tinha 16 placas de circuito impresso, cada qual contendo 32 chips. Cada um dos chips tinha 16 processadores com suas memórias. O CM-1 foi produzido em 1986.

Em 1987 foi lançado o CM-2 ([Figura 9.27](#)) com coprocessadores aritméticos de ponto flutuante e mais memória. Em 1991, a empresa lança o CM-5 agora baseado em chips Sparc RISC e com uma arquitetura diferente do hipercubo. Ele era um dos mais potentes computadores de seu tempo.



**Figura 9.27: CM-2.<sup>50</sup>**

## **9.43 Eiffel – 1986**

A linguagem Eiffel não chegou a ser exatamente um sucesso na história da computação, mas os conceitos que ela introduziu tiveram grande impacto em outras linguagens muito importantes atualmente, como Java e C#.

O nome da linguagem foi escolhido como uma homenagem a Gustave Eiffel, o engenheiro responsável pela torre de mesmo nome construída em Paris em 1887 para a Feira Mundial de 1889. A torre foi construída dentro do prazo e dentro do orçamento, exatamente como se espera dos projetos de software construídos com Eiffel. Além disso, ela é extremamente grande e complexa, mas construída com padrões de projeto robustos e repetíveis, exatamente como um grande projeto de software deveria ser.

Eiffel foi projetada em 1995 pela Eiffel Software inicialmente como ferramenta interna para desenvolvimento de aplicações. Os engenheiros de software da empresa precisavam de uma linguagem orientada a objetos robusta, eficiente e com os mais modernos

conceitos de engenharia de software, como “design por contrato”, usando pré e pós condições para garantir que operações executadas receberam seus dados adequadamente e entregaram resultados conforme esperado, e “invariantes”, que garantem que determinadas regras que devem sempre valer para os dados nunca sejam violadas por qualquer operação possivelmente executada sobre estes.

O primeiro compilador de Eiffel foi criado por Bertrand Meyer (França, 1950), o fundador da empresa. O projeto foi apresentado na primeira conferencia OOPSLA (Object-Oriented Programming, Systems, Languages, and Applications), e foi muito bem recebido. Logo depois, Eiffel foi lançada como um produto comercial, sendo adotada largamente na Europa, América do Norte e Extremo Oriente. Muitas universidades passaram a usar Eiffel como linguagem para ensino de programação, devido às suas estruturas limpas, concisas e poderosas.

A linguagem tornou-se ainda mais conhecida a partir de 1988 quando Bertrand Meyer publicou seu *best-seller Object-Oriented Software Construction*, um livro que passou a ser adotado em muitas universidades por apresentar de forma clara e concisa conceitos fundamentais de orientação a objetos e usando exemplos com a linguagem Eiffel. Ele ainda hoje é considerado um dos grandes clássicos da engenharia de software.

Atualmente, Eiffel é mantido na forma do EiffelStudio, uma IDE (Integrated Development Environment), mas apesar de suas excelentes qualidades, a linguagem sequer aparece na tabela TIOBE entre as 50 mais populares.

## 9.44 Amstrad PC-1512 – 1986

Até 1986 o PC ainda não tinha conquistado muito mercado na Europa. Isso começou a acontecer quando a Amstrad lançou um clone do PC, denominado PC-1512, no qual “1” significava que era a primeira máquina desse tipo lançada por eles e “512” era uma

referência ao tamanho da memória da máquina, considerada ainda bem grande para a época, especialmente porque a Amstrad até então só vendia computadores de 8 bits do tipo Sinclair, baseados em Z80.

O IBM PC não era difícil de clonar pois era todo feito com peças que se podia comprar no mercado. Não havia muita coisa nele que fosse segredo ou propriedade industrial da IBM. O nome interno pelo qual o 1512 era conhecido na Amstrad era “Airo”, ou seja, “Amstrad’s IBM Rip Off”.<sup>51</sup>

Um sistema operacional teria que ser escolhido para a máquina. Por essa época, a Microsoft tinha acabado de ultrapassar a Digital Research nas vendas de sistemas operacionais. Sir Alan Sugar (Reino Unido, 1947), dono da Amstrad, procurou a Microsoft para licenciar o MS-DOS para o 1512, mas o preço foi considerado muito alto. Sabendo disso, e vendo a possibilidade de voltar à liderança no mercado de sistemas operacionais, a Digital Research ofereceu o DR-DOS a um preço agressivamente baixo e também a interface gráfica GEM, concorrente do Windows.

Porém, um pouco antes do lançamento do 1512, a Microsoft acordou e se deu conta de que esse computador dominaria o mercado de PCs na Europa. Decidiu, então, abaixar muito o seu preço. No final, o 1512 acabou sendo lançado com as duas opções de sistema operacional: MS-DOS com Windows e DR-DOS com GEM. A maioria dos usuários escolheu o MS-DOS em função de liderança de mercado.

O computador tinha algumas características únicas, por exemplo, a fonte de alimentação que era construída junto com o monitor CRT. Assim, o resfriamento de ambos os dispositivos se dava por convecção, ou seja, o ar quente gerado pela fonte e monitor subia com força suficiente para sugar ar frio pela parte de baixo do gabinete, resfriando assim o sistema. Porém, essa estratégia genial do ponto de vista da engenharia foi um desastre do ponto de vista de marketing, porque embora a ausência de um ventilador interno

fizesse o computador ser muito mais silencioso do que outros do mesmo modelo, ele ganhou uma fama indevida de ser uma máquina que esquentava muito. Assim, em modelos posteriores, eles acrescentaram um ventilador só para acalmar os usuários. Sugar teria dito em uma entrevista a jornalistas: “*I’m a realistic person and we are a marketing organisation, so if it’s the difference between people buying the machine or not, I’ll stick a bloody fan in it. And if they want bright pink spots on it I’ll do that too. What is the use of me banging my head against a brick wall and saying ‘You don’t need the damn fan, sunshine?’*”<sup>52</sup> Os ventiladores foram colocados e, surpreendentemente, pontinhos cor-de-rosa começaram a aparecer nos lugares mais inusitados em anúncios e nos escritórios da empresa.

O 1512 era muito mais barato do que um IBM PC equivalente no mercado europeu. Sugar apostou que a máquina venderia pelo menos 70 mil unidades por mês. Havia dois modelos, um mais barato com drive de disquetes apenas e um modelo que custava mais do que o dobro, mas que vinha com um disco rígido de 20 MB. O erro de Sugar foi apostar que o modelo mais barato venderia mais. Foi exatamente o contrário: a esmagadora maioria dos compradores queria o modelo com o disco rígido que, embora mais caro, valia o investimento. No entanto, não havia uma quantidade suficiente desses modelos em produção e portanto a empresa começou a falhar em atender pedidos.

Além disso, os boatos de que a máquina não seria confiável, iniciados por causa da celeuma do ventilador, fizeram com que muitas empresas preferissem comprar máquinas mais caras da própria IBM.

A partir de 1987, a Amstrad lançou um novo modelo com mais memória, mas nessa época a maioria dos PCs já estavam usando o processador 80286 ou 80386 e essa máquina já seria defasada no lançamento. Além disso, os novos PCs baseados nesses processadores já não eram mais tão fáceis de clonar e os negócios

da Amstrad foram descendo pelo ralo. Em 1997 suas ações foram divididas entre várias empresas e hoje ela é uma subsidiária da BSkyB e não produz mais computadores.

## 9.45 Compaq Deskpro 386 – 1986

Quando falamos na evolução dos computadores baseados em circuitos Intel x86, esperamos ver sempre a IBM saindo à frente dos concorrentes no uso da tecnologia. Entretanto, isso não aconteceu quando a Intel lançou o microprocessador de 32 bits 80386. A primeira empresa a disponibilizar uma máquina com esse processador não foi a IBM, mas a Compaq.

Desde 1984 a Compaq vinha produzindo modelos de computador do tipo desktop, baseados na família x86. O seu primeiro XT já usava um 8086, que era um pouco melhor do que o 8088 da IBM. Em 1985 ela lança o Deskpro 286, muito similar a um PC/AT, e em 1986 já introduz também o Deskpro 386.

O Intel 80386 foi lançado pela Intel em 1985. Ele tinha 275 mil transistores, muito mais do que os 25 mil do processador RISC Acorn ARM, lançado mais ou menos na mesma época. Porém, o 386 era um processador CISC e, por isso mesmo, mais complexo. Ele era quase totalmente compatível com o código do antecessor 286. No início da produção, a Intel verificou que alguns chips tinham um erro quando usados com software de 32 bits. Como nem todos os circuitos tinham o problema, a Intel testou o seu estoque completo e separou os chips em dois grupos: os que funcionavam perfeitamente foram marcados com um duplo sigma ( $\Sigma\Sigma$ ) e isso significava que funcionavam bem em 32 bits. Já os defeituosos foram marcados como “16 BIT S/W ONLY” e foram vendidos do mesmo jeito porque aplicações de software que exigiam 32 bits eram muito raras na época.

Ele foi inicialmente produzido na versão DX, mas em 1988 a Intel lançou um modelo simplificado e bem mais barato, o SX, que tinha menos capacidade de endereçamento e menos performance, mas

era mais rápido do que um 286. Na [Figura 9.28](#) podemos ver o processador Intel 80386SX dentro de um Compaq Deskpro 386.



**Figura 9.28: Processador Intel 80386SX dentro de um Compaq Deskpro 386.**<sup>53</sup>

Inicialmente vendido a um alto custo, os preços baixaram na medida em que passou a ser usado como padrão na indústria de computadores. Embora ele logo tenha sido suplantado pelo 80486, com 1,2 milhão de transistores, o 386 continuou sendo fabricado até 2007 pela Intel, pois era usado em microcontroladores e outros dispositivos, inclusive na indústria espacial e de instrumentos eletrônicos.

## 9.46 Pixar – 1986

A história do maior e mais influente estúdio de animação por computador, a Pixar, remonta a pelo menos 1974, quando Alexander Schure (Estados Unidos, 1920-2009), dono de um estúdio de animação tradicional, criou um grupo de computação gráfica com o objetivo de produzir o primeiro desenho animado totalmente computadorizado. Os quatro membros originais do grupo foram Ed

Catmull (Estados Unidos, 1945), Malcolm Blanchard, Alvy Ray Smith (Estados Unidos, 1943) e David DiFrancesco (Estados Unidos, 1949). Em 1979, Ed Catmull e vários outros funcionários foram contratados por George Lucas (Estados Unidos, 1944) para compor a Lucasfilm Computer Division. Lucas estabeleceu os seguintes objetivos para o grupo:

- Um sistema de edição de filmes digitais não linear.
- Um sistema de edição de som digital.
- Uma impressora de filmes digitais.
- Pesquisa exploratória em computação gráfica.

Em 1984 foi apresentado na conferência da ACM SIGGRAPH (Special Interest Group on GRAPHics and Interactive Techniques) o filme de curta metragem *The Adventures of André and Wally B.*, o primeiro curta produzido pelo estúdio. Ele apresentava características bastante avançadas em termos de computação gráfica, como personagens complexos flexíveis e texturas pintadas a mão.

Em 1986, a Pixar desdobrou-se da Lucasfilm como uma empresa independente. Porém, apesar do desenvolvimento de novas técnicas de computação gráfica, a capacidade de processamento dos computadores da época ainda estava muito abaixo do necessário para criar filmes com a qualidade que desejavam. Nessa época a renderização de um simples frame em alto nível de detalhe em um computador VAX, por exemplo, poderia levar até duas semanas. Fazer um filme completo com 30 frames por segundo seria impraticável.

Porém, apesar de saberem que no curto prazo não haveria filme, a empresa contava com a Lei de Moore, que estabelecia que os computadores se tornariam cada vez mais poderosos com o passar dos anos. Seria apenas uma questão de tempo até surgirem computadores suficientemente rápidos para renderizar um desenho animado, talvez até um longa-metragem, com qualidade aceitável para ser exibido em cinemas. Neste meio-tempo, a empresa vendeu

hardware, especificamente o Pixar Image Computer, que foi adquirido por vários órgãos governamentais e organizações médicas.

Após ser demitido da Apple, Steve Jobs resolveu aportar recursos na Pixar como investidor. Pagou 5 milhões de dólares a George Lucas em troca de direitos de tecnologia e colocou mais 5 milhões diretamente no caixa da Pixar, tornando-se membro do conselho da empresa e seu diretor.

Ainda em 1986 é completado o curta *Luxo Jr.*, que foi apresentado na SIGGRAPH e em vários cinemas da região de Los Angeles antes dos filmes principais. *Luxo Jr.* foi o primeiro filme gerado por computador a ser indicado a um Oscar de animação de curta duração. Luxo Jr. é o pequeno abajur animado que até hoje aparece nas vinhetas de abertura dos filmes da Pixar, pulando sobre o “i” do logotipo da empresa até enterrá-lo e ficar no seu lugar.

Em 1987 a companhia lança *Sonho do Red (Red's Dream)*, com efeitos de chuva e iluminação complexa. Parte do filme foi renderizada pelo Pixar Image Computer. Em 1988 *Tin Toy* é completado, o primeiro filme computadorizado a receber um Oscar de melhor animação de curta duração.

Em 1989 é finalizado *Knick Knack*, o primeiro filme renderizado por computador em visão 3D estereoscópica. Nessa época, a empresa começa a produzir também comerciais usando computação gráfica.

Um dos grandes clientes do Pixar Image Computer eram os estúdios Disney, que estavam tentando migrar das técnicas totalmente manuais de animação para bases computacionais. Em 1991 houve uma primeira aproximação entre a Pixar e a Disney, quando as duas empresas anunciaram um acordo para produzir pelo menos um longa-metragem baseado em animação computadorizada. Assim, em 1995 é lançado *Toy Story*, o primeiro filme completo de longa duração a ser produzido com essa nova tecnologia. Ele foi lançado no final do ano, em novembro, e foi o

filme mais lucrativo de 1995, tendo faturado mais de 360 milhões de dólares, além de ter sido indicado a três Oscar.

Em 1997 a Pixar e a Disney fazem um novo acordo para produzir pelo menos cinco filmes de longa-metragem nos 10 anos seguintes. Neste ano, *O Jogo de Geri* (*Geri's Game*) recebe o segundo Oscar de animação de curta-metragem da empresa. Este curta foi lançado como apresentação prévia do longa *Vida de Inseto* (*A Bug's Life*) de 1998, seguido por outros sucessos como *Toy Story 2* (1999), *Coisas de Pássaros* (*For the Birds*, 2000), *Monstros S.A.* (*Monsters, Inc.*, 2001), *O Novo Carro do Mike* (*Mike's New Car*, 2002), *Procurando Nemo* (*Finding Nemo*, 2003), *Os Incríveis* (*The Incredibles*, 2004) e *A Banda de um Homem Só* (*One Man Band*, 2005).

A relação entre a Disney e a Pixar, entretanto, estava longe de ser tranquila. Houve muitos desacordos e desagrados entre as duas empresas, especialmente pelo lado da Pixar, que tinha todo o ônus da produção dos filmes e que ficava apenas com metade da renda da veiculação destes nos cinemas, enquanto que a Disney mantinha todos os direitos, inclusive o de produzir sequências para os filmes, sem autorização ou compensação para a Pixar.

Os problemas só foram resolvidos quando a Disney finalmente comprou a Pixar de uma vez por todas em 2006, pagando 7,4 bilhões de dólares em ações. Como Steve Jobs tinha 50,1% da Pixar, ele acabou se tornando também o maior acionista individual da Disney, com 7% da empresa e virou também membro de seu conselho. Roy Disney, irmão do falecido Walt e co-fundador da empresa, tinha apenas 1%.

Após a compra, a história de sucessos da Pixar não terminou. Vieram os lançamentos de *Carros* (*Cars*, 2006), *Ratatouille* (2007), *Wall-E* (2008), *Up – Altas Aventuras* (*Up*, 2009), *Toy Story 3* (2010), *Carros 2* (*Cars 2*, 2011), *Valente* (*Brave*, 2012), *Universidade Monstros* (*Monsters University*, 2013), *Divertidamente* (*Inside Out*, 2015) e o *Bom Dinossauro* (*The Good Dinosaur*, 2015), dentre

muitos outros. Atualmente está em produção *Finding Dory*, a ser lançado em junho de 2016.

## 9.47 Acorn Archimedes – 1987

O Acorn Archimedes de 1987 foi um computador pessoal, o primeiro baseado em microprocessador RISC a ser lançado com o circuito ARM (Acorn RISC Machine). Ele já foi lançado com quatro modelos disponíveis com 512 kB e 1 MB de memória. O mais completo vinha com inacreditáveis (para a época) 4 MB de memória principal e uma unidade de disco rígido de 20 MB. A atualização de um modelo para o outro era possível simplesmente pelo encaixe de circuitos extras na máquina. Na verdade, o controlador de memória era capaz de suportar 32 MB, mas tal quantidade de memória seria economicamente inviável e assim ele ficava limitado a 4 MB e usava o disco como memória virtual se necessário.

Esse computador, especialmente o modelo mais básico, foi projetado para substituir o BBC Micro, que então já estava defasado por ser um computador de apenas 8 bits, contra os 32 do Archimedes. Para manter compatibilidade com antigos sistemas, ele vinha com um software que emulava o 6502 do BBC Micro.

Em termos de resolução de tela, o Archimedes permitia 640x512 pontos com 16 cores ou 640x256 pontos com 256 cores. Em modo monocromático a resolução alcançava incríveis 1.024x1.024. Ele tinha circuitos de memória dedicados a vídeo, mas boa parte dessa capacidade vinha do processador RISC, extremamente poderoso. Um avaliador do computador mencionou que os sprites usados nos jogos se moviam pela tela tão suavemente quanto aqueles que eram gerados por hardware no Commodore Amiga, e o Archimedes fazia sprites com software.

O sistema operacional do Archimedes se chamava ARTHUR e consta que a sigla significava “A RISC Operating System before THURsday”.<sup>54</sup> Posteriormente, foi renomeado para “RiscOS”. Ele tinha um componente gráfico chamado Archimedes Desktop

Manager, escrito em BBC BASIC, e sua interface lembrava em certos aspectos a do Windows 2.0, mas em alguns pontos era superior a ele pois permitia, por exemplo, copiar vários arquivos de uma só vez para diferentes destinos.

Vários modelos do Archimedes foram produzidos durante mais de cinco anos até que o mercado começou a se voltar para os onipresentes PCs com Windows. A Acorn então resolveu modificá-lo para se tornar um computador ao estilo do PC, mas mantendo o processador RISC e sendo capaz de rodar o Microsoft Windows.

Mais tarde, como já vimos, a Acorn se desmembrou e vendeu seu negócio de processadores, criando a ARM. Hoje, embora ela ainda exista, não produz mais computadores.



Figura 9.29: Um Acorn Archimedes rodando Arthur 1.20.<sup>55</sup>

## 9.48 IBM PS/2 e OS/2 – 1987

Para entendermos a história do novo microcomputador e novo sistema operacional da IBM, respectivamente e não muito criativamente chamados de PS/2 e OS/2, em 1987 teremos que voltar um pouquinho na história.

Lá atrás, em 1980, Bill Gates se recusou a vender para a IBM uma licença de uso indeterminado para o MS-DOS por 80 mil dólares. O jovem dono de uma pequena empresa devia ter-se sentido acuado em frente a uma grande equipe de advogados com ternos caríssimos, mas ele insistiu: queria ganhar uma pequena quantia para cada cópia do MS-DOS que fosse instalada em uma máquina IBM. Mais ainda, desejava manter o direito de vender o MS-DOS para quaisquer outras empresas que eventualmente produzissem PCs.

A IBM, desesperada para tirar o atraso no mercado de computadores pessoais, no qual ela ainda sequer engatinhava enquanto a Apple e outras empresas nadavam de braçada, acabou concordando com os termos de Gates. Podemos dizer que isso, no final, foi o que acabou tirando a IBM do mercado de computadores pessoais no início do século XXI.

A IBM concordou porque acreditava que, mesmo pagando uma pequena taxa a Gates pelo uso do DOS, ainda teria muito lucro com a venda dos PCs. Além disso, nenhuma outra empresa fazia máquinas compatíveis com o PC na época, e o jovem empresário não teria outros para quem vender seu sistema. No entanto, o PC era uma máquina construída com circuitos de catálogo, como já vimos, que podiam ser comprados até pelo correio. Apenas a BIOS era propriedade intelectual da IBM, mas várias empresas conseguiram legalmente produzir do zero novas BIOS com a mesma funcionalidade da original sem infringir nenhum direito autoral.

A indústria de clones de PC cresceu pois não era necessário muito capital para começar a produzir tais máquinas. A IBM viu seu

mercado cair cada vez mais e os preços de seus próprios PCs também tiveram que baixar. Por outro lado, Gates vendia seu MS-DOS para as fabricantes de clones e lucrava tanto com a IBM quanto com eles. A IBM não suportou não ter controle sobre isso, e resolveu criar um novo sistema operacional, só dela, que iria diferenciar o seu PC dos clones. Como o MS-DOS na época ainda era um sistema operacional muito limitado, especialmente em relação à possibilidade de executar vários programas simultaneamente, o novo sistema seria um marco que permitiria à IBM retomar o mercado de PCs – ou pelo menos assim ela pensava.

O novo sistema se chamaria OS/2: “OS” para que os potenciais clientes pudessem associá-lo com o OS/360 do famoso mainframe dos anos 1960, e o “2” porque seria o segundo sistema operacional do PC, sucedendo ao MS-DOS.

Ao mesmo tempo, a IBM iria lançar uma máquina que sucederia o PC e que não seria mais tão fácil de clonar. Seria o PS/2, onde “PS” significa “Personal System” e “2” porque é o sucessor do PC, que foi o primeiro.

Entretanto, escrever um novo sistema operacional praticamente do zero era uma tarefa que tomaria tempo. Assim, a IBM decidiu que ela projetaria o sistema, mas o código seria em sua maior parte produzido pela Microsoft. Dessa vez, os direitos sobre o sistema pertenceriam completamente à IBM.

Porém, por que a Microsoft concordaria em escrever um software projetado para competir, e talvez até eliminar, a sua principal fonte de renda? Steve Ballmer explicou assim: *“It was what we used to call at the time ‘Riding the Bear’. You just had to try to stay on the bear’s back, and the bear would twist and turn and try to throw you off, but we were going to stay on the bear, because the bear was the biggest, the most important... you just had to be with the bear, otherwise you would be under the bear.”*<sup>56</sup>

Agora, qual seria o microprocessador da nova máquina? A melhor opção na linha x86 da Intel era o 80386, um verdadeiro processador de 32 bits capaz de gerenciar 4 GB de memória e que podia entrar em modo “Virtual 8086”, no qual várias aplicações de 8 bits, como o MS-DOS, poderiam rodar simultaneamente. Porém, isso soava para os executivos da IBM como algo muito parecido com um mainframe e, para que a empresa não criasse um produto que competisse com sua principal mercadoria, resolveram que o PS/2 usaria o já ultrapassado 80286, que só acessava 16 MB de memória. A Microsoft detestava esse processador porque ele só funcionava em plena capacidade se estivesse em “modo protegido”. A outra opção era o “modo real”, no qual ele não era muito mais poderoso do que um 8086. O MS-DOS, no entanto, só funcionava bem no modo real e para piorar tudo, o processador só passava de modo protegido para modo real se fosse desligado e reinicializado. Mais um motivo para a IBM escolher o 286: ele afastaria usuários de MS-DOS que teriam que optar pelo OS/2 ou perder poder de processamento.

Outro problema que a IBM precisava lidar era o surgimento das interfaces gráficas como as do Amiga, Atari-ST e mesmo o Windows. O novo sistema precisaria ter essa característica também. Finalmente, o desenvolvimento do OS/2 iniciou: tarefas de programação estavam sendo enviadas à Microsoft, mas a burocracia da IBM era tão grande em relação a segredo industrial que se conta que os programadores desejavam ter advogados presentes quando conversavam uns com os outros. Isso não foi o pior: mais grave nesse momento foi a decisão da IBM de pagar à Microsoft valores proporcionais ao número de linhas de código produzidas. Para um gerente que não entende nada de programação isso pode até parecer uma boa ideia, mas na indústria de computação ela é péssima! Ocorre que existem infinitas formas de se escrever qualquer rotina em linguagem de programação. Programadores usualmente usam a forma que lhes parece mais econômica, escrevendo o mínimo possível de linhas de código para fazer a tarefa. Porém, é possível e fácil ser prolixo com linguagem

de programação. Assim, um programador que recebe seu pagamento em função do número de linhas de código produzidas vai escrever uma rotina com 50 linhas ou com 500 linhas para fazer a mesma coisa? Ele pode ganhar 10 vezes mais resolvendo a mesma tarefa. O resultado desse tipo de política são programas que são muito maiores em termos de uso de memória; e memória na época era um insumo muito caro para ser desperdiçado.

O OS/2 1.0 foi lançado em 1987 e adivinhem? Ainda sem interface gráfica. Só a telinha preta e a linha de comando como o velho DOS. Ele tinha vantagens em relação ao DOS, mas os usuários ainda não conseguiam vê-las.

A IBM contava com a baixa no preço das memórias para disponibilizar máquinas relativamente baratas com muita memória que pudessem rodar adequadamente o OS/2. De fato, o preço do MB caiu de 880 dólares em 1985 para 135 dólares em 1987, e esperava-se que continuasse caindo. Porém, essa mesma queda aumentou muito a demanda e as fábricas não davam conta. Assim, em 1988 o preço do MB voltou a subir para a casa de 500 dólares e permaneceu ali por pelo menos dois anos.

Era necessário um mínimo de 4 MB para rodar o OS/2, mas como a maioria das máquinas ainda vinha com 1 MB ou menos, isso significava que um potencial cliente teria que comprar o OS/2 1.0 sem interface gráfica da IBM por 325 dólares e ainda gastar mais 1.500 dólares de memória para poder usá-lo. Bem, os clientes preferiram continuar com o MS-DOS, que podia não ser tão bom quanto, mas dava conta da maioria das necessidades do usuário médio.

O PS/2, quando lançado, tinha algumas inovações interessantes, como o fato de ter sido o primeiro microcomputador a usar os novos drives de 3½ polegadas e pioneiro em utilizar conectores redondos para ligar o teclado, o que acabou sendo um padrão nos anos seguintes. Mas lhe faltava performance. De fato, no momento do lançamento, havia várias versões do PS/2 disponíveis. Os modelos

25 e 30 eram de baixa performance e usavam o antigo processador 8086 a um clock baixo. Porém, estas máquinas eram mais como chamarizes para fazer os compradores se interessar pelos modelos 50 e 60, esses já com o chip 286. Os top de linha, a um preço absurdo, eram os modelos 70 e 80 com o processador 386.

Entretanto, mesmo essas máquinas com o 386 estavam ali mais pelo marketing do que qualquer outra coisa, porque o OS/2 foi propositalmente produzido para não usar todo o potencial desse processador. A IBM queria que os computadores pessoais voltassem a ser usados mais como terminais de seus mainframes do que como máquinas isoladas. O problema é que outras empresas não viam as coisas assim e apostavam na interconectividade entre microcomputadores e não na ligação deles com mainframes.

A indústria de clones, porém não deu mínima para isso e começaram a produzir computadores que só na aparência lembravam o OS/2, mas não tinham seus circuitos. Em algum momento conseguiram inclusive se juntar e criar seus próprios padrões, diferentes dos da IBM para não ter problemas com o departamento legal da empresa, mas todos compatíveis entre si. Isso isolou o PS/2 e seu sistema operacional ainda mais.

Quando Bill Gates percebeu que a indústria de clones estava toda migrando para máquinas com processadores 386, ele se deu conta de que o OS/2 não teria vida longa e resolveu “descer das costas do urso”. O problema é que teria que fazer isso com muito, muito cuidado.

No final de 1989 a Microsoft estava para lançar o Windows 3.0, que foi a primeira versão realmente útil e popular do sistema. O OS/2 já contava com uma versão gráfica nessa época, mas a IBM ainda via a Microsoft como parceira na área de sistemas operacionais e assim propôs à empresa um acordo no qual a IBM daria ampla propaganda e visibilidade ao novo Windows. Em troca, ela queria

que a Microsoft renunciasse ao direito de vender o DOS para outras empresas que não a IBM. Gates pensou no assunto e caiu fora.

A IBM viu isso como uma traição e proibiu todos os seus departamentos de desenvolver qualquer produto ou componente relacionado ao Windows. Porém, a Microsoft ainda tinha obrigações contratuais em relação ao desenvolvimento do OS/2. Como em um divórcio, as duas empresas resolveram dividir o OS/2 ao meio: a IBM tomaria para si a responsabilidade sobre a nova versão 2.0 ainda sendo desenvolvida, enquanto que a Microsoft ficaria responsável pela versão 3.0, apenas vagamente planejada, conhecida internamente como OS/2 NT (New Technology).

O OS/2 1.3 foi lançado em 1991 quando os preços das memórias finalmente estavam baixando novamente, e obteve sucesso moderado. Enquanto isso, o Windows 3.0 vendia como banana na feira. O Windows era parecido com o OS/2, embora não tão bom em algumas coisas, mas era bem mais barato. Além disso, a Microsoft começou nesta época a dar descontos enormes para as fabricantes de clones que vendessem máquinas já com o Windows instalado de fábrica.

Já em 1992 a Microsoft lançou o Windows 3.1 e, apenas um mês, depois a IBM lança o OS/2 2.0, uma versão que era vendida como “orientada a objetos”. A orientação a objetos, como sabemos, já existia desde a década de 1960, mas no início dos anos 1990 era a palavra da moda. Para alguma coisa ser boa no mundo da computação ela precisava ser “orientada a objetos”. Porém, o que era um sistema operacional orientado a objetos? Bem, os ícones, telas e outros elementos podiam ser tratados como coisas sobre as quais se podia operar e que tinham propriedades, conhecidas como “estado interno”. O OS/2 2.0 permitia, por exemplo, que um usuário arrastasse a cor de uma *palette* para dentro de uma janela e isso mudaria a cor da janela. Podia também alterar fontes e tamanho de letras, gerando uma confusão visual sem tamanho. No final, a avaliação dos usuários foi algo do tipo: “OK, o OS/2 faz coisas que outros sistemas não fazem, mas quem quer realmente isso?” O

problema era ainda maior quando usuários novatos faziam essas coisas sem querer e depois não conseguiam descobrir como voltar atrás. Pior de tudo: essas características sem muita utilidade consumiam muita memória.

O OS/2 2.0 exigia um mínimo de 4 MB para rodar, mas isso era mínimo demais. Depois de carregar pelo menos 21 (isso mesmo, 21) disquetes para instalar o sistema operacional, a memória já estava tão cheia que se você simplesmente clicasse em um ícone da tela com o botão direito do mouse, o computador já começaria a fazer swap, ou seja, usar memória de disco por causa da falta de memória principal. Um sistema operacional assim era inútil.

Com 8 ou 16 MB o sistema funcionava bem, mas o Windows 3.1 era bastante funcional com apenas 2 MB. De fato, o OS/2 realmente era melhor do que o Windows. Ele conseguia, por exemplo, rodar jogos em DOS em modo de tela cheia enquanto executava outras tarefas em background. O aplicativo que abria a telinha preta do DOS no OS/2 era tão bom que você podia rodar o Windows dentro dele. Você poderia inclusive utilizar vários Windows em diferentes janelas ao mesmo tempo, se tivesse memória para isso, e se um deles travasse por algum motivo, isso não afetaria os demais.

Isso parecia genial, mas para sobreviver um sistema operacional precisa de aplicativos. Os desenvolvedores viram como o OS/2 era bom em rodar aplicações Windows e não tiveram dúvida: passaram a escrever aplicativos apenas para o Windows, pois dessa forma eles rodariam tanto num sistema quanto no outro. Pouquíssimas empresas desenvolveram aplicativos diretamente para o OS/2. Companhias grandes como a Lotus até se aventuraram a isso, mas as pequenas que tentaram faliram em pouco tempo porque não tinham mercado para vendê-los. A IBM chegou a usar como propaganda a frase: “Um DOS melhor que o DOS, um Windows melhor que o Windows.” De fato, era melhor mesmo, mas num sentido que foi negativo para a IBM.

A IBM chegou a contratar a Borland para desenvolver aplicativos para o OS/2, mas o contrato não especificava que eles precisavam ser eficientes ou livres de erros. Assim, a Borland, enviava os produtos o mais rápido possível e, todos sabemos quem é a inimiga da pressa...

As vendas do OS/2 não foram tão ruins. Mais de 1 milhão de cópias chegaram a ser vendidas, apesar dos erros de marketing da IBM na época, como usar atores do seriado *M\*A\*S\*H* dos anos 1970 como garotos propaganda do PS/2; afinal eram médicos de campanha na Guerra do Vietnã, então, qual a conexão com computadores?

O fim do OS/2 veio com o lançamento de um novo sistema operacional pela Microsoft. A empresa ainda trabalhava no projeto do OS/2 3.0 quando Gates contratou o arquiteto do sistema VMS usado em minicomputadores de escala industrial, e o colocou como seu responsável. A primeira coisa que ele fez foi jogar fora todo o código produzido até então e reiniciar o projeto com novas bases: seria um sistema de alta performance, tolerante a falhas, independente de plataforma e altamente preparado para operar com redes de computadores. O nome seria Windows NT.

A IBM, ciente deste movimento, contra-atacou com o lançamento antes do prazo de uma nova versão do OS/2, a OS/2 Warp. Queriam passar a ideia de um sistema muito rápido como as naves estelares de *Jornada nas Estrelas*, que viajam acima da velocidade da luz usando o conceito de “warp” ou “dobra espacial”. Porém, esqueceram de pedir permissão para a Paramount para usar essa referência e, quando o fizeram, a produtora não a concedeu. Assim, eles poderiam usar a palavra, mas não a referência à velocidade da luz. O problema é que o nome do novo sistema já era conhecido no mercado, e *warp* em inglês literalmente significa uma dobragem, ou seja, algo torto ou amassado, e isso não seria bom para a sua imagem.

Ainda assim, a IBM chegou a convidar o ator Patrick Stewart, que encarnou o capitão Jean Luck Picard, para ser o garoto propaganda do produto, criando assim, uma associação indireta e implícita.

Steward, no entanto, não aceitou e a IBM teve que se contentar com a atriz Kate Mulgrew, que incorporou a capitã Kathryn Janeway da nave *Voyager*, uma sequência da série *Jornada nas Estrelas*. Só que, se a ideia era criar algum tipo de associação com o sistema operacional, novamente houve um ato falho, pois enquanto a *Enterprise* de Picard avançava por onde ninguém jamais esteve, buscando novos mundos e novas civilizações, a história de Janeway é a de uma nave que se perdeu do outro lado da galáxia e buscava desesperadamente voltar para casa, coisa que não conseguia com recursos próprios – ou seja, estava mais para *Perdidos no Espaço* do que para *Jornada nas Estrelas*.

Em 1995, a Microsoft lançava outra pá de cal no OS/2. O novo Windows 95 era muito mais estável do que o OS/2, e, portanto, mais confiável para uso comercial e industrial. Apesar da fama que o Windows tinha de travar e mostrar a odiada tela azul, o OS/2 Warp era ainda pior.

O problema era causado por uma falha de design em um componente chamado SIQ (Synchronous Input Queue). Pode-se até fazer apocrifamente uma relação com a palavra “sick”, que em inglês significa “doente”. O SIQ era um componente que funcionava como uma autoestrada de várias pistas, mas com uma única cabine de pedágio. Todos os elementos da interface gráfica do OS/2 passavam por ali. Se um deles pifasse na cabine, ninguém mais passava. Os processos até continuariam rodando em background, mas seria inútil pois o usuário não conseguiria interagir ou visualizar nenhum deles. Ironicamente, se fossem rodadas apenas aplicações não nativas ao OS/2, o sistema ficava bem mais estável porque neste caso não usavam o SIQ.

Quando a IBM capitulou e pediu à Microsoft que fornecesse licenças do Windows 95 para seus PS/2, a Microsoft exigiu que a IBM parasse de desenvolver o OS/2. Ela nem precisava pedir. Após o lançamento do OS/2 Warp 4 em 1996, a IBM solicitou a um grupo de especialistas que avaliasse o potencial de mercado do produto. A avaliação não foi boa e, assim, o OS/2 acabou. De um dia para o

outro, 1.300 funcionários da IBM alocados a esse projeto perderam seus empregos.

A IBM continuou vendendo o OS/2 até 2001, mas ninguém comprava além de alguns clientes, como bancos, que usavam o sistema em caixas automáticos. A [Figura 9.30](#) mostra um caixa automático na Austrália com OS/2, com defeito, dando *boot*. Em breve, o Windows NT acabou devorando esse mercado também. O OS/2 foi vendido para a Serenity Systems e rebatizado como “eComStation”. Continua, consistentemente, não vendendo nada.



Figura 9.30: OS/2 rodando em um caixa automático.<sup>57</sup>

Finalmente em 2004 a IBM vende todo o seu setor de computadores pessoais para a chinesa Lenovo, deixando assim o mercado após 23 anos. Ela reinou até o momento em que tentou tirar a Microsoft do mercado – mas perdeu essa luta.

## 9.49 Perl – 1987

Perl é uma linguagem de programação criada em 1987 por Larry Wall (Estados Unidos, 1954), que não estava satisfeito com as ferramentas para desenvolver sistemas que possuía. Segundo ele, elas até permitiam fazer o trabalho, mas era mais difícil do que realmente precisava.

Wall dizia que o segredo do progresso na computação está na preguiça e na arrogância: preguiça de fazer as coisas do jeito mais difícil e arrogância de acreditar que vai conseguir construir um jeito melhor de fazer. Porém, desde o início sabia que não estava fazendo uma nova linguagem apenas para si. Assim, tratou de envolver uma comunidade nisso. Talvez por isso o logotipo de Perl seja um camelo – que, segundo a piada, é um cavalo projetado por um comitê. De fato, o logotipo sequer é um camelo mesmo, mas um dromedário. O verdadeiro motivo dessa associação foi o livro de programação Perl publicado por Wall, que tinha um dromedário na capa e até hoje é conhecido como o “livro do camelo”. A O'Reilly, editora que o publicou, tinha uma tradição de colocar imagens de animais nas capas dos livros. Vários outros foram publicados com diferentes animais, sem nenhum motivo aparente exceto pelo fato de que as capas com animais são uma identificação clara dos livros da editora.

Para o nome da nova linguagem, Wall queria que fosse curto e com conotações positivas. Ele disse que jamais batizaria linguagens com nomes como Scheme ou Python. Depois de olhar no dicionário, segundo ele, todas as palavras com três ou quatro letras e rejeitá-las, pensou em dar à linguagem o nome de sua esposa Gloria, mas achou que isso podia complicar as coisas na esfera doméstica e acabou pensando em PEARL, para o qual ele inventou a sigla “Practical Extraction And Report Language”.<sup>58</sup> Depois de ouvir falar em uma obscura linguagem gráfica que teria o mesmo nome, ele mudou a grafia para PERL, mantendo a mesma pronúncia, mas dando à sigla um significado alternativo: “Pathologically Eclectic Rubbish Lister”.<sup>59</sup>

O nome, porém, nunca foi usado em maiúsculas como se fosse uma sigla, mas escrito sempre como “Perl”. Em algum momento a linguagem “Perl” ainda foi distinguida do programa “perl” que a implementava, este escrito só em minúsculas. Isso provavelmente deveu-se à cultura Unix, que, ao contrário da cultura IBM, usava minúsculas em vez de maiúsculas para escrever programas e comandos.

Perl acabou se tornando uma família de linguagens de alto nível. São linguagens de propósito geral e interpretadas. Ela foi criada como uma linguagem de script para o sistema operacional Unix, ou seja, uma linguagem que permitia escrever programas usando os comandos do próprio sistema operacional. O objetivo inicial da linguagem era facilitar o processo de geração de relatórios.

Perl foi inspirada principalmente pela linguagem C, mas também por shell script (sh), a linguagem de extração de dados da Bell Labs chamada AWK, e o stream editor (sed), que é uma ferramenta de processamento de textos do Unix.

Entre vários motivos, Wall criticava as linguagens e sistemas da época por serem orientados ao computador e não ao programador. Por exemplo, por que um programador precisava declarar um campo como “nome” ou “endereço” como VARCHAR(255), indicando que ele teria capacidade para conter 255 caracteres? Por que, no caso de uma chave primária de banco de dados, o comprimento não podia passar de 60? “Quem diabos inventava esses números?”, pensava ele. Assim, Perl é uma linguagem de programação muito mais flexível. Seu lema é: “Existem vários meios para fazer a coisa.”

Em uma entrevista, Wall foi perguntado sobre quem usava Perl. Ele disse que assistiu a uma palestra de um funcionário da NSA (National Security Agency), que mencionou usar Perl. Ele abordou o palestrante após a palestra e, sem se identificar, perguntou se ele o autorizaria a espalhar por aí que o pessoal da NSA usava a linguagem. O palestrante então retrucou: “E quem não usa??” Wall

explicou que, a partir de então, ele passou não a dizer que a NSA usava Perl, mas que a NSA acreditava que todo o mundo usava Perl.

Entre outras coisas, a linguagem foi usada para salvar o projeto Genoma porque diferentes laboratórios ao redor do mundo usavam diferentes sistemas de banco de dados não muito compatíveis entre si. Assim, Perl foi usada para fazer o meio de campo entre esses sistemas, integrando toda a base de dados. Por essa e outras razões, ela às vezes é apelidada de “fita adesiva”; em 1998, comentava-se que Perl era a cola que evitava que a Internet se esfacelasse em pedaços.

Em 2001 uma revisão de Perl 5 acabou se desenvolvendo como linguagem independente. Assim, hoje a família conta com duas linguagens, Perl 5 e Perl 6.

## 9.50 VGA – 1987

O IBM PS/2 pode ter tido uma vida relativamente curta, mas deixou pelo menos um importante legado para os dias de hoje, que foi o padrão de vídeo criado pela IBM especialmente para ele: o VGA ou Video Graphics Array.

Inúmeras placas de vídeo foram construídas para os PCs ao longo dos anos, mas a IBM usava via de regra as placas CGA nas primeiras máquinas, depois substituídas por um padrão de vida curta e um pouco melhor, denominado EGA ou Enhanced Graphics Adapter.

O VGA, lançado em 1987, foi o último destes padrões a ser desenvolvido pela IBM. Tratava-se de um adaptador de vídeo com capacidade de 640x480 pontos em até 16 cores ou 320x200 em 256 cores. Não chegava a rivalizar com o Amiga ou o Atari ST, mas era o que se tinha de melhor para o PC.

A placa VGA manteve compatibilidade com os modelos CGA e EGA de forma que software desenvolvido para estas continuasse funcionando com VGA.

O padrão VGA incorporou em um único chip o que as placas anteriores incorporavam em vários chips. Assim, em vez de usar um adaptador baseado em uma placa independente que precisava ser acoplada a um slot livre do PC, o VGA podia ser simplesmente colocado sobre a placa-mãe, pois ocuparia muito pouco lugar se comparado com os antecessores. E assim a IBM fez com seus PS/2.

Com o passar do tempo outras empresas, especialmente fabricantes de clones, começaram a produzir adaptadores de vídeo compatíveis com VGA, mas com resoluções ainda maiores. A maioria desses novos adaptadores passou a chamar-se SVGA ou Super VGA. Infelizmente, esse era apenas um nome genérico compartilhado por vários tipos de sistemas não necessariamente compatíveis entre si, embora compatíveis com VGA. As resoluções mais típicas eram de 640x480, 800x600 e 1.024x768. O número de cores podia variar, conforme a resolução e a quantidade de memória de vídeo disponível entre 256, e 16 milhões.

Adaptadores mais modernos se autodenominam UVGA ou Ultra VGA e têm resoluções mais altas ainda. Infelizmente, também neste caso não há compatibilidade. Parte do problema é resolvido pelo software do sistema operacional que permite configurar a resolução, número de cores e frequência do adaptador de vídeo. Porém, sempre é necessário saber se os sinais de vídeo gerados pelo computador são ou não compatíveis com o dispositivo de vídeo. Talvez por isso, frequentemente notebooks modernos não consigam reproduzir seus vídeos nos projetores multimídia.

A [Figura 9.31](#) mostra a evolução atual da resolução de tela dos diferentes padrões de vídeo, começando com CGA no canto superior esquerdo até QSXGA, ou Quad Super XGA, que com 2.560x2.048 pontos é um formato moderno predominantemente usado em equipamentos médicos.

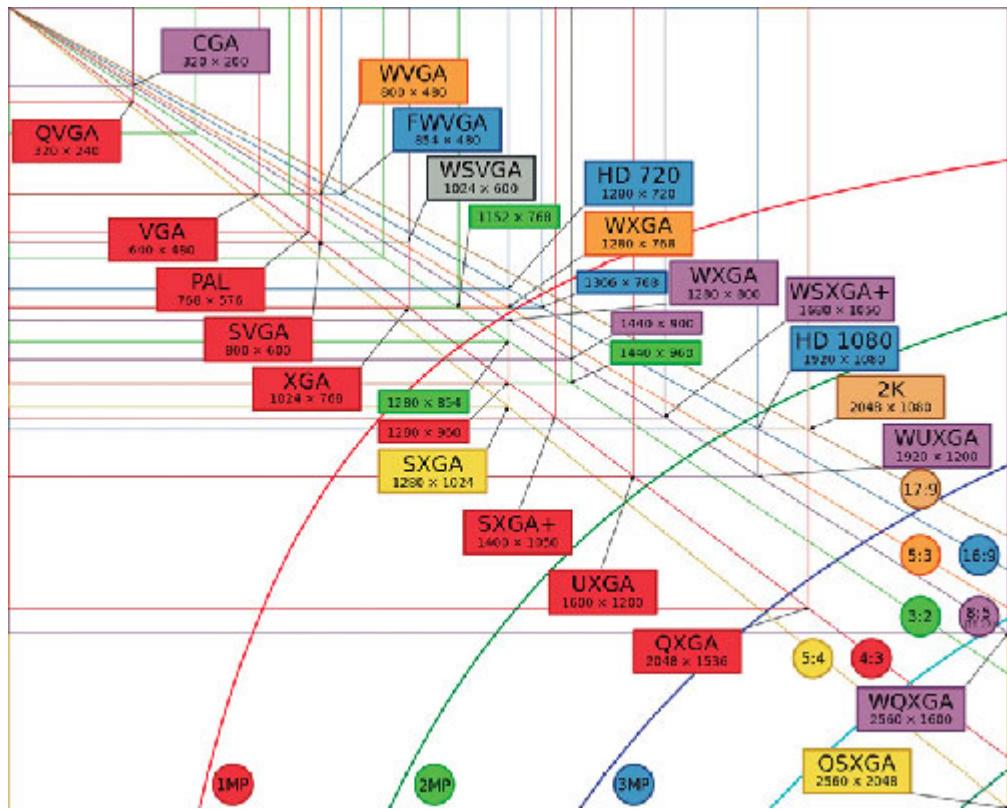


Figura 9.31: Evolução da resolução dos diferentes padrões de vídeo.<sup>60</sup>

Dois grandes erros sobre VGA que se cometia era: achar que o “A” era de *adapter*, como no caso de CGA e EGA, quando na verdade era de “array”; e achar que a resolução VGA era de 640x480 com 256 cores, quando na verdade reproduzia apenas 16 cores nessa resolução.

## 9.51 Photoshop – 1987

Em 1987 um estudante de doutorado chamado Thomas Knoll (Estados Unidos, 1960) escreveu o Display, um programa para fazer com que monitores monocromáticos exibissem imagens em tons de cinza. Thomas era filho do professor Glenn Knoll, entusiasta por fotografia e informática. Com o Apple II do pai, Thomas travou seus primeiros contatos com a computação. Em 1987 ele comprou um dos primeiros Macintosh a serem produzidos, na esperança de

poder usá-lo também como processador de imagens, mas, apesar de o Mac ter uma tela gráfica, os pixels eram pretos e brancos, não havendo tons intermediários, o que muito frustrou Thomas. Assim, após programar várias rotinas que simulavam o efeito de tons se cinza no Mac, ele construiu o sistema Display.

Thomas mostrou o sistema a seu irmão John, que trabalhava na Industrial Light & Magic, empresa responsável pelos efeitos especiais computadorizados para a Lucasfilm. John, que mais recentemente chegou a trabalhar nos efeitos de filmes de *Jornada nas Estrelas* e *Piratas do Caribe*, ficou entusiasmado pelo programa de Thomas e os dois passaram a colaborar para melhorá-lo.

No final do ano, John comprou o novo Macintosh em cores por 10 mil dólares e Thomas reescreveu seu Display para funcionar em modo colorido. Nessa época foram adicionadas algumas das ferramentas mais conhecidas, como a correção de cores, corte suave da imagem e filtros.

Desempregado, Thomas quase abandonou o projeto em 1988 por falta de dinheiro, mas conseguiu finalizar uma versão suficientemente apresentável para que John a levasse ao Vale do Silício e tentasse viabilizar uma exploração comercial.

Porém, o nome “Display” não era vendável. Eles pensaram em várias alternativas até que alguém em uma das empresas que John visitava, ao saber da dificuldade, sugeriu “PhotoShop”. A sugestão foi aceita, mas mais tarde o “S” maiúsculo foi substituído por minúsculo.

John deixou cópias do Photoshop em várias companhias, dentre elas a Adobe, que no final foi a grande parceira para a sua produção, mas não no início.

A primeira versão comercial do programa saiu como um aplicativo incluído na compra de Scanners da fabricante BarneyScan. Cerca de 200 cópias do Photoshop versão 0.87 apresentadas como BarneyScan XP foram distribuídas.

Ele também causou sensação na Apple, mas ali houve um primeiro incidente de pirataria. Conta-se que os engenheiros da empresa distribuíram cópias do Photoshop para seus amigos... um monte de amigos.

A Aldus, criadora do PageMaker, também foi procurada, mas não demonstrou interesse. Depois que o Photoshop virou uma febre, eles se arrependeram amargamente.

Em setembro de 1988 John retornou à Adobe e a empresa estava maravilhada com o programa. Eles até já tinham contrato assinado com outro fornecedor, mas perceberam que o Photoshop era muito superior. A Adobe comprou a licença para distribuir o produto com um aperto de mão. Apenas em abril de 1989 a papelada foi finalizada. A Adobe não seria dona do produto até anos depois do Photoshop se tornar um sucesso. De início, ela apenas tinha licença para distribuí-lo.

O Adobe Photoshop foi lançado comercialmente apenas em 1990. Ele era fornecido em um único disquete de 3½ polegadas e fez sucesso porque com ele, a um baixo custo, o dono de um Mac passava a ter uma ferramenta de processamento de imagens equivalente às de equipamentos muitas vezes mais caros. A [Figura 9.32](#) mostra sua tela de abertura no Mac.

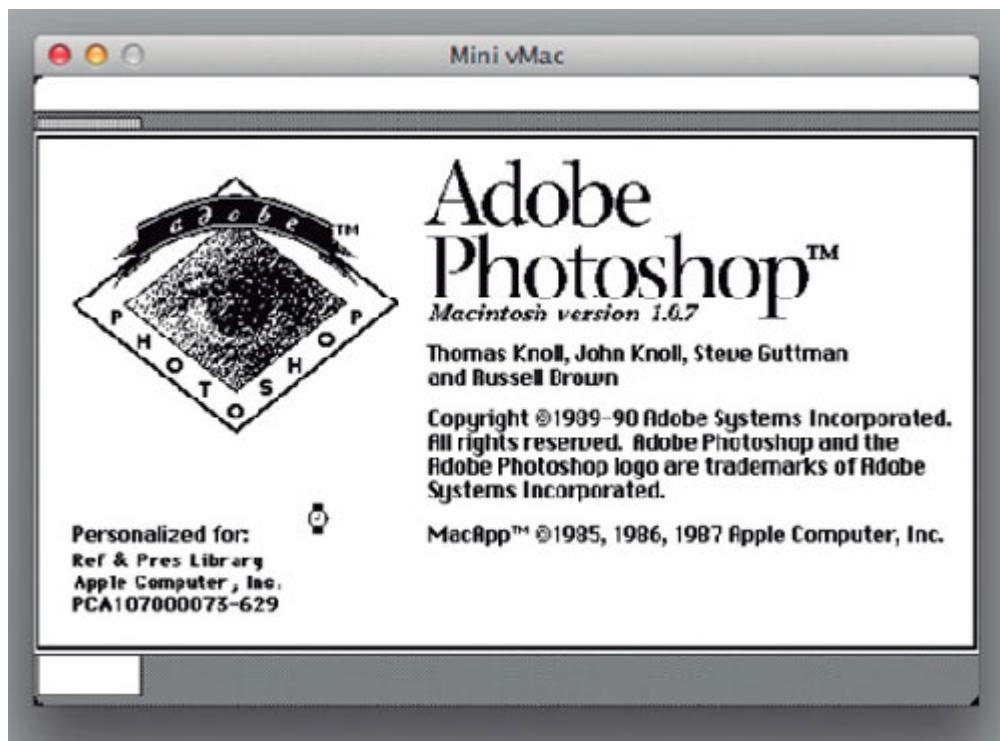


Figura 9.32: Tela de abertura do Photoshop 1.0.7.<sup>61</sup>

Em 1993 o Photoshop foi finalmente portado para o Windows e logo depois para os sistemas operacionais IRIX e Solaris.

O impacto do Photoshop na indústria gráfica foi tão grande que hoje qualquer pessoa associa a palavra ao ato de melhorar imagens, especialmente fotos, de forma a remover imperfeições ou até criar ilusões de coisas que não existem. Vários sites apresentam modelos fotográficas “antes e depois do Photoshop”. Em inglês, o termo chega a ser usado como um novo verbo: “to photoshop an image”<sup>62</sup> é uma frase comum nos estúdios gráficos. Há também sites dedicados a mostrar os desastres do Photoshop – não da ferramenta em si, mas das imagens que foram modificadas e que, por descuido dos designers, geraram efeitos impossíveis, como umbigos desaparecendo da barriga de modelos ou membros em posições anatomicamente impossíveis.

## 9.52 Hypercard – 1987

O Hypercard é considerado por muitos como o elo perdido na evolução da World Wide Web. Tratava-se de um aplicativo desenvolvido inicialmente para o Macintosh que permitia ao usuário criar pilhas de “cartões”, cada qual podia conter, texto, imagens, campos alimentados por informações de um banco de dados, botões e... hiperlinks.

O Hypercard foi usado naqueles tempos para construção dos primeiros sistemas multimídia e hipermídia, mas os sistemas gerados por ele eram para uso individual em um único computador. O Hypercard logo tornou-se obsoleto pelo surgimento da Web, que basicamente implementava as mesmas ideias; com a Web os sistemas hipermídia passavam a estar disponíveis na Internet, coisa que o Hypercard não fazia.

O aplicativo foi criado por Bill Atkinson (Estados Unidos, 1951) a partir de 1985, e originalmente se chamava WildCard. O Hypercard, distribuído gratuitamente com os computadores Macintosh, foi responsável por uma mudança de paradigma. Muitas pessoas que por sua formação não seriam capazes de desenvolver sistemas usando as linguagens de programação da época, rapidamente se tornavam capazes de desenvolver sistemas hipermídia usando a interface simples e intuitiva do Hypercard.

Em 2002, com a morte do aplicativo já sacramentada, Atkinson lamentou não ter percebido na época que poderia ter criado hiperlinks entre arquivos localizados em diferentes computadores da rede. Ele ficou muito preso à cultura de computação dentro da caixa da Apple e não se deu conta de que a computação em rede estava surgindo. Atkinson comenta que, se tivesse se dado conta disso, poderia ter inventado a World Wide Web já em 1987, dois anos antes de seu surgimento.

## 9.53 Worm – 1988

Em 1988, um estudante de doutorado da Universidade Cornell chamado Robert Tappan Morris (Estados Unidos, 1965) criou um

programa que alegadamente não tinha intenções maléficas, mas que causou um desastre em muitos computadores ligados à Internet. O programa entrou para a história como “The Worm”.<sup>63</sup> Morris disse que queria conhecer o verdadeiro tamanho da Internet e, assim, escreveu um programa que se alastrava de um computador para outro, instalando cópias de si mesmo e ativando-as para que pudessem passar para outros computadores.

O problema é que Morris esqueceu, ou pensou que não seria necessário, colocar um comando para que, caso o computador já estivesse infectado, ele não fosse infectado novamente. Sem essa salvaguarda o mesmo computador poderia ser infectado por um número indefinido de Worms, e cada um dos quais disputaria tempo de processamento com outros processos da máquina até que o processador estaria ocupado com tantos Worms que simplesmente travaria e não conseguiria executar mais nada. Cerca de 34 minutos após a meia-noite, Andy Suddith de Harvard teria postado a mensagem: “Pode haver um vírus solto pela Internet.” Depois disso, computadores tiveram que ser desligados e conexões cortadas para tentar evitar uma maior infecção. A Internet, criada para ser robusta, caiu aos pedaços por causa de um programa de 99 linhas escrito por um estudante de 23 anos.

Hoje em dia, o Worm até é considerado por muitos como um vírus, mas de um tipo especial. Enquanto a maioria dos vírus precisam ser carregados de um lado para outro pelos usuários e de alguma forma inadvertidamente ativado por eles (por exemplo, ao usar um disquete ou pendrive infectado, ou ao abrir um anexo suspeito em um e-mail), os worms são muito mais agressivos. Eles mesmos encontram seu caminho de uma máquina para outra e ativam suas cópias sem precisar da intervenção dos usuários. Assim, os worms se alastram muito mais rapidamente. Além disso, um vírus precisa se instalar dentro de outro programa, mas um worm tem vida própria – exatamente como na analogia com suas versões biológicas.

O termo “worm” possivelmente surgiu da obra de ficção científica *The Shockwave Rider*,<sup>64</sup> escrita por John Brunner em 1975. No livro, o mundo é dominado por um regime totalitário que baseia seu poder em uma grande rede de computadores. Um defensor da liberdade infesta essa rede com um programa chamado “tapeworm”<sup>65</sup>, que se espalha e derruba a rede, acabando assim com o poder desse governo.

Worm atacou apenas máquinas VAX e Sun. Não existem números exatos, mas estima-se que cerca de 6 mil máquinas, cerca de 10% das que estavam ligadas à Internet, tenham sido paralisadas pelo Worm de Morris, causando prejuízos que podem ter chegado à casa dos 10 milhões de dólares.

O Worm de Morris ajudou a mostrar como as redes eram frágeis em termos de segurança. Se ele realmente estivesse mal-intencionado, os estragos poderiam ter sido muito maiores. Mesmo assim, ele foi julgado e considerado culpado por fraude e abuso envolvendo computadores. Após vários recursos, foi condenado a três anos em liberdade condicional, 400 horas de serviço comunitário e uma multa de 10.050 dólares, sem contar os custos de sua supervisão. Hoje Morris é professor permanente no MIT.

## 9.54 IRC – 1988

IRC ou Internet Relay Chat foi o primeiro serviço mundial de bate-papo. Ele foi criado por Jarkko Oikarinen (Finlândia, 1967), estudante de graduação que trabalhava na Universidade de Oulu no verão de 1988. Ele comenta que não tinha muita coisa para fazer, pois sua atribuição era gerenciar um servidor Sun, mas isso não lhe tomava todo o tempo disponível. Assim, ele resolveu ocupar suas horas livres escrevendo um programa de comunicações para melhorar a usabilidade do OuluBox, um BBS de acesso público administrado por ele. O objetivo era permitir discussões não só na forma de boletins de notícias, mas também discussões online.

Para desenvolver o IRC, Oikarinen se inspirou no Bitnet Relay Chat e no rmsg, escrito por Jyrki Kuopala. O IRC foi desenvolvido sobre um programa de mensagens que não funcionava muito bem no OuluBox, chamado MUT ou MultiUser Talk, este, por sua vez, inspirado no programa talk do Unix.

Depois de receber uns 10 usuários no IRC, ele convidou através do Bitnet Relay conhecidos seus nas universidades de Tampere e Helsinki para instalarem servidores IRC em seus computadores. Outras universidades logo se ligaram à rede.

No final, IRC virou um programa com vida própria e as conexões com BBS foram esquecidas. Eles acharam que não seria uma boa ideia misturar as coisas: um programa para cada coisa era melhor. Porém, a noção de mensagens particulares entre duas pessoas foi logo ofuscada pela existência dos “canais”, que permitiam comunicação online em grupo.

IRC foi se espalhando inicialmente pela FUNET, rede da Finlândia, ainda isolada de outros países e depois pela NORDUNET, que integrava os países escandinavos. O primeiro servidor fora da Escandinávia foi instalado no MIT a pedido de Oikarinen. Em 1991, o uso do IRC ao redor do mundo cresceu vertiginosamente. Pessoas se ligavam aos canais para saber notícias sobre a Guerra do Golfo. Era uma forma de burlar a imprensa oficial que veiculava apenas notícias pré-selecionadas.

De 2003 para cá o uso do IRC vem diminuindo gradualmente, em grande parte substituído por outros produtos, como WhatsApp.

## 9.55 Archie – 1989

Quem conhece os famosos web crawlers, ou buscadores, como o mecanismo que tornou a Google a gigantesca potência que é hoje, precisa saber que tudo começou muito pequeno. Antes do Altavista, Yahoo! e outros buscadores – mesmo antes da World Wide Web –, já havia um buscador.

O programa considerado o primeiro buscador da história chamava-se “Archie” e foi criado por Alan Emtage (Barbados, 1964), estudante de computação da Universidade McGill no Canadá. Ele iniciou ali seus estudos em 1983, obtendo o Bacharelado e Mestrado. Em 1986, começou a trabalhar como administrador da infraestrutura de informática da universidade. Na época, apenas duas universidades canadenses tinham conexão com a Internet e a McGill era uma delas.

A universidade tinha muitos recursos para comprar hardware, mas praticamente nada para comprar software. Assim, um dos trabalhos de Emtage era buscar na rede por programas gratuitos que pudessem ser usados na universidade.

Ainda não havia uma Web nem Web browsers, mas programas podiam ser baixados através da internet por um serviço denominado FTP, ou File Transfer Protocol<sup>66</sup>, ainda hoje usado. Os arquivos ficavam em diferentes servidores, cada qual com vários diretórios e cada diretório com vários arquivos.

No início, Emtage entrava nos servidores manualmente, olhando diretório por diretório e arquivo por arquivo, preenchendo uma tabela com os nomes e endereços daqueles que pareciam ser úteis.

Então ele ficou cansado disso e escreveu um conjunto de scripts (sequências de comandos do sistema operacional), que fariam automaticamente todo o trabalho de visitar os servidores e coletar dados sobre os arquivos ali armazenados. Segundo o próprio Emtage conta, enquanto esses scripts rastejavam (*crawl*) pela rede, ele podia dormir. Mais uma vez a preguiça propulsiona o progresso, pois assim nasceu o primeiro Internet crawler.

Mais tarde ele liberou seu sistema para uso público e no ano seguinte foi responsável por mais de 50% do tráfego de internet do Canadá.

O nome escolhido para o programa foi “Archie” pois, segundo Emtage, ele simplesmente removeu um “v” da palavra “archive” e achou o resultado interessante. Havia uma revista em quadrinhos na

época muito famosa, a *Archie Comics*, na qual Archie era o personagem principal. Muita gente acreditava que o nome do sistema era uma homenagem ao personagem. Emtage negou, dizendo que odiava esses quadrinhos: uma vez teria tentado ler a revista e achara a coisa mais insípida que já tinha visto. Mesmo assim, outros crawlers da época acabaram sendo batizados com os nomes de personagens da mesma HQ, como Veronica e Jughead (chamado de “Moleza”, no Brasil).

## 9.56 Poqet PC – 1989

O Poqet PC, lançado em 1989, foi um computador do tipo subnotebook, ou seja, significativamente menor do que um notebook, feito para caber em um bolso. Suas dimensões eram 22 x 11 x 2,5 cm.



Figura 9.33: Poqet PC.<sup>67</sup>

Ele era alimentado por duas pilhas AA e, graças a um projeto intensivo de economia de energia (possivelmente um dos primeiros na história), elas podiam durar facilmente duas semanas ou mais. Muitas das técnicas de economia de energia do Poqet PC são usadas ainda hoje em notebooks e smartphones.

A pequena máquina era dobrável como uma ostra – ou seja, a tela sobre teclado –, como um notebook moderno. A tela era de cristal líquido monocromático, sem iluminação própria. Apesar de pequena, tinha a mesma resolução de uma placa CGA: 640x200 pontos ou

80x25 caracteres em modo texto. Usava um microprocessador 80C88 e tinha 640 kB de ROM e de 64 até 512 kB de RAM.

O preço também era atrativo: apenas 2 mil dólares. Não era top de linha em termos de desempenho, mas era muito prático em função de seu tamanho. Ele tinha BASIC residente e rodava a maioria dos aplicativos para PC que funcionassem com essa configuração de memória.

O Poqet PC foi muito bem recebido e seus criadores chegaram a ganhar dois prêmios: o prêmio de excelência técnica em portáteis da *PC Magazine* e o prêmio de distinção da *Byte*, ambos em 1989. A *PC Magazine* chegou a afirmar na época que: "If this Microsoft Windows thing doesn't pan out, the Poqet will rule the world."<sup>68</sup>

Porém, por algum motivo a Poqet Computer Corporation acabou sendo vendida para a Fujitsu, que pouco depois perdeu o interesse no produto e o descontinuou.

## 9.57 World Wide Web – 1989

Em 1989, Tim Berners-Lee (Reino Unido, 1955) trabalhava como engenheiro de software no grande laboratório CERN (Organização Europeia de Pesquisa Nuclear) em Genebra, onde ficava o maior acelerador de partículas do mundo na época. Lá, ele observava cientistas do planeta inteiro trabalhar, mas tendo enormes dificuldades em trocar informações, especialmente em projetos grandes. Arquivos em formatos, computadores e mídias diferentes, conexões incompatíveis ou simplesmente em línguas distintas faziam com que, no final, em vez de tentar obter as informações em arquivos, as pessoas resolvessem perguntar umas às outras na hora do café.

Em março, ele concebeu e apresentou a seu chefe um documento que tinha escrito "Information Management: A proposal".<sup>69</sup> Esse documento simplesmente iniciou uma nova era para a Humanidade, pois mesmo que Berners-Lee não tivesse noção disso na época, era a proposta de criação da World Wide Web (WWW). O mais incrível

de tudo foi que o seu chefe, Mike Sendall (Reino Unido, 1939-1999), não aceitou a proposta de início. Ele a achou vaga, apesar de excitante, e realmente escreveu essas palavras na capa. Porém, depois ele acabou se tornando um dos grandes entusiastas da ideia e a apoiou bastante.

Na proposta, Tim estuda os problemas de perda de informação enfrentados pelo CERN. Ele analisa duas possibilidades: organizar a informação em árvores de diretórios (mas argumenta que o mundo real não se organiza dessa forma), ou organizar a informação por palavras-chave (mas o problema é que diferentes pessoas nem sempre escolhem as mesmas palavras-chave). Finalmente, concluiu que a melhor forma de organizar a informação no CERN era através de um hipertexto, um conceito que já existia e com o qual Tim já tinha trabalhado. Porém, esse hipertexto seria acessível pela rede através de ferramentas universais.

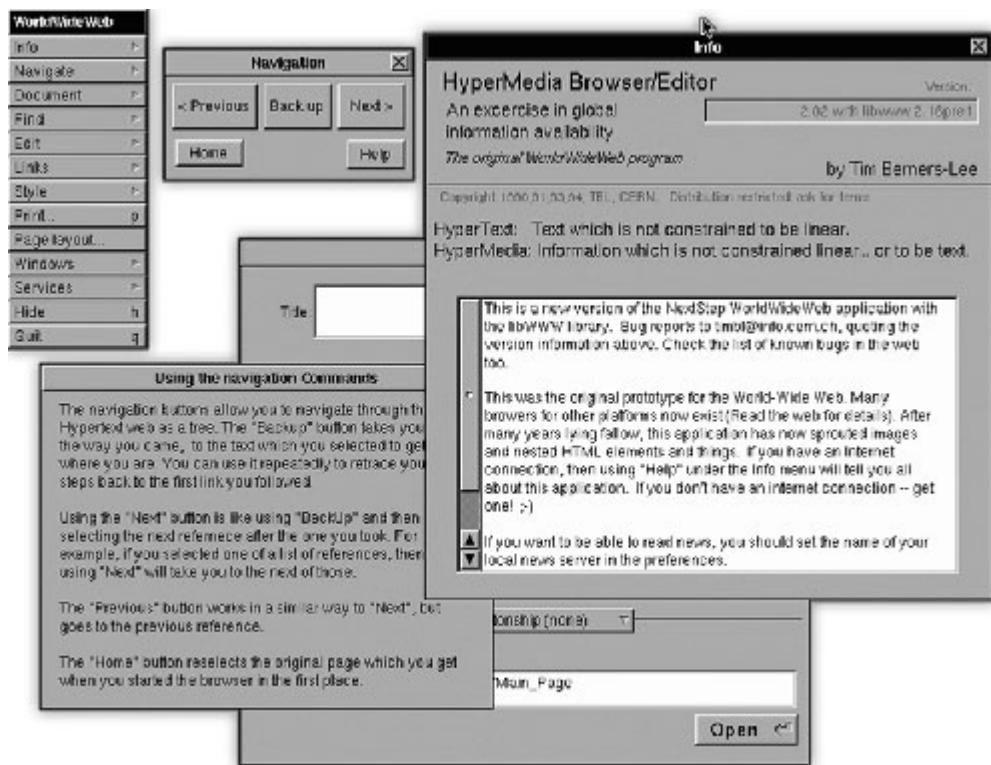
A Web nunca foi um projeto oficial do CERN, mas de algum modo Mike permitiu que Tim trabalhasse nele usando uma NeXT Workstation, fabricada pela empresa de mesmo nome, que foi fundada em 1988 por Steve Jobs e outros veteranos da Apple.

Em outubro de 1990, Tim já tinha criado três tecnologias que seriam o coração da Web:

- HTML: Hypertext Markup Language. Uma linguagem universal e livre para descrever e formatar documentos que seriam exibidos nos aplicativos de visualização da Web, ou browsers.
- URI: Uniform Resource Identifier. Um sistema de endereçamento de páginas que permite dar um nome único a qualquer página dentro da Web, como, por exemplo, [www.inf.ufsc.br](http://www.inf.ufsc.br). Atualmente essa tecnologia é referenciada como URL (Uniform Resource Locator), que passou a ser sinônimo de “endereço na Web”.
- HTTP: Hypertext Transfer Protocol. Um protocolo de comunicação entre clientes (browsers) e servidores (computadores que disponibilizam acesso a páginas na Internet).

Sim, tudo isso, que usamos hoje em nossos computadores e smartphones foi criado por uma única pessoa: Tim Berners-Lee. A Rainha Elisabeth II o proclamou *Sir* em 2004 por seu pioneirismo.

Ele também programou o primeiro browser, que se chamava WorldWideWeb.app ([Figura 9.34](#)) e o primeiro servidor de internet, o httpd. No final de 1990, a primeira página da Web estava no ar e em 1991 pessoas fora do CERN foram convidadas a acessar a recém-nascida Web.



**Figura 9.34:** Interface do browser WorldWideWeb.app de Berners-Lee.<sup>70</sup>

Alguns anos mais tarde esse browser foi rebatizado como Nexus para evitar confusão com o nome do espaço abstrato de informação, “World Wide Web”. Em função da qualidade das ferramentas de desenvolvimento, ele afirma ter criado o browser em apenas dois meses, mas acreditava que em outros ambientes levaria mais de um ano.

As primeiras versões do browser só mostravam texto e links, que eram sublinhados. Ainda não eram azuis porque o NeXT não trabalhava com cores, apenas tons de cinza. Além disso, as imagens não apareciam junto com o texto, mas em janelas separadas.

O endereço do primeiro servidor da WWW era nxoc01.cern.ch, depois renomeado para info.cern.ch. A primeira página da Web foi <http://nxoc01.cern.ch/hypertext/WWW/TheProject.html>, mas ela infelizmente não está mais no ar.

O número de servidores Web passou a crescer exponencialmente ao redor do mundo: em janeiro 1993 eram 50 e em outubro várias centenas. Vários outros browsers foram desenvolvidos em diferentes lugares para diversos computadores. Um dos que mais se popularizou no início da Web foi o Mosaic, lançado em 1993 pela Universidade de Illinois. Ele já funcionava no Unix, Windows e Mac OS. Sua popularidade cresceu devido principalmente à sua capacidade de exibir texto e imagens em uma mesma janela e também pela reputação de seus criadores de rapidamente resolverem problemas e atenderem a recomendações dos usuários. Portanto, essa história continuará.

## 9.58 Até Aqui...

Os anos 1980 presenciaram um fantástico desenvolvimento da computação pessoal. O IBM PC foi lançado e seus clones vieram em seguida. A Microsoft cresceu muito com as vendas do DOS e posteriormente do Windows.

Esse período testemunhou o avanço das interfaces gráficas. O ícone era o Macintosh, embora outros fabricantes também tenham produzido interfaces semelhantes.

Usuários ganharam poder para produzir aplicações com dBASE II. Outros aplicativos populares, como o Word e o Photoshop, surgiram neste período.

Em termos de linguagens de programação, os anos 1980 viram o surgimento de Ada, C++, Turbo Pascal e Eiffel.

O hipertexto foi popularizado pelo Hypercard, mas realmente só tomou o controle das mentes com o World Wide Web. Essa tecnologia ainda estava nascendo no final dos anos 1980, mas mudaria o mundo na década seguinte.

---

<sup>1</sup> “Comodore VIC-20 computer at Play Expo 2013” by Rept0n1x - *Own work*. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Comodore\\_VIC-20\\_computer\\_at\\_Play\\_Expo\\_2013.JPG#mediaFile:Comodore\\_VIC-20\\_computer\\_at\\_Play\\_Expo\\_2013.JPG](https://commons.wikimedia.org/wiki/File:Comodore_VIC-20_computer_at_Play_Expo_2013.JPG#mediaFile:Comodore_VIC-20_computer_at_Play_Expo_2013.JPG)

<sup>2</sup> “Apple III+” by Bilby - *Own work*. Licensed under CC BY 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Apple\\_III%2B.jpg#mediaFile:Apple\\_III%2B.jpg](https://commons.wikimedia.org/wiki/File:Apple_III%2B.jpg#mediaFile:Apple_III%2B.jpg)

<sup>3</sup> “ZX80” by Daniel Ryde, Skövde - *Originally from the Swedish Wikipedia..* Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:ZX80.jpg#mediaFile:ZX80.jpg>

<sup>4</sup> Sistema Operacional Rápido e Sujo.

<sup>5</sup> “StartingMsdos” by *The original uploader was Andrewpmk at English Wikipedia - Transferred from en.wikipedia to Commons by Codename Lisa. Licensed under Public Domain via Commons.* Disponível em: <https://commons.wikimedia.org/wiki/File:StartingMsdos.png#mediaFile:StartingMsdos.png>

<sup>6</sup> Digibarn Computer Museum e Xerox Corp. Cortesia. Disponível em: <http://www.digibarn.com/collections/systems/xerox-8010/>

<sup>7</sup> “Bundesarchiv B 145 Bild-F077948-0006, Jugend-Computerschule mit IBM-PC” by Bundesarchiv, B 145 Bild-F077948-0006 Engelbert Reineke CC-BY-SA 3.0. Licensed under CC BY-SA 3.0 de via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Bundesarchiv\\_B\\_145\\_Bild-F077948-0006,\\_Jugend-Computerschule\\_mit\\_IBM-](https://commons.wikimedia.org/wiki/File:Bundesarchiv_B_145_Bild-F077948-0006,_Jugend-Computerschule_mit_IBM-)

[PC.jpg#mediaFile:Bundesarchiv\\_B\\_145\\_Bild-F077948-0006,\\_Jugend-Computerschule\\_mit\\_IBM-PC.jpg](#)

<sup>8</sup> “CGA video card” by Jakub Torenc - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:CGA\\_video\\_card.JPG#mediaFile:CGA\\_video\\_card.JPG](https://commons.wikimedia.org/wiki/File:CGA_video_card.JPG#mediaFile:CGA_video_card.JPG)

<sup>9</sup> “Osborne01” by Photographer: Tomislav Medak from Flickr / Editing: Bill Bertram (Pixel8) - Disponível em: <http://www.flickr.com/photos/tomislavmedak/3803228025>. Licensed under CC BY-SA 3.0 via Commons. <https://commons.wikimedia.org/wiki/File:Osborne01.jpg#mediaFile:Osborne01.jpg>

<sup>10</sup> Tradução: Os três amigos.

<sup>11</sup> Tradução: Nós definimos o padrão.

<sup>12</sup> “Computer Museum Sun-1” by cmnit - Flickr: Computer Museum: Sun-1. Licensed under CC BY-SA 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Computer\\_Museum\\_Sun-1.jpg#mediaFile:Computer\\_Museum\\_Sun-1.jpg](https://commons.wikimedia.org/wiki/File:Computer_Museum_Sun-1.jpg#mediaFile:Computer_Museum_Sun-1.jpg)

<sup>13</sup> Tradução: Projeto Auxiliado por Computador.

<sup>14</sup> Tradução: Sistema de Arquivos para Rede.

<sup>15</sup> By Bill Bertram - Own work, CC BY-SA 2.5. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=133083>

<sup>16</sup> “BBC Micro” by No machine-readable author provided. StuartBrady assumed (based on copyright claims). - No machine-readable source provided. Own work assumed (based on copyright claims). Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:BBC\\_Micro.jpeg#mediaFile:BBC\\_Micro.jpeg](https://commons.wikimedia.org/wiki/File:BBC_Micro.jpeg#mediaFile:BBC_Micro.jpeg)

<sup>17</sup> Tradução: O Poderoso Micro.

<sup>18</sup> “GRiDCompassInSpace” by NASA – Transferred from en.wikipedia to Commons. Licensed under Public Domain via Wikimedia Commons. Disponível em:

<https://commons.wikimedia.org/wiki/File:GRiDCompassInSpace.jpg#mediaFile:GRiDCompassInSpace.jpg>

<sup>19</sup> “286 CPU” by Pascal - Disponível em: <https://www.flickr.com/photos/pasukaru76/3536246838/>. Licensed under CC0 via Wikimedia Commons.

[https://commons.wikimedia.org/wiki/File:286\\_CPU.jpg#mediaFile:286\\_CPU.jpg](https://commons.wikimedia.org/wiki/File:286_CPU.jpg#mediaFile:286_CPU.jpg)

<sup>20</sup> “Compaq portable-IMG 7218” by Rama & Musée Bolo - Own work. Licensed under CC BY-SA 2.0 fr via Wikimedia Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:Compaq\\_portable-IMG\\_7218.jpg#mediaFile:Compaq\\_portable-IMG\\_7218.jpg](https://commons.wikimedia.org/wiki/File:Compaq_portable-IMG_7218.jpg#mediaFile:Compaq_portable-IMG_7218.jpg)

<sup>21</sup> Tradução: Projeto em sala limpa.

<sup>22</sup> “ZX Spectrum48k” by Bill Bertram - *Own work. Licensed under CC BY-SA 2.5 via Commons.* Disponível em: <https://commons.wikimedia.org/wiki/File:ZXSpectrum48k.jpg#mediaFile:ZXSpectrum48k.jpg>

<sup>23</sup> Referência à crise que se seguiu nos Estados Unidos após a União Soviética ter lançado com sucesso o primeiro satélite em órbita da Terra durante a Guerra Fria.

<sup>24</sup> “Apple IIe Inside” by StromBer. *Licensed under CC BY-SA 3.0 via Wikimedia Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Apple\\_Ile\\_Inside.JPG#mediaFile:Apple\\_Ile\\_Inside.JPG](https://commons.wikimedia.org/wiki/File:Apple_Ile_Inside.JPG#mediaFile:Apple_Ile_Inside.JPG)

<sup>25</sup> “Apple Lisa (Little Apple Museum) (8032162544)” by Álvaro Ibáñez from Madrid, Spain - *Apple Lisa (Little Apple Museum).* Licensed under CC BY 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Apple\\_Lisa\\_\(Little\\_Apple\\_Museum\)\\_\(\\_8032162544\).jpg#mediaFile:Apple\\_Lisa\\_\(Little\\_Apple\\_Museum\)\\_\(\\_8032162544\).jpg](https://commons.wikimedia.org/wiki/File:Apple_Lisa_(Little_Apple_Museum)_(_8032162544).jpg#mediaFile:Apple_Lisa_(Little_Apple_Museum)_(_8032162544).jpg)

<sup>26</sup> Tradução: Arquitetura de Sistema Local Integrada.

<sup>27</sup> Tradução: LISA, Acrônimo Bobo Inventado.

<sup>28</sup> Tradução: Vamos Inventar Algum Acrônimo.

<sup>29</sup> “Ibm px xt color” by Ruben de Rijcke - *Own work. Licensed under CC BY 3.0 via Wikimedia Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Ibm\\_px\\_xt\\_color.jpg#mediaFile:Ibm\\_px\\_xt\\_color.jpg](https://commons.wikimedia.org/wiki/File:Ibm_px_xt_color.jpg#mediaFile:Ibm_px_xt_color.jpg)

<sup>30</sup> “Ibm pcjr with display” by Rik Myslewski - *Own work. Licensed under CC0 via Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Ibm\\_pcjr\\_with\\_display.jpg#mediaFile:Ibm\\_pcjr\\_with\\_display.jpg](https://commons.wikimedia.org/wiki/File:Ibm_pcjr_with_display.jpg#mediaFile:Ibm_pcjr_with_display.jpg)

<sup>31</sup> By User:Materialgeeza - *File:Stereolithography apparatus.jpg*, CC BY-SA 3.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=25850809>

<sup>32</sup> Tradução: O Sistema de Compartilhamento de Tempo Incompatível.

<sup>33</sup> Tradução: “GNU não é Unix”.

<sup>34</sup> “Heckert GNU white” por Aurelio A. Heckert <aurium@gmail.com> - *gnu.org. Licenciado sob CC BY-SA 2.0, via Wikimedia Commons.* Disponível em: [https://commons.wikimedia.org/wiki/File:Heckert\\_GNU\\_white.svg#mediaFile:Heckert\\_GNU\\_white.svg](https://commons.wikimedia.org/wiki/File:Heckert_GNU_white.svg#mediaFile:Heckert_GNU_white.svg)

<sup>35</sup> Tradução: Estojo de Compilador da Universidade Livre. (O acrônimo desse sistema em inglês deveria fazer muito sucesso.)

<sup>36</sup> Tradução: Protocolo de Transferência de Arquivos.

<sup>37</sup> Tradução: Grande Irmão.

<sup>38</sup> “Macintosh 128k transparency” by w:User:Grm wnr - Modifications of Image:Macintosh 128k.jpg and w:Image:Macintosh 128k No Text.jpg. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Macintosh\\_128k\\_transparency.png#mediaFile:Macintosh\\_128k\\_transparency.png](https://commons.wikimedia.org/wiki/File:Macintosh_128k_transparency.png#mediaFile:Macintosh_128k_transparency.png)

<sup>39</sup> Tradução: diretório vazio.

<sup>40</sup> “Microsoft Windows 1.0 pages2 3” by Microsoft. - Brochure scanned by Swtpc6800, Michael Holley, in November 2010. Licenced under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Microsoft\\_Windows\\_1.0\\_pages2\\_3.jpg#mediaFile:Microsoft\\_Windows\\_1.0\\_pages2\\_3.jpg](https://commons.wikimedia.org/wiki/File:Microsoft_Windows_1.0_pages2_3.jpg#mediaFile:Microsoft_Windows_1.0_pages2_3.jpg)

<sup>41</sup> “Atari 1040STf”. Licensed under CC BY-SA 2.5 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Atari\\_1040STf.jpg#mediaFile:Atari\\_1040STf.jpg](https://commons.wikimedia.org/wiki/File:Atari_1040STf.jpg#mediaFile:Atari_1040STf.jpg)

<sup>42</sup> Tradução: Interface Digital para Instrumentos Musicais.

<sup>43</sup> “Amiga 1000” by Original uploader was Kal-El at bs.wikipedia - Originally from bs.wikipedia; Licenced under CC BY-SA 3.0 via Wikimedia Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:Amiga\\_1000.jpg#mediaFile:Amiga\\_1000.jpg](https://commons.wikimedia.org/wiki/File:Amiga_1000.jpg#mediaFile:Amiga_1000.jpg)

<sup>44</sup> Traduação: Interface de Programação de Aplicações.

<sup>45</sup> Tradução: João Travesseiro.

<sup>46</sup> Sophie Wilson é transexual e no início do projeto do ARM ainda se chamava Roger Wilson. Hoje é uma das mais importantes profissionais na área de eletrônica de microprocessadores.

<sup>47</sup> “CD autolev crop” por Ubern00b - Obra do próprio. Licenciado sob CC BY-SA 3.0, via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:CD\\_autolev\\_crop.jpg#mediaFile:CD\\_autolev\\_crop.jpg](https://commons.wikimedia.org/wiki/File:CD_autolev_crop.jpg#mediaFile:CD_autolev_crop.jpg)

<sup>48</sup> Tradução: Corporação das Máquinas Pensantes.

<sup>49</sup> Tradução: “Ei, nós acertamos! Nada mal para amadores.”

<sup>50</sup> “Thinking machines cm2” by Don Armstrong - Don Armstrong - Disponível em: [http://gallery.donarmstrong.com/2005/08\\_august/computer\\_museum\\_20050813/010\\_thinking\\_machines\\_20050813.jpg](http://gallery.donarmstrong.com/2005/08_august/computer_museum_20050813/010_thinking_machines_20050813.jpg). Licensed under GPL via Commons.

[https://commons.wikimedia.org/wiki/File:Thinking\\_machines\\_cm2.jpg#mediaFile:Thinking\\_machines\\_cm2.jpg](https://commons.wikimedia.org/wiki/File:Thinking_machines_cm2.jpg#mediaFile:Thinking_machines_cm2.jpg)

<sup>51</sup> Tradução: “A trapaça da Amstrad contra a IBM.”

<sup>52</sup> Tradução: “Eu sou uma pessoa realista e nós somos uma organização de mercado. Assim, se essa é a diferença entre as pessoas comprarem o computador ou não, eu posso enfiar um maldito ventilador nele. E se eles quiserem bolinhas cor de rosa nele eu coloco também. Qual é a vantagem de eu ficar batendo minha cabeça em um muro de tijolos e ficar dizendo: ‘você não precisa do maldito ventilador, queridinho?’”

<sup>53</sup> “I386SX” by The original uploader was WikiPancu at English Wikipedia - Transferred from en.wikipedia to Commons. Licenced under CC BY-SA 3.0 via Wikimedia Commons. Disponível em:

<https://commons.wikimedia.org/wiki/File:I386SX.jpg#mediaFile:I386SX.jpg>

<sup>54</sup> Tradução: Um Sistema Operacional RISC antes da Quinta-feira.

<sup>55</sup> Licensed under CC BY-SA 3.0. Fonte: Chris's Acorns. Disponível em: <http://chrisacorns.computinghistory.org.uk/Computers/A300.html>

<sup>56</sup> Tradução: “Era o que nós costumávamos chamar à época de ‘cavalgar um urso’. Você apenas tinha que tentar permanecer nas costas do urso, e ele iria girar e corcovear e tentar atirar você de lá, mas nós íamos ficar sobre o urso, porque ele era o maior, o mais importante... Você somente tinha que ficar com o urso, caso contrário, você estaria debaixo dele.”

<sup>57</sup> “Atm os2warp” by Malvineous - Own work. Licenced under Public Domain via Wikimedia Commons. Disponível em:

[https://commons.wikimedia.org/wiki/File:Atm\\_os2warp.jpg#mediaFile:Atm\\_os2warp.jpg](https://commons.wikimedia.org/wiki/File:Atm_os2warp.jpg#mediaFile:Atm_os2warp.jpg)

<sup>58</sup> Tradução: Linguagem Prática de Extração e Relatórios.

<sup>59</sup> Tradução: Um Organizador de Lixo Patologicamente Eclético.

<sup>60</sup> “Vector Video Standards2” by Original uploader was XXV at en.wikipedia Later version(s) were uploaded by Jjalocha, Aihtdikh at en.wikipedia. - Transferred from en.wikipedia. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Vector\\_Video\\_Standards2.svg#mediaFile:Vector\\_Video\\_Standards2.svg](https://commons.wikimedia.org/wiki/File:Vector_Video_Standards2.svg#mediaFile:Vector_Video_Standards2.svg)

<sup>61</sup> ©Adobe Systems Inc. Cortesia. Disponível em: <http://www.webdesignerdepot.com/2010/02/20-years-of-adobe-photoshop/>

<sup>62</sup> Tradução: “photosopear” uma imagem.

<sup>63</sup> Tradução: A Minhoca.

<sup>64</sup> Tradução: “O piloto da onda de choque”.

<sup>65</sup> Essa palavra pode ser um trocadilho: “tapeworm” literalmente é o parasita conhecido como “tênia”, mas a palavra também pode ser lida como “minhoca de

fita”, aludindo a algo que se insinua nos meios de armazenamento de computadores mais antigos: as fitas magnéticas.

<sup>66</sup> Tradução: Protocolo de Transferência de Arquivos.

<sup>67</sup> “Poqet 1” by *The original uploader was Sandos at English Wikipedia - Transferred from en.wikipedia to Commons. Licenced under CC BY-SA 3.0 via Wikimedia Commons.* Disponível em:

[https://commons.wikimedia.org/wiki/File:Poqet\\_1.jpg#mediaFile:Poqet\\_1.jpg](https://commons.wikimedia.org/wiki/File:Poqet_1.jpg#mediaFile:Poqet_1.jpg)

<sup>68</sup> Tradução: “Se essa coisa do Microsoft Windows não arrasar, o Poqet vai dominar o mundo.”

<sup>69</sup> Tradução: “Gerenciamento de Informação - uma proposta.”

<sup>70</sup> “WorldWideWeb.1” by *Ck\_mpk - Own work. Licensed under Public Domain via Wikimedia Commons.* Disponível em:

<https://commons.wikimedia.org/wiki/File:WorldWideWeb.1.png#mediaFile:WorldWideWeb.1.png>

# **PARTE X**

## **O Mundo se Conecta na World Wide Web**

Os anos 1990 viram o crescimento vertiginoso da World Wide Web e das ferramentas relacionadas a ela, como os browsers Mosaic, Netscape e Internet Explorer, além de buscadores como o Wanderer, Altavista e Google. Se os anos 1980 foram marcados pela guerra dos clones contra a IBM, os anos 1990 foram marcados pela guerra dos browsers, que quase levou a Microsoft a ser dividida em duas empresas.

Várias novas linguagens de programação especialmente voltadas para a Web foram criadas neste período, tais como Java, PHP, Python, Ruby e JavaScript. Os anos 1990 também viram a miniaturização dos dispositivos computacionais, com o surgimento dos palmtops, como o Pilot e o primeiro smartphone, chamado Simon. Além disso, a miniaturização dos dispositivos de armazenamento de dados também continuou, com o surgimento dos flash drives.

Para fechar com “chave de ouro”, o mundo se viu às voltas com o famoso “bug do milênio” que mostrou de uma vez por todas o quanto a sociedade já estava à mercê da computação... e de seus equívocos.

### **10.1 PowerPoint – 1990**

O PowerPoint certamente é o programa mais usado em salas de aula em todo o mundo até hoje. A ideia é atribuída a Robert

Gaskins, que queria um programa que permitisse com facilidade a criação de slides para apresentações de aulas e palestras. O sistema foi programado por Thomas Rudkin e Dennis Austin da empresa Forethought Inc. Seu nome original era “Presenter”, mas foi rebatizado como “PowerPoint” porque “Presenter” já era marca registrada.

Ele foi desenvolvido inicialmente para o Macintosh ([Figura 10.1](#)). Em 1987, a Forethought foi comprada pela Microsoft por 14 milhões de dólares e tornou-se sua unidade de gráficos para negócios, continuando a desenvolver o PowerPoint até seu lançamento em maio de 1990 juntamente com o Windows 3.0.

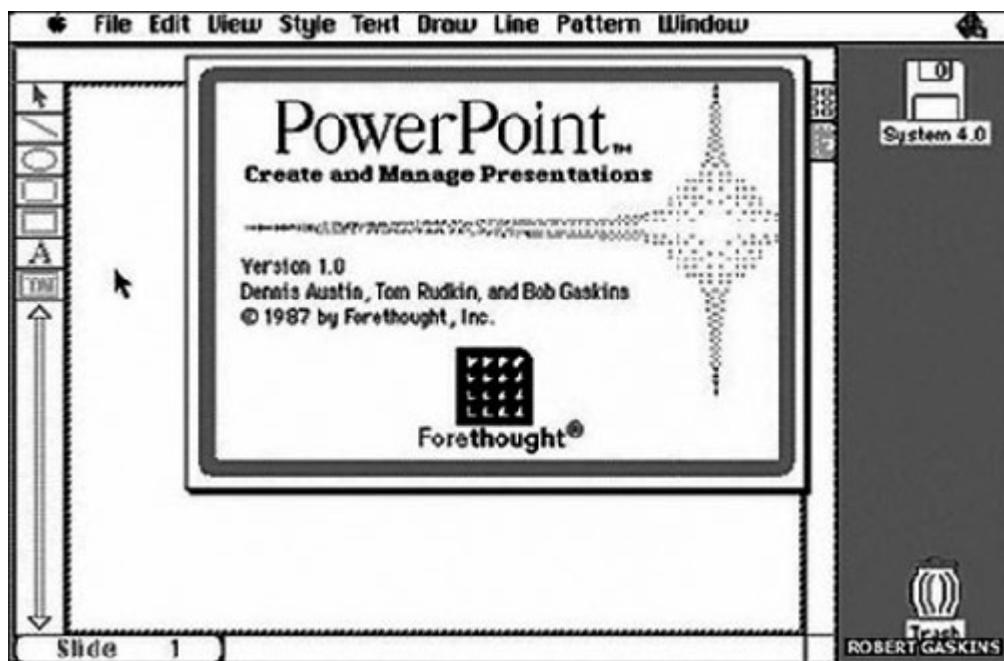


Figura 10.1: Microsoft PowerPoint ® 1.0 para Macintosh.<sup>1</sup>

Nos primeiros tempos, porém, não havia projetores multimídia. Então o objetivo do PowerPoint era permitir a edição do conjunto de slides que depois seriam impressos em lâminas plásticas, para serem apresentados com retroprojetores.

Desde cedo o PowerPoint permitiu o uso de animações e gráficos nos slides. Mais tarde até filmes podiam ser incluídos nas apresentações. Embora ele tenha sido demonizado por alguns que consideram apresentações baseadas em PowerPoint usualmente cansativas e pouco produtivas, acredita-se que tenha pouparado muito tempo de professores e executivos, que adquiriram a condição de rapidamente preparar apresentações que, sem ele, talvez nem fossem tão boas. Quem estudou até os anos 1980 deve lembrar dos professores que ditavam matéria aos alunos ou, os mais modernos, que traziam lâminas plásticas para projetar, as quais muitas vezes já apresentavam as marcas de ferrugem dos clipes que as mantinham presas. Gaskins diz que apresentações em lâminas já existiam quando o PowerPoint foi criado, e muitas delas eram cansativas e repetitivas. Ele, portanto, não as inventou, apenas facilitou sua produção.

Porém, uma crítica que Gaskins aceitou foi o fato de que o PowerPoint possibilitou a sofisticação dessas apresentações de forma que muitas vezes o seu excesso de cores, fontes e imagens pouco adicionavam para a compreensão da mensagem. Pior que isso: esses efeitos podem ser usados para distrair a audiência da mensagem.

O PowerPoint praticamente não teve concorrentes no começo, o que se mantém até hoje. Mesmo com clones gratuitos disponíveis, ele ainda é utilizado em pelo menos 1 bilhão de computadores mundo afora. Dentre os seus concorrentes iniciais destaca-se o Hipercard, que era usado também para construir apresentações. Porém, o PowerPoint rapidamente o passou em número de usuários.

## 10.2 Python – 1991

Python surgiu em 1991 como uma linguagem que procurava simplificar a tarefa de programação. Mais ou menos na mesma linha de Perl, como vimos anteriormente, Python procurava minimizar a

quantidade de decisões que um programador precisava tomar quando fosse desenvolver um programa. Por exemplo, “o array que vou usar deve ter 20 ou 30 posições?”. Em linguagens como Pascal, essa preocupação era relevante porque a linguagem exigia que o programador decidisse o tamanho do array, mas em Python isso era irrelevante, porque os arrays e outras estruturas nessa linguagem são extremamente flexíveis. Devido a sua sintaxe limpa e fama de ser uma linguagem altamente produtiva, Python tem sido cada vez mais escolhida como a linguagem para ensinar programação das universidades mundo afora.

O seu criador foi Guido van Rossum (Holanda, 1956). Ele iniciou o projeto de Python no Instituto Nacional de Pesquisas em Matemática e Ciência da Computação na Holanda no final dos anos 1980.

A principal influência para a concepção de Python foi a linguagem ABC, um projeto no qual Rossum trabalhou nos anos 1980. O objetivo do projeto era a criação de uma linguagem de programação que pudesse ser ensinada a usuários de computador especializados, mas que não fossem programadores, como por exemplo físicos, matemáticos, linguistas etc. A experiência de ensinar linguagens de programação clássicas para esse tipo de profissional normalmente é frustrante porque elas exigem conhecimento profundo de detalhes do funcionamento dos computadores para que o aluno consiga começar a fazer seus primeiros programas. Decidir, por exemplo, se uma lista deve ser implementada como lista encadeada, array, árvore binária ou qualquer outra estrutura de baixo nível não devia ser uma preocupação para um programador eventual: ele tem que ser capaz definir uma lista e pronto!

Na época em que o projeto ABC iniciou, a linguagem mais usada para ensinar programação para esse tipo de usuário era BASIC, mas essa linguagem antiga já sofria de esclerose múltipla, com estruturas de baixo nível imbricadas na programação em alto nível. Por exemplo, para trocar a cor de fundo da tela era necessário, em

algumas implementações de BASIC, armazenar um determinado valor numérico em uma determinada posição de memória que você tinha que saber qual era. Esse tipo de coisa não deveria existir em ABC e nem em Python.

Em 1986, Rossum passou a trabalhar no projeto de um sistema operacional distribuído. Ele criou uma espécie de ambiente de produção utilizando alguns dos conceitos de ABC de que gostava e deixando de fora os que achava ruins. Ele tentou criar para esse sistema operacional uma linguagem de script que tivesse apenas as características boas de ABC.

A principal característica de Python que o diferenciava de ABC, segundo ele era a extensibilidade. ABC foi criada por um comitê de designers que, talvez de forma um tanto arrogante, fez um projeto fechado. Eles simplesmente tentaram prever as possibilidades e antever todos os cenários de uso da linguagem. Já Python seria uma linguagem extensível na qual futuros usuários poderiam acrescentar novas características à medida que a necessidade fosse surgindo.

Uma característica única dela é o uso da endentação. Na maioria das linguagens de programação estruturadas, recomenda-se que comandos subordinados a estruturas de seleção ou repetição estejam endentados, ou seja, se você escrever um comando de repetição como “repita 4 vezes:”, os comandos que vão ser repetidos quatro vezes estarão nas linhas logo abaixo. Neste caso, recomenda-se que eles sejam endentados, ou seja, que o texto com o comando comece algumas posições mais à direita do que a linha à qual eles estão subordinados. Por exemplo:

Repita 4 vezes:

    Ande 10 para a frente

    Gire 90 graus

Fim

O problema é que essas linguagens também permitem que o programador escreva:

Repita 4 vezes:

Ande 10 para a frente

Gire 90 graus

Fim

Algumas ainda permitem:

Repita 4 vezes: Ande 10 para a frente Gire 90 graus Fim

Então a endentação, nessas linguagens, é só uma forma de organizar visualmente o texto, mas não tem nenhum efeito sobre o que vai ser executado. O que define o final da estrutura de repetição no caso anterior é o comando “Fim”.

Linguagens como Pascal não tem um comando “Fim” associado diretamente ao comando de repetição ou seleção. Se você escrever em Pascal um comando de repetição e colocar dois comandos após ele, mesmo que estejam endentados, apenas o primeiro comando será repetido. O segundo só será executado uma única vez depois que a repetição terminar:

```
FOR i:=1 TO 4 DO
```

```
aFrente(10);
```

```
giraDireita(90);
```

A intenção do programador acima provavelmente seria repetir os dois comandos, mas apenas o primeiro será repetido. Pascal e outras linguagens exigem a utilização de um bloco quando se quer subordinar mais de um comando a uma estrutura de repetição ou seleção. No caso de Pascal, o bloco é marcado pelos comandos BEGIN e END:

```
FOR i:=1 TO 4 DO
```

```
BEGIN
```

```
aFrente(10);
```

```
giraDireita(90)2
```

```
END;
```

Se o leitor for programador, sabe do que estou falando; se não for, talvez não tenha entendido muito bem, mas este é exatamente o caso: entender essa característica sintática arbitrária das linguagens de programação não tornaria você um programador melhor. Esse tipo de coisa não ajuda uma pessoa a resolver problemas com algoritmos, apenas comprova que ela conhece a fundo as idiossincrasias de uma linguagem de programação.

Bem, Python acaba com isso. A linguagem elimina as estruturas de bloco no estilo BEGIN-END e passa a tratar a própria endentação como uma característica sintática da linguagem. Assim, se você quiser repetir os dois comandos em Python vai escrever:

```
for i in range (1, 4):
```

```
    aFrente(10)
```

```
    giraDireita(90)
```

Por outro lado, se você quisesse repetir apenas o comando aFrente e depois executar o comando giraDireita, escreveria assim:

```
for i in range (1, 4):
```

```
    aFrente(10)
```

```
    giraDireita(90)
```

Logo, a endentação é parte da sintaxe e o conceito de bloco se torna desnecessário. Nas linguagens tradicionais, você teria que usar os dois conceitos ao mesmo tempo e poderia utilizá-los de forma inconsistente, como vimos anteriormente.

Em fevereiro de 2016 Python aparecia em 5º lugar na tabela TIOBE, perdendo apenas para Java e três variantes de C.

## 10.3 Linux – 1991

Linus B. Torvalds (Finlândia, 1969) começou a aprender informática ainda criança com o Commodore Vic-20 que seu avô materno comprou nos anos 1970. Ele tinha apenas 10 anos quando se chateou com a pouca quantidade de programas disponíveis, e começou a escrever os seus próprios. Inicialmente, usava a

linguagem BASIC, mas logo ela se tornou insuficiente e ele começou a utilizar o Assembly.

Pode-se dizer que como adolescente ele foi um típico “nerd”. Seus interesses principais durante o Ensino Médio eram a matemática e a programação. Seus pais tentavam sem sucesso fazê-lo se interessar por esportes, festas e garotas. De fato, ele só se casou anos mais tarde com uma mulher que o convidou para sair depois de assistir a uma de suas palestras.

Em 1988, ele iniciou o curso de Ciência da Computação na Universidade de Helsinki e, embora já fosse um exímio programador, foi ali que ele aprendeu a linguagem C, que usaria depois para escrever o kernel do Linux.

Em 1991 ele comprou um clone de PC com processador 386, 4 MB de memória e sistema operacional MS-DOS. Porém, ficou muito frustrado porque percebeu que o DOS, feito para o 8088, praticamente não explorava nenhuma das características poderosas do novo processador. Assim, ele decidiu que iria usar Unix em vez do DOS no seu PC. O problema era encontrar uma licença de Unix por menos de 5 mil dólares.

Ele acabou comprando um clone do Unix bem mais barato chamado MINIX, que foi desenvolvido pelo professor Andrew Tanenbaum (Estados Unidos, 1944) como uma ferramenta para ensinar Unix aos alunos da universidade. Porém, esse sistema, além do fato de não ser gratuito, ainda tinha desvantagens: ele não era tão eficiente quanto seu original e parte do seu código-fonte não era aberto.

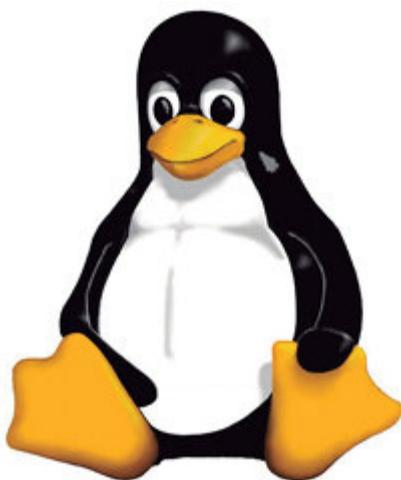
Linus queria usar seu PC em casa como um terminal do computador da universidade, que rodava Unix. Mas o MINIX também não permitia isso. Ele decidiu então criar seu próprio programa de emulação de terminal independente do MINIX usando o compilador C do projeto GNU.

Ele rapidamente construiu seu emulador de terminal e logo se motivou a adicionar outras potencialidades como, por exemplo, a possibilidade de salvar e transferir arquivos. Assim ia nascendo um

novo sistema operacional, inspirado no Unix, mas livre e independente, e que viria a se chamar Linux.

Não pense que Linus era narcisista. Ele na verdade batizou seu sistema como Freax, uma combinação de “free”<sup>3</sup> com o “x” de Unix. Porém, quando ele disponibilizou o arquivo com este nome no servidor de FPT da FUNET, seu amigo Ari Lemmke (Finlândia, 1963), um dos administradores do servidor de FTP, achou que o nome ia pegar mal porque soava como “Freaks”<sup>4</sup> e (felizmente) renomeou o arquivo como “Linux”: o Unix de Linus. Mais tarde Linus concordou com o nome, que já tinha “pegado”.

Em agosto de 1991, Linus anunciou o novo sistema em um grupo de notícias USENET de usuários de MINIX. Desde o primeiro momento, o código seria livre e gratuito. Tanenbaum escreveu no ano seguinte algumas críticas ao Linux, que foram rebatidas por Linus imediatamente. Desde então o sistema tem crescido sem parar, tanto em características e aplicativos, quanto em número de usuários ao redor do mundo. Seu logo consiste de um pinguim gordinho sentado e aparentando estar bastante satisfeito consigo mesmo ([Figura 10.2](#)).



**Figura 10.2:** Tux, o pinguim – o logotipo de Linux.<sup>5</sup>

Em 1997 Linus foi trabalhar na Califórnia em uma empresa que desenvolvia processadores com baixo consumo de energia. Em 2003, ele deixa a companhia para trabalhar exclusivamente para o OSDL (Open Source Development Labs), um consórcio formado por empresas como IBM, HP, Intel, AMD, RedHat e Novell para promover o desenvolvimento do Linux. A OSDL fundiu-se com a Free Standards Group em 2007 para se tornar a Fundação Linux.

No ano em que este livro foi escrito (2016), Linux é pouco usado em computadores pessoais, com apenas 1,5% do mercado, mas é o grande preferido quando se trata de servidores de Web, onde está instalado em nada menos do que 96,55% das máquinas. O sistema operacional Android, bastante usado em smartphones (79,3% do mercado) foi construído com base no kernel do Linux e o Chrome OS também.

## 10.4 NCSA Mosaic – 1993

Em 1992 a World Wide Web não era conhecida ainda senão por um pequeno grupo de pesquisadores em universidades e instituições de pesquisa que tinham os computadores e a capacidade técnica para instalar browsers e servidores.

O CERN, em Genebra, berço da WWW, começou nessa época a usar um browser chamado ViolaWWW, que só funcionava em Unix e foi criado por Pei-Yuan Wei, na Universidade da Califórnia em Berkeley. Porém, o grande público ainda não tinha ouvido falar nessa tal de “World Wide Web”. A Internet vivia de serviços como FTP, USENET, GOPHER e outros, mas ainda pouquíssimas pessoas sabiam o que era um browser.

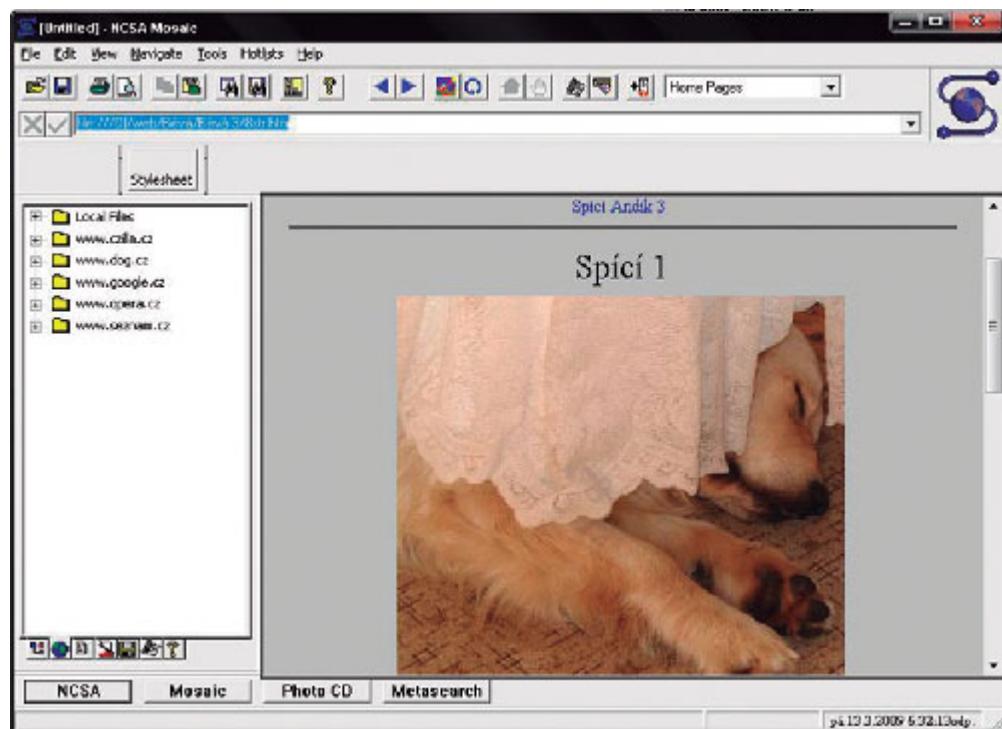
O primeiro passo para a explosão da Internet foi dado no dia em que um grupo de pesquisadores do NCSA, o Centro Nacional de Aplicações de Supercomputadores da Universidade de Illinois, ouviu falar no trabalho desenvolvido no CERN e baixou uma versão do ViolaWWW. O browser foi demonstrado ao grupo de design de

software do NCSA que pôde visualizar a página do CERN através da Internet, e todos ficaram muito espantados.

Dois estudantes desse grupo, Marc Andreessen (Estados Unidos, 1971) e Eric Bina (Estados Unidos, 1964), iniciaram o desenvolvimento de um browser próprio para o NCSA, que rodaria sobre o sistema X Windows que é a interface gráfica de sistemas baseados em Unix. O novo browser foi lançado no início de 1993 e divulgado nos grupos de notícias da USENET por Berners-Lee do CERN menos de uma semana depois.

O nome escolhido para o produto foi “Mosaic”, porque ele seria capaz de suportar uma vasta gama de protocolos da Internet como http, ftp, gopher etc. A [Figura 10.3](#) mostra a interface do Mosaic.

Uma versão para Macintosh foi lançada ainda no mesmo ano, fazendo do Mosaic o primeiro browser a ser implementado em mais de uma plataforma. Antes do final do ano já havia uma versão também para o Microsoft Windows.



**Figura 10.3: Tela do Mosaic 1.0 no Macintosh.<sup>6</sup>**

Robert Metcalfe (Estados Unidos, 1946), coinventor da Ethernet, comenta em 1995 que quando Tim Berners-Lee criou HTTP, URL e HTML para o Unix, algumas poucas pessoas puderam perceber que a Web poderia ser melhor do que o Gopher,<sup>7</sup> mas quando Andreessen e Bina criaram o Mosaic, muitos milhões de pessoas perceberam que a Web poderia ser melhor do que sexo.

Porém, o que fez do Mosaic o aplicativo que mudaria a face da Internet? Várias coisas... Ele foi desenvolvido para várias plataformas, inclusive Macintosh e Windows que era o que a maioria das pessoas estava usando na época. Unix sempre foi um sistema mais usado por profissionais, não pelo grande público. Além disso, o browser é reputado como o primeiro a apresentar imagens e texto na mesma tela. Os anteriores separavam esses dois elementos em janelas diferentes. Talvez mais importante de tudo: o Mosaic era muito fácil de instalar. Ele podia ser baixado da internet gratuitamente através de FTP e se instalava sozinho sem necessitar de maiores configurações.

Outra coisa que ajudou a espalhar o Mosaic foi a reputação: Andreessen era muito cuidadoso com o código, procurando sempre fazê-lo o mais perfeito possível e Bina estava sempre atento aos grupos de notícias da USENET. Se houvesse quaisquer comentários sobre problemas com o Mosaic, como bugs, eles rapidamente consertavam e liberavam uma nova versão aperfeiçoada.

Porém, a ascensão do NCSA Mosaic ao topo logo foi freada por um browser ainda melhor: o Netscape Navigator. A partir de 1995, a grande maioria dos usuários do Mosaic passa a usar o Netscape. Inclusive, Andreessen e Bina foram trabalhar nessa empresa sobre a qual falaremos depois. O desenvolvimento do Mosaic foi oficialmente descontinuado pelo NCSA em 1997.

## **10.5 Pentium – 1993**

Em 1993 os nerds de todo o mundo tiveram uma surpresa: a Intel lançava a quinta geração da linha de processadores x86 e, em vez de batizá-lo como i80586, como todos esperavam, ela chamou o novo processador de “Pentium”. Como assim? Se os robôs de *Guerra nas Estrelas*, as naves espaciais de *Jornada nas Estrelas* e quase todos os computadores até o momento tinham em seus nomes combinações de letras e números, que diabo era “Pentium”?

Acontece que a Intel estava farta de ver clones de seus processadores compartilhando a parte numérica do nome. Por exemplo, a Intel criou o i486, mas não podia impedir outras fabricantes de batizarem seus processadores com o mesmo número e outra combinação de letras, como, por exemplo, o Am486 da AMD. A Intel bem que tentou, mas os tribunais disseram que números não podiam ser registrados como marca. Assim, abandonaram essa prática e passaram a dar nomes aos seus processadores.

O Pentium foi um processador muito importante e chegou a ser durante alguns anos sinônimo de PC. As pessoas diziam: “Comprei um Pentium!” Ele foi tão importante que ainda é fabricado hoje, embora as versões atuais sejam bem diferentes do original com seus 3 milhões de transistores e 66 MHz.

A Intel pagou um preço alto pela escolha da retrocompatibilidade com a família x86. Para manter os antigos programas rodando no Pentium, ela teve que manter determinadas características legadas. Não puderam, por exemplo, optar por uma arquitetura RISC, que vinha mostrando enormes vantagens em relação à CISC. Assim, ocorre que o Pentium, embora bem superior ao 486, perdia feio para os processadores RISC.

Essa escolha foi consciente, porém. A Intel mirava com esse processador o mercado de computadores pessoais e comerciais. Ele podia não ser excelente para processamento vetorial ou cálculos científicos complexos, mas rodava as planilhas, bancos de dados e

processadores de texto, e era isso o que importava para esse mercado.

No ano seguinte, porém, a Intel teve seu próprio fiasco. Um erro no processador fazia com que ele retornasse um resultado incorreto em alguns casos. O erro foi descoberto pelo professor Thomas R. Nicely, que o reportou à Intel. A empresa disse que já sabia do erro, mas que não dava importância porque como era uma operação muito específica isso não afetaria a grande maioria dos usuários. A revista *Byte* estimou na época que a probabilidade de ocorrência do erro era de 1 em 9 bilhões, caso fossem feitas divisões de números aleatórios em ponto flutuante.

O caso veio a público e se tornou reportagem de impacto na CNN. Pressionada pela opinião pública, a Intel ofereceu trocar o processador defeituoso por um novo para as pessoas que solicitassem. Poucos usuários realmente solicitaram a troca, mas o maior prejuízo da Intel foi em relação à sua imagem como empresa confiável.

Em relação aos processadores defeituosos recebidos por ela, foram transformados em chaveiros. Na frente via-se o interior do processador e a expressão “*Intel Inside*”. No verso constava uma frase de Andie Grove (Hungria, 1936), que acabou se tornando um mantra dentro da Intel: “*Bad companies are destroyed by crises; good companies survive them; great companies are improved by them.*”<sup>8</sup>



Figura 10.4: Chaveiro do Pentium.<sup>9</sup>

## 10.6 Wanderer – 1993

O Wanderer ou World Wide Web Wanderer foi um web crawler desenvolvido em Perl em 1993 por Matthew Gray, que dentre outras coisas, foi a pessoa que instalou o site do Instituto de Tecnologia de Massachusetts, o MIT: [www.mit.edu](http://www.mit.edu). O Wanderer foi criado com o propósito de analisar o tamanho da World Wide Web, encontrando seus sites e páginas ao redor do mundo e construindo uma lista destes, a chamada Wandex.

Ele é considerado o primeiro robô de buscas, ou *spider*, da WWW, ou seja, um mecanismo que automaticamente encontra páginas da

Web. O Archie e seus clones são mais antigos, mas eles faziam buscas de arquivos em computadores ligados na Internet, mas não na Web propriamente dita, até porque na época do lançamento do Archie a Web ainda era muito restrita.

O Wanderer poderia ter sido o primeiro mecanismo de busca da Web, mas seu autor não tinha essa intenção. Ele foi usado para mensurar o tamanho da Web até 1996. Ele provavelmente não era capaz de acessar todas as páginas da Web, mas rodava de forma sistemática e produzia dados consistentes.

Naquela época, em 1993, o número total de sites da Web não passava muito de uma centena. Assim, o objetivo inicial do sistema de Gray era realmente descobrir novos sites, mas com o crescimento vertiginoso da quantidade de sites, ele passou apenas a coletar dados quantitativos sobre a Web. O relatório do Wanderer, publicado em 1996, apresentava os dados mostrados na [Tabela 10.1](#).

**Tabela 10.1: Relatório do Wanderer<sup>10</sup>**

Mês	Nº de sites	% sites .com
06/93	130	1,5
12/93	623	4,6
06/94	2.738	13,5
12/94	10.022	18,3
06/95	23.500	31,3
01/96	100.000	50,0

O Wanderer chegou a causar alguma confusão no início porque, devido a um problema em sua programação, podia acessar a mesma página centenas de vezes no mesmo dia e como a velocidade da Internet não era lá essas coisas nessa época, isso causava perda de velocidade na rede.

## 10.7 MP3 – 1993

Nos anos 1980, armazenar e transmitir músicas em formato digital era um problema. Um típico computador doméstico não poderia conter mais do que um único álbum musical em seu disco rígido. As velocidades de transmissão da internet nessa época faziam com que a transmissão de um álbum levasse um dia inteiro ou mais. Não faltaram pesquisas com o objetivo de compactar arquivos de áudio de forma que pudessem ser manipulados de maneira mais eficiente.

O principal modelo de compressão de áudio digital, e hoje padrão internacional, foi o MP3 ou MPEG Audio Layer III. Esse padrão faz uso de uma característica da audição humana que consiste em deixar de ouvir certas frequências sonoras quando outras frequências são tocadas simultaneamente. O fenômeno já era conhecido desde o século XIX e continuou sendo estudado e detalhado ao longo do século XX.

A ideia de usar esse fenômeno para codificar arquivos de som de forma mais eficiente aparentemente surgiu de forma independente em 1979 na Bell Labs, com Manfred R. Schroeder (Alemanha, 1926-2009), e no MIT, com Michael A. Krasner, que defendeu uma tese de doutorado com este tema.

Inicialmente os estudos se referiam apenas a compactação de arquivos de áudio contendo voz, sem nenhuma ênfase ainda em termos de codificação de músicas.

Em 1988, a ISO (International Standards Organization) estabeleceu um subcomitê para cuidar da padronização da representação de áudio e vídeo digitais. Esse grupo envolveu vários pesquisadores, inclusive do instituto Fraunhofer da Alemanha e da Bell Labs.

Um dos membros do Fraunhofer foi Karlheinz Brandenburg (Alemanha, 1954), que realizou um doutorado e pós-doutorado nos anos 1980 sobre compressão de músicas digitais. Brandenburg foi um dos principais responsáveis pela formulação do padrão MP3 em sua forma final. Porém, ele próprio não aceita o título de “pai do MP3”, afirmando que foi um trabalho de equipe que envolveu dezenas de pessoas.

A primeira música que foi exaustivamente testada para refinar o padrão foi “Tom’s Diner” de Suzanne Vega (Estados Unidos, 1959). Pelas características da voz da cantora e a forma como a música foi produzida, ela era extremamente difícil de compactar sem que ruído fosse adicionado. No final, após ouvir a canção milhares de vezes e ajustar o algoritmo de compressão a cada vez, Brandenburg conseguiu que fosse reproduzida sem nenhuma distorção. Ele afirmou, posteriormente, que apesar de tudo ainda gostava da música. Inclusive Suzane Vega foi convidada anos mais tarde para, juntamente com Brandenburg, participar de um evento em Cannes para marcar a criação do MP3.

MP3, claro, não se chamava assim até ser adotado pelo MPEG em 1993. O formato competiu com vários outros candidatos e venceu. Ele era capaz de condensar arquivos de música para menos de 1 décimo de seu tamanho sem que nenhuma perda de qualidade fosse perceptível.

Com o crescimento da Web, o MP3 rapidamente tornou-se viral. Literalmente milhões de arquivos nesse formato começaram a viajar pela Internet de um computador para outro, para o prazer dos fãs de música e o desespero das gravadoras, que precisaram começar a repensar seus modelos de negócio. Uma dessas mudanças foi a disponibilização legal de música com direitos autorais a partir do pagamento de pequenas quantias. Esses serviços, conhecidos como “podcasting”, se tornaram usuais. A Apple foi uma das empresas que faturou muito com o sistema de podcasting após o lançamento do iPod a partir de 2001.

O Instituto Fraunhofer, que é detentor da patente do MP3, recebeu muitos milhões de dólares em *royalties* devido ao grande uso desse formato pela Internet. Brandenburg não ficou bilionário com isso, mas a lei alemã garante ao pesquisador uma fração do lucro obtido com patentes para as quais tenha contribuído. Ele só não diz a ninguém quanto é.

Atualmente Brandenburg é diretor do Instituto Fraunhofer para tecnologia de mídia digital e continua realizando pesquisas e contribuindo para a área, por exemplo, na análise de gostos musicais a partir de amostras de músicas que as pessoas ouvem.

## 10.8 Yahoo! – 1994

Yahoo! (com o ponto de exclamação) foi inicialmente um site criado por dois estudantes de Stanford chamados David Filo (Estados Unidos, 1966) e Jerry Yang (Taiwan, 1968), com a lista de seus sites favoritos na Internet. O site se chamava “Jerry and David’s Guide to the World Wide Web”.

Eles haviam baixado o Mosaic logo depois de ter sido disponibilizado e desenvolveram uma verdadeira fixação pela Web. O fato de que o orientador deles estava em licença sabática e, assim eles não tinham tanta cobrança para fazer sua pesquisa “séria”, ajudou bastante. Assim, os dois se embrenharam na rede. Filo chegou a comentar que eles queriam procrastinar suas dissertações e o trabalho na Web era a distração ideal.

O trabalho não foi de início muito cooperativo: eles competiam para ver quem descobria os sites mais interessantes. Porém, logo juntaram esforços e combinaram suas listas. Com o passar do tempo as pessoas começaram a enviar a eles e-mails sugerindo novos sites.

Para a escolha do nome, era moda na época escolher nomes como “*yet another...*”<sup>11</sup>. Eles escolheram então o acrônimo Yahoo! e criaram uma sigla que encaixava nele, ficando assim “Yet Another Hierarchical, Officious Oracle”<sup>12</sup>. O site originalmente era <http://akebono.stanford.edu/yahoo>, já fora de uso.

Naquela época, em 1994, ainda havia relativamente poucos sites na Web e motores de busca ainda não eram difundidos. Assim, o Yahoo! acabou sendo para muitos o portal de entrada da Internet, mais ou menos como o Google é hoje.

De fato, um detalhe que aumentou ainda mais o número de usuários do Yahoo! foi que o Netscape, o novo browser que substituiu o Mosaic na preferência do público, resolveu, talvez por falta de opção, colocar um link para o Yahoo no botão “Diretório”. Assim, ele se tornou o site de busca oficial do navegador que estava começando a ganhar o mundo. Isso fez o fluxo de acessos aumentar muito. Antes do final de 1994, o Yahoo! já tinha ultrapassado 1 milhão de acessos. Com isso a Universidade de Stanford, que até então tinha sido bastante generosa em permitir o uso de seus servidores para o projeto dos alunos, pediu a eles gentilmente que procurassem outro servidor. O servidor do Yahoo! foi então transferido para a Netscape e Filo e Yang deixaram a universidade para se tornar empreendedores, embora ainda não tivessem certeza sobre como fariam dinheiro com isso.

Inicialmente, o Yahoo! não implementava a busca na Web por palavra-chave. O site era organizado em listas hierárquicas nas quais um usuário podia procurar o assunto que lhe interessava e verificar quais páginas havia sobre ele. O trabalho de encontrar e catalogar sites era em grande parte feito manualmente pelos funcionários da empresa.

Na medida que a quantidade de páginas e índices aumentou, foi implementada uma ferramenta de busca. Ela não efetuava a busca na Web, mas nas tabelas que o Yahoo! já tinha registradas. A Yahoo! na virada do século ainda não tinha desenvolvido seu próprio Web crawler. Em 2000, eles fizeram um acordo de mutua divulgação com a Google de forma que um site referenciasse e usasse serviços do outro. Na época, o Yahoo! era o líder em termos de buscas na mente das pessoas, mas a Google tinha o melhor Web crawler de todos com mais do que o dobro de páginas indexadas do que os concorrentes. Como resultado, a popularidade da Google cresceu astronomicamente, tanto que ela repassou a Yahoo! como líder nas páginas de busca por informação na Web. Em 2003 a Yahoo! resolveu criar o próprio Web crawler a partir dos sistemas de outras empresas que ela já tinha adquirido.

O Yahoo! foi possivelmente uma das primeiras empresas da Internet a faturar quase que exclusivamente com propaganda. Consta que quando Mike Moritz (Reino Unido, 1954), representando um dos investidores, visitou o alojamento dos rapazes em Stanford e se deparou com o quarto que era exatamente aquilo que nenhuma mãe jamais admitiria que seu filho tivesse, com pilhas de caixas de pizza espalhadas e vários servidores zumbindo pelos cantos ele perguntou :“E aí? Quanto vão cobrar dos usuários pela assinatura?” Por sorte, já tinham decidido que a ferramenta seria gratuita para os usuários, pois temiam perder a grande quantidade de acessos caso empresas como a Microsoft ou a IBM resolvessem criar seus próprios sites de busca. Se a TV era grátis e o rádio era grátis, e ambos viviam de propaganda, por que a Internet não podia ser assim?

A empresa de Moritz, a Sequoia, investiu então 1 milhão de dólares na Yahoo! em abril de 1994. Por volta de 1999, esse investimento já valia 8 bilhões. No pico da bolha chegou a 30 bilhões.

Porém, a Yahoo! não tinha sequer uma patente para se proteger da concorrência. O que aconteceria se uma empresa qualquer copiasse seu diretório? Houve uma aproximação da AOL (America Online), que ofereceu 2 milhões de dólares pela Yahoo!. Os rapazes acharam a proposta muito baixa e recusaram, ao que a AOL respondeu que esmagaria a empresa deles em dois meses. Eles permaneceram firmes acreditando, e felizmente tendo razão, que o que os diferenciava das outras empresas era o fato de terem sido os primeiros. A confiança do usuário, aliada ao pioneirismo e a um bom atendimento, têm historicamente sido cruciais para empresas que vivem efetivamente de imagem permanecerem no mercado.

Em 1995, Yang decidiu que a Yahoo! seria a primeira marca forte da Internet. Eles fizeram algo considerado radical para esse tipo de empresa na época: anúncios na TV e no rádio. Assim, além dos milhões de usuários da internet, e das dezenas de milhões que estavam entrando na rede, eles se tornaram conhecidos também

pelas centenas de milhões que nunca tinham usado um computador na vida.

A Yahoo! foi uma das poucas companhias a sobreviver ao estouro da bolha das .com na virada do século. Hoje em dia, ela ainda é a segunda maior empresa de buscas na Internet, perdendo apenas para a Google, que tem cerca de 80% do mercado. Sua sede é mostrada na [Figura 10.5](#).



Figura 10.5: Sede da Yahoo!<sup>13</sup>

## 10.9 Simon Personal Communicator – 1994

O Simon Personal Communicator era um telefone celular desenvolvido em parceria pela IBM e a BellSouth Celular. Além da função normal de telefone, possuía uma tela touchscreen monocromática com ícones que podiam ser tocados por um apontador, os quais permitiam acessar aplicativos como e-mail, calculadora, calendário, relógio e o jogo *Scramble* (quebra-cabeça de oito). Além disso, podia ser acoplado a uma câmera, um tocador de música e mapas eletrônicos. Era capaz de receber fax e também funcionava como pager, ou seja, recebia mensagens codificadas de uma central de recados e as exibia na tela. Quando você digitava

um texto nele, o equipamento já tentava adivinhar as letras que se seguiriam e as sugeria.

Como classificar esse aparelho senão como um smartphone? Ele é considerado o primeiro, bem à frente de seu tempo, visto que os smartphones mais parecidos com os que temos hoje surgiram apenas na década seguinte.

O Simon era vendido a 899 dólares com um plano de telefonia de dois anos e 1099 sem este plano. Cerca de 50 mil foram vendidos na época. A [Figura 10.6](#) mostra o Simon colocado sobre seu carregador de baterias.

O Simon, porém, estava muito à frente de seu tempo. Para que ele se tornasse o sucesso que foram os iPhones anos mais tarde faltavam ainda algumas tecnologias. Entre elas, bateria: com uso intensivo de dados, ela não durava mais de 60 minutos, e em uma área com cobertura de sinal fraca, que era a realidade da maioria dos lugares, 30. Além disso, as redes telefônicas da época não tinham sido feitas para comunicar dados, mas voz. Assim, era normal conexões muito lentas ou que ficavam caindo com frequência.



**Figura 10.6: Simon Personal Communicator.<sup>14</sup>**

Mais ainda: esse foi um período em que a IBM amargava grandes perdas. Entre 1991 e 1993, a empresa teria perdido 16 bilhões de dólares e 100 mil empregos. Assim, as dificuldades do mercado de computadores, aliados à ausência de tecnologias necessárias para que o Simon funcionasse plenamente, conspiraram contra ele e o produto acabou sendo abandonado após um curto período de seis

meses de no mercado. Foi uma pena, porque a segunda geração do Simon pretendia criar um dispositivo aproximadamente do mesmo tamanho de um iPhone – o qual acabaria sendo lançado só em 2007, 13 anos depois.

## 10.10 Netscape – 1994

Se o Mosaic foi o browser que trouxe a Web para o grande público, o Netscape Navigator, que tomou seu lugar a partir de 1994, certamente foi sinônimo de browser nos anos 1990. A [Figura 10.7](#) mostra a sua tela de abertura.

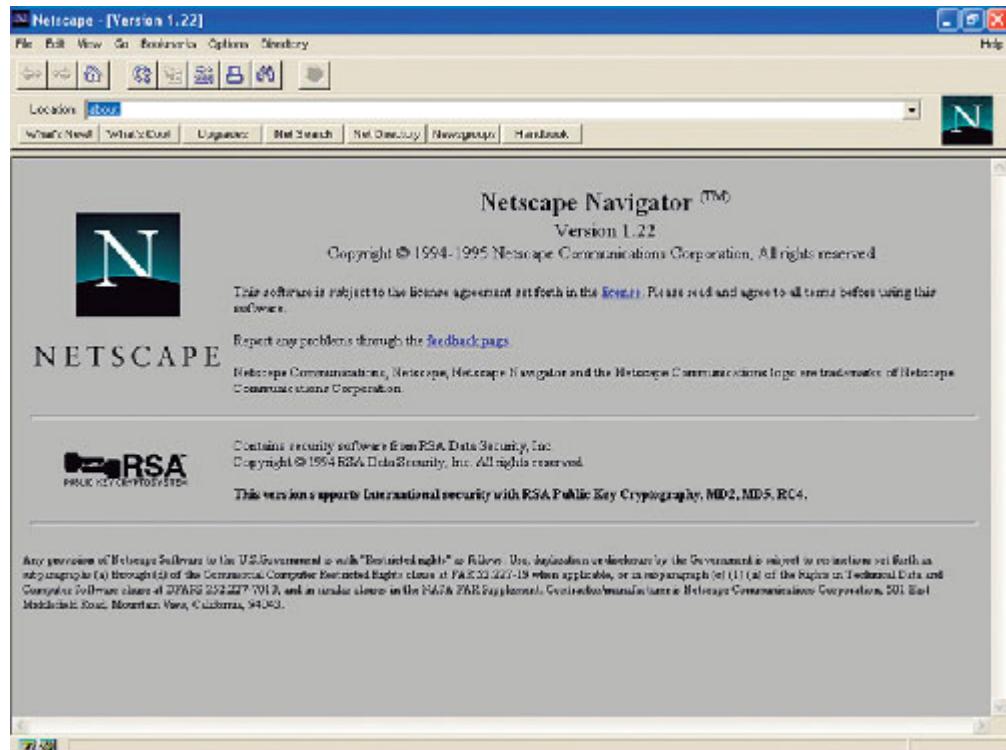


Figura 10.7: Netscape Navigator.<sup>15</sup>

Há uma certa confusão em relação ao nome do navegador no início da história da Netscape porque o primeiro nome da empresa era Mosaic Communications Corporation, mas essa não era a mesma empresa conhecida como NCSA Mosaic. Jim Clark, fundador da

nova companhia, contratou Marc Andreessen, um dos fundadores da NCSA Mosaic, e outros membros da NCSA Mosaic para trabalhar na Mosaic Communications Corporation em um browser que seria chamado Mosaic Netscape.

Na verdade, o contato inicial não foi para o desenvolvimento do navegador e sim para a produção de um jogo online para Nintendo 64. Como o console demorou para ser lançado, eles desistiram do projeto e resolveram lançar um navegador.

Mais tarde, para evitar problemas com a NCSA, a empresa trocou seu nome para Netscape Communications Corporation e seu browser foi rebatizado como Netscape Navigator.

Em 1995 o valor de mercado na Netscape já chegava à casa dos 3 bilhões de dólares. Isso, claro, chamava a atenção da Microsoft, que corria por fora para entrar no mercado da Internet com o lançamento do Windows 95 e seu próprio browser, o Internet Explorer. O Explorer tinha todas as características do Netscape. Como resposta, para se manter no mercado, a Netscape lançou em 1996 o Netscape Communicator, que além do tradicional browser tinha um cliente USENET, um editor de Web, um cliente de e-mail e um livro de endereços.

Assim, a briga entre as duas empresas, conhecida como “guerra dos browsers”, foi uma das maiores confrontações da área nos anos 1990. Consta que no dia do lançamento do Explorer 4.0, os funcionários da Microsoft depositaram um gigantesco logotipo do Internet Explorer bem no gramado da sede da Netscape. Os funcionários da Netscape não deixaram por menos: colocaram sobre o logotipo o dragão Mozilla, mascote da empresa, com um cartão que dizia “Netscape 72 Microsoft 18”, referindo-se à percentagem de mercado de cada browser.<sup>16</sup>

Em 1998, a Netscape resolveu abrir o código fonte do Communicator, iniciando assim a organização Mozilla, que futuramente desenvolveria o Firefox. Nesse ano, o desenvolvimento do Netscape deu uma freada, o que possibilitou à Microsoft a

oportunidade de finalmente passar à frente no mercado. A Netscape acabou vendida para a AOL no mesmo ano por 4,2 bilhões de dólares. Porém, apenas em 2000 uma nova versão do Netscape (a 6) chega ao mercado. Em 2008, a AOL encerra a produção do Netscape para sempre.

Entretanto, a versão que foi introduzida na Fundação Mozilla deu origem ao Mozilla Firefox, um browser ainda importante e bastante usado nos dias de hoje.

## 10.11 Amazon – 1994

Em 1994 a Internet ainda era dominada pela área acadêmica, governamental e militar, embora os sites .com estivessem crescendo gradualmente. Um novo modelo de negócio surgia e algumas empresas estavam arriscando suas cartas nele. Dentre inúmeras iniciativas do tipo, uma merece destaque aqui pela inovação e impacto: a Amazon.com.

Originalmente, a Amazon era uma livraria que vendia livros de papel pela Internet. A empresa tinha um diferencial em relação a outras, que era o fato de não ter praticamente nenhum estoque. Eles possuíam um cadastro de fornecedores e um site através do qual vendiam aos clientes títulos que eles não tinham. Quando havia pedidos de compra, a Amazon adquiria os livros e os repassava ao cliente. Por que isso deu certo? Talvez porque eles tenham sido a primeira empresa a fazer isso. Novamente, o pioneirismo na história da computação paga seus prêmios. Seu fundador, Jeff Bezos (Estados Unidos, 1964), nascido Jeffrey P. Borgensen, alegava que não havia nada na sua empresa que não pudesse ser copiado por outras, mas esse era também o caso do McDonald's, copiado inúmeras vezes e ainda forte no mercado.

Um detalhe que talvez poucos saibam é que nos primeiros cinco anos a Amazon trabalhava sem lucro: ela vendia os livros pelo mesmo preço que comprava. Isso, claro, deixava os investidores doentes, mas depois de se tornar a principal empresa de comércio

eletrônico do mundo, os lucros vieram – e vieram grandes. O primeiro ano em que a Amazon deu lucro foi 2001, mas já foi na casa de 5 milhões para um faturamento na ordem de 1 bilhão. Na mente dos clientes a coisa era simples: você podia procurar por um livro em sites de livrarias e editoras, mas sempre o acharia pelo mesmo preço ou até mais barato no site da Amazon. Então por que ficar procurando?

A empresa, claro, não parou por aí. Hoje a Amazon vende de tudo. Em termos de livros, é cada vez maior o mercado de suas versões eletrônicas, lidas no Kindle, aparelho que ela desenvolveu especialmente para esse fim. Considera-se que hoje a Amazon compete com o Walmart como loja, com a Apple como fabricante de dispositivos e com a IBM como provedora de serviços de dados. O logotipo atual da Amazon.com inclui uma seta ligando a letra “A” à letra “Z”, um sorriso, indicando que ela vende de tudo.

No início, Bezos queria chamá-la de Relentless,<sup>17</sup> o que segundo muitos é um traço marcante de sua própria personalidade. Porém, seus amigos o dissuadiram disso, pois o nome certamente não pegaria bem. Afinal, quem gostaria de se tornar cliente de uma empresa implacável? Bezos já tinha registrado o domínio relentless.com e o mantém até hoje, entretanto. O leitor pode digitar esse endereço no seu browser e será direcionado à página inicial da Amazon.

Ainda assim, quando a empresa foi fundada em 1994 seu nome original era “Cadabra”. Bezos mudou o nome para Amazon no ano seguinte depois que um advogado confundiu o nome com “Cadáver”.

O termo Amazon foi escolhido basicamente por duas razões: a Amazônia é, na visão de Bezos, um lugar exótico e diferente, como ele queria que sua empresa fosse; além disso, a bacia hidrográfica do Amazonas é a maior do mundo, como ele queria que sua empresa fosse. Mais ainda, um nome que inicia com “A” sempre viria primeiro nas listas ordenadas alfabeticamente.

Seguindo a tradição das empresas de tecnologia, a Amazon foi fundada na garagem de Bezos em Bellevue, Washington.

Ele tem um estilo próprio de dirigir a empresa. Mantém um e-mail público e qualquer pessoa pode mandar mensagens diretamente a ele. Dizem que lê todas as mensagens e, caso sejam reclamações, repassa-a à pessoa responsável dentro da empresa com uma única observação: "?". Quando os funcionários recebem um e-mail de Bezos com essa marca, eles reagem como se tivessem recebido um pacote fazendo *tic-tac*. Agem com igual cuidado, fazendo de tudo para resolver o problema o quanto antes e da melhor forma, reportando a solução tanto aos chefes quanto ao cliente.

## 10.12 AltaVista – 1995

Ok, houve outros Web crawlers antes de 1995, mas nenhum se tornou tão popular quanto o AltaVista, em sua época. O programa foi criado pela DEC com o objetivo de facilitar a busca por arquivos e páginas úteis na incipiente World Wide Web. O sistema teve inspiração no Memex de Vannevar Bush. Havia outros buscadores e vários surgiram depois, mas o AltaVista tinha seus trunfos: ele era baseado em um processo de busca extremamente eficiente que conseguia rastrear mais páginas do que se acreditava existir.

O AltaVista foi um sucesso imediato. Já no primeiro dia de operação, teve 300 mil acessos. Na época, o site ainda era subordinado à DEC: [www.atavista.digital.com](http://www.atavista.digital.com). Em 1997, o número de acessos já superava 80 milhões por dia. Neste mesmo ano, os lucros da empresa com patrocínios atingiram 50 milhões de dólares.

Porém, em 1998 a DEC foi vendida para a Compaq e a empresa resolveu transformar o AltaVista de um buscador com interface minimalista em um portal para que competisse diretamente com o Yahoo!. Em 1998 a Compaq pagou 3,3 milhões de dólares pelo domínio Altavista.com, que pertencia a outra empresa, a Altavista Technology Incorporated.

Em 1999, se percebeu que a estratégia do portal não estava dando certo e, ao mesmo tempo, o abandono da ferramenta de busca fez com que ela perdesse terreno para o Google. Em 2001, o AltaVista deveria ter um impulso maior, mas houve o estouro da bolha das .com e os projetos de investimento foram cancelados. Nessa época os escritórios do AltaVista estavam vazios – tão vazios que um funcionário espiritualmente sentou um esqueleto em frente a uma das estações de trabalho desligadas.

Em 2002 a empresa voltou a focar no buscador, mas já era tarde, o da Google era muito mais eficiente. Em 2003 a AltaVista foi comprada pelo Yahoo!. Finalmente, em 2013 o serviço foi finalmente descontinuado. O domínio ainda existe, mas se você digitar no seu browser, ele vai redirecionar para a página do Yahoo!.

## 10.13 Delphi – 1995

Em 1995 a Borland lançou mais um grande hit. Depois do sucesso do Turbo Pascal que, mais do que um compilador já era quase um ambiente de desenvolvimento completo, a empresa lança aquele que seria um ambiente completo de fato, o Delphi.

O ambiente na verdade nunca deveria ter tido este nome. A Borland tinha uma tradição de dar codinomes espirituosos aos seus projetos. Por exemplo, o Quattro Pro, que estava sendo desenvolvido para competir com o Lotus 1, 2, 3, tinha como codinome “Budda”. Quando as pessoas perguntavam “por que Budda?”, explicavam que era porque ele ia assumir a posição de Lótus.<sup>18</sup>

No caso do Delphi, este codinome foi criado porque eles queriam desenvolver um ambiente de desenvolvimento de aplicações baseado em Object Pascal que tivesse forte suporte para banco de dados. Na época, o principal banco de dados era o Oracle e, assim, puxando um pouquinho para o lado da mitologia grega, quando você queria falar com o Oráculo (Oracle) você devia seguir até a cidade de Delfos (Delphi, em inglês).

Entretanto, o nome era considerado retrógrado pela relação com a mitologia e, além disso, podia trazer associações indesejadas porque o Oráculo de Delfos frequentemente dava respostas vagas e indecifráveis. Eles não queriam esse tipo de associação com o produto.

Estava tudo certo para o ambiente se chamar Borland AppBuilder quando fosse lançado, mas, infelizmente (ou não) para a Borland, poucos meses antes de ser introduzido, a Novell lançou um produto chamado Visual AppBuilder. Sem chance de manter o nome como previsto e na falta de um termo melhor, o produto acabou ficando mesmo com o codinome. A piada do Oracle também não era de todo ruim.

Entre os engenheiros a votação do nome não foi unânime. Um deles votou contra “Delphi” e afirmou que a única forma realmente aceitável de escolher seria fazendo uma pesquisa entre os clientes da empresa, inclusive no exterior, pois às vezes nomes ou acrônimos escolhidos em uma língua podem ser palavras desagradáveis em outras. Assim foi feito, e quanto mais se tentava impedir o termo “Delphi”, mais ele grudava nas mentes dos envolvidos com o projeto. E assim ficou.

Delphi foi lançado inicialmente para a plataforma Windows 3.1. O sistema era considerado um RAD ou ambiente para Rapid Application Development.<sup>19</sup> Ele permitia desenvolver aplicações para o Windows. Sempre houve dúvida sobre sua aplicabilidade a sistemas de grande porte, já que suas origens são o Turbo Pascal, um sistema criado para programação de pequenas aplicações. Porém, no ramo das pequenas aplicações não havia dúvida: Delphi era um fenômeno de popularidade.

Outro problema que Delphi teve que enfrentar foi que a Borland demorou para dar suporte ao desenvolvimento de aplicações para Web. Como Java nasceu na mesma época já com essa característica, e à medida que a necessidade desse tipo de

aplicação cresceu, Delphi foi perdendo terreno para Java como linguagem preferida para desenvolvimento de aplicações (Web).

Em 2006, a área de desenvolvimento de ferramentas da Borland foi transformada em uma empresa subsidiária chamada CodeGear, que foi vendida em 2008 para a Embarcadero Technologies. Atualmente, o Embarcadero Delphi é um produto em pleno desenvolvimento e uso. Na tabela TIOBE em fevereiro de 2016, o Delphi/Object Pascal aparecia na 10<sup>a</sup> posição entre as linguagens de programação.

## 10.14 Java – 1995

Outra linguagem de programação que surgiu em 1995 foi Java. O projeto iniciou em 1991 na Sun: inicialmente chamado de Stealth Project, um projeto secreto, depois foi rebatizado como Green Project. O assim chamado Green Team, com 13 pessoas e liderado por James Gosling (Canadá, 1955), iniciou na perspectiva de que a próxima grande tendência em desenvolvimento de sistemas seria a junção de computadores com dispositivos digitais. Não havia ainda linguagens adequadas para esse tipo de desenvolvimento.

Originalmente, a linguagem deveria se chamar GreenTalk, possivelmente uma referência ao Green Team e a Smalltalk, visto que algumas pessoas afirmam que Java foi uma tentativa de criar um C++ mais parecido com Smalltalk.

Depois seu nome foi mudado para Oak.<sup>20</sup> Essa escolha veio primeiro do fato de que havia um carvalho ao lado do escritório de Gosling, segundo porque essa árvore representa força e estabilidade e é símbolo de muitos países. No entanto, já havia uma Oak Technologies registrada. No final, a linguagem foi rebatizada como Java.

Java não é uma sigla: é um nome, escolhido porque devia ser dinâmico, revolucionário, animado, único, fácil de pronunciar e divertido. Selecionado a partir de um grande conjunto de possibilidades, Java é o nome de uma ilha da Indonésia onde o primeiro café do mundo foi produzido. Essa coisa do café tem

grande apelo entre os programadores, que são conhecidos como grandes consumidores da bebida. Assim, a escolha estava feita. O logotipo de Java até hoje apresenta uma fumegante xícara de café.

A nova linguagem foi demonstrada em um dispositivo interativo portátil projetado para a indústria de TV a cabo. Porém, o conceito de dispositivos móveis ainda levaria muitos anos para se tornar economicamente viável, e portanto a linguagem ainda estava muito à frente de seu tempo.

Ela, porém, também era adequada para desenvolvimento de aplicações na Web. E a Web estava começando e carecia desse tipo de linguagem. HTML era uma linguagem de marcação de textos e era muito limitada; você não conseguiria fazer um “aplicativo” com HTML, apenas páginas estáticas com texto, imagens e links. Havia, assim, um mercado gigantesco e crescente para Java.

Ela foi projetada para minimizar qualquer tipo de dependência com hardware. A filosofia da linguagem é traduzida no acrônimo WORA, ou “Write Once, Run Anywhere”.<sup>21</sup> E isso se aplicava não apenas ao código-fonte, mas também aos programas compilados em Java, que deveriam rodar em qualquer computador sem ter que ser recompilados. Para que isso fosse possível, Java utilizou o conceito de máquina virtual. Um programa compilado em Java não é traduzido para uma linguagem de máquina de um microprocessador específico, como acontece com outras linguagens. O código Java é traduzido para uma linguagem conhecida como Bytecode, uma forma genérica de linguagem de baixo nível independente de processador. Para que ela possa rodar em uma arquitetura específica, é necessário implementar para essa arquitetura uma versão da Máquina Virtual Java. Trata-se de um programa que interpreta um aplicativo em Bytecode e o executa sobre a arquitetura do computador em questão. Dessa forma, um programa Java compilado roda em qualquer computador, desde que ele tenha implementado e esteja rodando uma máquina virtual Java.

Java tem muitas estruturas inspiradas em C++, mas elimina praticamente todas as estruturas de baixo nível, ou seja, aquelas que dão acesso direto ao microprocessador, visto que esse não é o objetivo da linguagem.

Originalmente Java era propriedade da Sun, mas aos poucos a empresa foi liberando versões e partes gratuitas e livres. Hoje é possível fazer muita coisa com Java e suas ferramentas sem pagar nada.

Rapidamente os desenvolvedores de browsers para Web, como a Netscape, começaram a incluir a possibilidade de executar Applets, ou miniaplicações Java, nos sites. Assim, os criadores de páginas da Web poderiam incluir, além de texto e imagens, a possibilidade de se ter verdadeiros programas com entrada de dados, consultas e quaisquer outras características que pudessem ser programadas. De fato, o próprio anúncio de Java em 1995 foi realizado em conjunto por John B. Gage (Estados Unidos, 1942), diretor científico da Sun, e Marc Andreessen, cofundador da Netscape. O anúncio já previa sua incorporação ao Netscape Navigator.

Atualmente, segundo a tabela Tiobe, Java é de longe a linguagem mais popular do mundo e continua crescendo. Um em cada cinco projetos de software no mundo é desenvolvido em Java. Desde 2002 Java e C tem se alternado como as duas linguagens mais populares do mundo.

## 10.15 Wiki – 1995

Um Wiki é um tipo de site que permite edição colaborativa. A Wikipédia, criada em 2001, é o seu exemplo mais conhecido, sendo atualmente uma das páginas mais acessadas do mundo. Porém, o conceito de Wiki é mais antigo e foi criado em 1995 – por Ward G. Cunningham (Estados Unidos, 1949), para ser mais exato.

Um Wiki difere de um Blog, por exemplo, porque não há um líder ou responsável pelo conteúdo produzido. Qualquer pessoa tem o

mesmo direito de escrever, alterar ou comentar qualquer página e qualquer texto.

O sistema originalmente desenvolvido por Cunningham chamava-se WikiWikiWeb, e era descrito como “o mais simples banco de dados online que poderia possivelmente funcionar”. Sua filosofia era um sistema simples e rápido para permitir edição online de páginas Web por pessoas que não fossem programadores. A ideia era chamar o sistema de QuickWeb, ou qualquer coisa parecida, que lembrasse rapidez. Porém, no final ele batizou o sistema como WikiWikiWeb, em função do nome do ônibus que ligava os terminais do Aeroporto de Honolulu, Hawaii ([Figura 10.8](#)). Cunningham lembra de ouvir um funcionário do aeroporto indicando o WikiWiki (“rápidorápido” em havaiano) e achou que a palavra tinha a sonoridade que ele esperava.



**Figura 10.8:** O ônibus WikiWiki.<sup>22</sup>

A primeira aplicação da WikiWikiWeb, ou simplesmente Wiki, foi um repositório de padrões de desenvolvimento de software que Cunningham organizou. O sistema foi programado em Perl.

Cunningham inspirou-se, entre outras coisas, no HyperCard, um sistema de edição de hipermídia lançado em 1987 que ele mesmo desenvolveu para a Apple. O Hypercard usava o conceito de cartões e links, mas era voltado para desenvolver aplicações em hipertexto para serem acessadas por uma única pessoa em um computador sem acesso à Web, que na época não existia. Assim, o Wiki seria o Hypercard da Web.

Wiki é considerado o primeiro sistema a implementar de fato as ideias de Vannevar Bush de 1945 sobre o Memex, ou seja, levou 50 anos para as suas ideias se tornarem realidade.

As interações em uma Wiki são autorreguladas pela própria comunidade, mas isso não significa que sejam sempre tranquilas. Conta-se que em 1998 um grupo de pessoas que começava a discutir o modelo ágil de desenvolvimento de software, conhecido com XP (eXtreme Programming), começou a inundar a Wiki de seus assuntos, o que irritou a parte da comunidade que queria manter a discussão restrita apenas a padrões de software. Assim, várias páginas passaram a exibir o rótulo “XpFreeZone”,<sup>23</sup> que informava sem maiores pudores que quaisquer discussões sobre XP não eram bem-vindas ali.

Em 1999 um desenvolvedor da Microsoft chamado Sam Gentile postou em sua página “I’m through here!”<sup>24</sup> porque sentia nos usuários da Wiki uma postura “antiMicrosoft”. Assim, ele começou sistematicamente a apagar tudo o que tinha postado na Wiki. Outros usuários postaram esses conteúdos de volta, o que levou a discussões éticas sobre o direito de um usuário apagar suas contribuições e o direito de terceiros postarem-nas novamente.

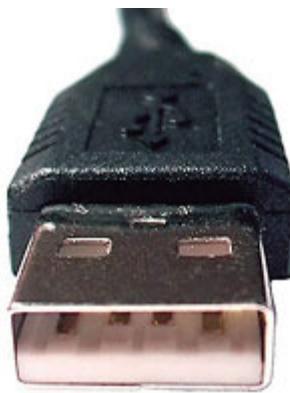
## 10.16 USB – 1995

A famosa porta ou cabo USB (Universal Serial Bus), que tanto usamos hoje, surgiu pela primeira vez em 1995 como resultado de um esforço concentrado de sete empresas para padronizar e universalizar os protocolos e conectores entre computadores e

outros dispositivos. As empresas envolvidas no projeto tinham o poder necessário para estabelecer o padrão independentemente de qualquer organismo de normas. Eram elas: a IBM, Microsoft, Intel, Compaq, DEC, NEC e Nortel.

O USB serve não apenas como meio de comunicação entre o computador e outros dispositivos como impressora, teclado, mouse, câmera, scanner, smartphone etc., mas também como fonte de energia elétrica. É por isso que hoje podemos carregar nossos smartphones via USB em um computador ou, através de um transformador, na rede elétrica.

Ao contrário de outros padrões, como HDMI, que usam conectores iguais nos dois lados do cabo, o USB usa conectores diferentes. Isso foi feito para evitar sobrecargas, pois como o cabo transmite energia elétrica suficiente para carregar um smartphone, uma ligação incorreta de aparelhos poderia causar problemas. Assim, o conector de Tipo A é usado na fonte de energia, digamos, o computador. Já o Tipo B é usando no outro aparelho, usualmente um dispositivo ligado à máquina e que consumirá essa energia. A [Figura 10.9](#) mostra o conector macho do tipo A.



**Figura 10.9: Conector macho USB de tipo A.<sup>25</sup>**

Nos primeiros anos de existência do USB, computadores pessoais começaram a incluir esse tipo de conexão (usualmente duas portas)

na traseira e depois também na sua parte dianteira. Com o passar do tempo, as conexões antigas no formato RS-232 caíram em desuso e a maioria dos computadores hoje é produzido apenas com conexões USB. Em 1998, o iMac G3 foi o primeiro computador a usar apenas portas USB, eliminando todos os padrões mais antigos do seu design.

A primeira versão do USB (1.0) de novembro de 1995 tinha velocidades de até 12 Mbit/s. O padrão USB 2.0, lançado em 2000, já alcançava 480 Mbit/s. Finalmente, a versão 3.0, lançada em 2008 chega à velocidade de 5 Gbit/s.

Em relação ao tamanho do conector USB, existem três tipos: o original que é o que usualmente temos nos computadores e notebooks, o Mini USB, mais usado em câmeras digitais e o Micro USB presente nos smartphones mais modernos.

## 10.17 PHP – 1995

PHP é uma linguagem que sempre esteve associada ao conceito de World Wide Web. Ela surgiu em 1994 e foi disponibilizada pela primeira vez em 1995 por Rasmus Lerdorf (Groenlândia, 1968). A primeira versão era apenas um sistema programado em Perl para que ele pudesse saber quem estava acessando sua página Web. Ele nem imaginava que um dia esse sistema pudesse ser usado por alguém além dele próprio, mas pessoas começaram a pedir para usar cópias dele nos seus sites. Lerdorf cedeu, e como é regra nestes casos, os usuários começaram a pedir por mais funcionalidades.

Quando o sistema foi disponibilizado em 1995 ele foi batizado como PHP, ou seja, Personal Home Page Tools. Consistia de um parser bem simples que entendia alguns comandos e algumas utilidades comuns nas páginas Web da época, como um contador de visitas e um livro de visitas.

Lá pela metade de 1995, Rasmus reescreveu o parser usando a linguagem C, de forma que ele também pudesse interpretar dados

de formulários HTML, e adicionou suporte a um banco de dados baseado em SQL. O pacote passou então a ser conhecido como PHP/FI, no qual FI significava “Form Interpreter”, para referenciar o interpretador de comandos SQL.

A vantagem do uso de PHP em relação a outras tecnologias da época estava no fato de que era possível “embutir” código programável em PHP dentro do código HTML que definia uma página Web. Assim, em vez de se ter programas separados das definições das páginas, a coisa ficava toda junta em um arquivo só, inclusive os comandos SQL que acessavam o banco de dados para exibir esses dados em um site.

A sintaxe de PHP foi inspirada em outras linguagens como C, Perl e Java. É hoje uma tecnologia aberta e funciona na maioria dos browsers. Você já deve ter aberto muitas páginas na Web que terminavam com o sufixo “.php”, o que significa que a definição da página usa código PHP.

Em 1997 o projeto deixou de ser um hobby de Rasmus para ser adotado por uma comunidade de desenvolvedores. Zeev Suraski (Israel, 1976) e Andi Gutmans reescreveram todo o sistema e relançaram como PHP/FI 2. O acrônimo então foi redefinido de forma recursiva, isto é, um acrônimo que usa o próprio acrônimo na sua definição. Assim, hoje, PHP significa: PHP Hypertext Preprocessor.<sup>26</sup>

Pela facilidade de uso e semelhança sintática com linguagens conhecidas, PHP foi sendo cada vez mais usado por desenvolvedores de sites da Web. Em fevereiro de 2016, aparecia em 6º lugar na tabela Tiobe. PHP é, portanto, mais um exemplo de uma iniciativa despretensiosa que acabou virando um grande projeto na área de informática.

## 10.18 Windows 95 – 1995

O Windows 95 foi um grande passo tecnológico para a Microsoft, mas para o público nem tudo aparecia muito claramente. Uma das

grandes inovações que veio para ficar, pelo menos até hoje, foi o menu Start (Iniciar). Para chamar a atenção do público para essa característica, a Microsoft colocou como música tema do Windows 95 “Start me Up”<sup>27</sup> dos Rolling Stones. Recentemente, Danny Oran, ex-funcionário da Microsoft que detém a patente pelo menu Start e pelo Taskbar (barra de ferramentas) disse estar um pouco desapontado por, após tantos anos, essas características ainda estarem no Windows 10. Por outro lado, isso significa que em mais de 20 anos ninguém apareceu com uma ideia melhor.

Além das características mencionadas, o sistema também introduziu o conceito de atalho para arquivos e diretórios, nomes de arquivos com mais de oito caracteres passaram a ser possíveis e o botão direito do mouse recebeu funções.

Oran era psicólogo comportamental, e foi contratado pela Microsoft em 1992 para ajudar a tornar o Windows mais fácil de usar. Entre vários experimentos, consta que uma vez ele colocou um usuário novato defronte ao Windows 3.1 e solicitou que editasse um texto qualquer. Depois de o usuário ficar 20 minutos olhando para a tela sem saber o que fazer, Oran convenceu-se de que o problema não eram os usuários, mas o sistema operacional que era muito difícil de usar. Ao ser questionado pelos engenheiros, teria afirmado: “Nossos usuários são idiotas!” Porém, depois veio a saber que aquele usuário em questão era um engenheiro de propulsão que trabalhava para a Boeing. Se um engenheiro de foguetes não conseguia usar o sistema, quem conseguiria? A falha era do design.

Um dia, ele se deliciou ao ver um usuário completar uma tarefa usando sua criação, o botão “Start” antes mesmo de ele ter terminado de dar as instruções. Apesar do grande progresso que foi obtido com a usabilidade dessa interface, uma contradição sempre a acompanhou: para desligar o computador você tinha que clicar primeiro no botão “Start”.

Outro problema sério do Windows 3.1 era que quando um programa era minimizado, era difícil voltar a ter acesso a ele. Os usuários

simplesmente não sabiam onde procurar. Assim, iam abrindo janelas e mais janelas e só percebiam que a coisa estava saindo do controle quando a máquina se tornava tão lenta que não conseguiam mais fazer nada. Já o Windows 95 tinha a barra de tarefas que mostrava todos os programas que estavam abertos, inclusive os minimizados, na forma de botões facilmente acessíveis.

O Windows 95 estabeleceu um recorde ao vender 7 milhões de cópias nas primeiras cinco semanas. Pessoas fizeram fila nas lojas no dia do lançamento e também no dia anterior.

Tecnicamente, ele veio preparado para 32 bits, embora internamente algumas funções ainda funcionassem com 16 bits para manter compatibilidade com sistemas do Windows 3. Outra inovação bastante alardeada, embora nem sempre funcionasse tão bem quanto se esperava, era a nova tecnologia “plug and play”,<sup>28</sup> ou PnP, que algumas pessoas apelidaram de “plug and pray”.<sup>29</sup> A ideia era que novos dispositivos como impressoras, mouse ou terminais de vídeo não precisariam mais ser “instalados” pelo usuário. Até então, se você comprasse uma nova impressora, teria que instalar no Windows os programas apropriados para ela. Se você não tivesse o disquete ou CD, não conseguiria de forma alguma fazer a impressora funcionar, mas com a tecnologia *plug and play* bastaria conectá-la ao seu computador e ela mesma faria a instalação do software necessário.

A configuração mínima para rodar o Windows 95 era um 386 DX ou um 486, que era mais recomendado. Além disso, era necessário um mínimo de 4 MB de memória principal, embora o recomendado fosse 8.

Um item controverso foi a disponibilização do Internet Explorer como parte integrante do Windows 95. A Netscape acusou a Microsoft, que tinha seus sistemas operacionais em 80% dos computadores do mundo, de concorrência desleal e monopólio. O Departamento de Justiça americano abriu processo contra a Microsoft e em 1999 ela foi condenada a ser dividida em duas empresas: uma para produzir

sistemas operacionais e outra para produzir todos os demais aplicativos. A Microsoft entrou com recurso por considerar a sentença muito severa, e foi ouvida pela corte de apelações. No final, o caso foi resolvido com um acordo através do qual a Microsoft compartilharia parte de sua tecnologia com outras empresas, que poderiam portanto desenvolver sistemas para competir com ela. Vários estados americanos não concordaram com a decisão, mas o caso efetivamente se encerrou aí.

## 10.19 Ruby – 1995

A linguagem Ruby surgiu a partir de discussões de Yukihiro Matsumoto (Japão, 1965), também conhecido como Matz, que desejava uma linguagem de scripts verdadeiramente orientada a objetos que fosse sintaticamente simples, portável e que tivesse outras características que pudessem aumentar a produtividade de quem a usasse. Ele alegava conhecer Perl, mas não gostava dela porque achava que não parecia ser uma linguagem séria. Também conhecia Python, mas ela não era exatamente orientada a objetos. Ele queria uma linguagem com a qual fosse divertido programar e não um pesadelo. Essa linguagem deveria ser primordialmente compreensível por humanos e, em segundo lugar, por máquinas.

Como Matz não encontrou a linguagem ideal, ele resolveu criar uma. Passou vários meses escrevendo um interpretador e em 1995 divulgou seu trabalho em newsgroups no Japão. Para definir a linguagem, buscou inspiração nas melhores estruturas e conceitos de Smalltalk, Perl, Ada, Eiffel e LISP.

Outras versões saíram em 1996 e 1997, mas só em 1998 foi lançada uma verdadeiramente estável, a 1.2. Nesse ano também ele começa a divulgar sua linguagem fora do Japão ao criar uma lista de discussão sobre Ruby em inglês.

Em 1999, Matz e Keiju Ishitsuka publicam o primeiro livro em japonês sobre Ruby, cujo título traduzido para o inglês era *The Object-Oriented Scripting Language Ruby*.<sup>30</sup> Um livro em inglês foi

publicado no ano seguinte: *Programming Ruby*.<sup>31</sup> Nessa época, a linguagem já era mais popular do que Python no Japão e começava a se espalhar pelo mundo. A [Figura 10.10](#) apresenta o logotipo de Ruby criado por Matz. Com o passar do tempo, concursos foram realizados para aprimorar o logotipo, mas todos mantiveram o mesmo desenho base do rubi e a cor vermelha.



**Figura 10.10:** Logotipo de Ruby.<sup>32</sup>

Em 2005, com o framework Ruby on Rails, um ambiente de desenvolvimento rápido de aplicações que permitia uma grande produtividade por parte dos programadores, Ruby fica realmente popular. Esse ambiente é muito usado na comunidade de criadores de aplicações Web. Ele foi criado por David H. Hansson (Dinamarca, 1979) em Ruby, e graças a ele, ela tornou-se uma das linguagens mais usadas hoje no mundo, estando em 11º lugar na tabela Tiobe em fevereiro de 2016.

Ruby foi projetada para seguir o princípio conhecido como POLA, “Principle Of Least Astonishment”,<sup>33</sup> o que significa que as estruturas da linguagem não devem surpreender os programadores. Em muitas linguagens de programação encontramos essas surpresas, como estruturas que funcionam de forma inesperada. Em determinado momento, Matz foi criticado por pessoas que diziam ter programado sempre com outras linguagens e que ficavam surpresas com alguns comportamentos de Ruby, mas ele esclarecia que o princípio só se aplica a quem já é programador Ruby; quem vem de outras culturas de programação certamente vai achar algumas

coisas estranhas. Ele observa que outras linguagens, como C++, continuavam surpreendendo mesmo quem já as programava há muitos anos e era isso o que queria evitar que Ruby fizesse.

Um dos maiores problemas, porém, sempre foi a velocidade final do código implementado, mas cada versão subsequente de Ruby procurava melhorar um ou outro aspecto desse problema.

## 10.20 JavaScript – 1995

O ano de 1995 foi pródigo no surgimento de novas linguagens de programação. JavaScript também foi criada neste ano. Apesar do nome, ela não tem nenhuma relação com a linguagem de programação Java. JavaScript, juntamente com CSS (Cascading Style Sheets<sup>34</sup>) e HTML é considerada uma das tecnologias base da Web. Ela é uma linguagem interpretada e orientada a objetos.

Consta que o interpretador de JavaScript foi desenvolvido em apenas 10 dias por Brendan Eich (Estados Unidos, 1961), que trabalhava na Netscape. Originalmente a linguagem iria se chamar Mocha e depois LiveScript. Ela finalmente foi lançada com o nome de JavaScript para pegar carona na fama de Java, que na época estava se tornando bastante popular. Isso criou alguma confusão com os usuários, que pensavam que ela era algum tipo de extensão de Java. Na verdade, a maior influência de JavaScript foi C++, mas existem semelhanças pois C++ também foi uma forte inspiração para Java.

A maioria dos Web browsers atuais roda JavaScript sem a necessidade de instalar qualquer extensão. Ela é uma linguagem de grande impacto, aparecendo em 9º lugar na tabela Tiobe em fevereiro de 2016. Porém, é voltada para desenvolvedores de sites, ou seja, pessoas cujas habilidades de programação são bastante limitadas, e não para construção de sistemas de grande porte.

## 10.21 eBay – 1995

Outra empresa de comércio eletrônico fundada em 1995 foi a eBay. Ela é conhecida como um site de leilão eletrônico, mas mais do que isso, é uma plataforma de aproximação entre compradores e consumidores. Seu modelo de negócio é claro: não é uma empresa que vende itens, mas uma empresa que conecta pessoas.

A eBay foi fundada na sala de estar de Pierre Omidyar (França, 1967), como parte de seu site pessoal. Na época, o sistema se chamava AuctionWeb. Inicialmente foi pensado como um espaço para que pessoas pudessem anunciar produtos, como nas famosas “vendas de garagem” comuns nos Estados Unidos – nas quais itens usados são colocados à venda, normalmente na frente de casa. Em outros países, como o Brasil, o mais comum é que tais objetos sejam comercializados em brechós ou feiras das pulgas.

Um dos primeiros produtos vendidos foi um apontador laser que não funcionava, por 14 dólares. Quando Omidyar perguntou ao comprador se ele tinha entendido que o apontador estava quebrado e não funcionava, o comprador disse que sim: ele colecionava apontadores laser quebrados.

Inicialmente o site era mais um hobby do que um negócio. Ele nem se importava em cobrar nada das pessoas que compravam e vendiam bens através do seu site. Porém, a coisa teve que profissionalizar quando o provedor de Internet passou a cobrar 250 dólares em vez de 30 dólares por mês devido ao grande fluxo de acessos.

Em 1997, com um contrato para leiloar passagens aéreas, o número de transações multiplicou por 10 em relação ao ano anterior. Nesse ano também surgiu o novo nome, inspirado na empresa de consultoria de Omidyar, a Echo Bay Technology Group. Entretanto, como o domínio echobay.com já estava registrado para uma mineradora de ouro, ele foi abreviado para eBay.com.

Em 1998, Omidyar e Jeffrey Skoll (Canadá, 1965), primeiro presidente da eBay, contrataram Meg C. Whitman (Estados Unidos, 1956), que estudou na famosa Harvard Business School para

profissionalizar o negócio. Meg empregou executivos com 20 anos de experiência em média em empresas como a PepsiCo e Disney. Neste mesmo ano, Omidyar e Skoll tornaram-se bilionários quando abriram a eBay ao mercado de ações. A [Figura 10.11](#) mostra a sede da empresa em San Jose, Califórnia. O logotipo que aparece na foto como “ebaY” foi usado entre 2007 e 2012. Hoje ele é escrito apenas com minúsculas: “ebay”.



**Figura 10.11: Sede da eBay em San Jose, na Califórnia.<sup>35</sup>**

Rapidamente a empresa deixou de ser vista apenas como um site de leilão de itens colecionáveis para subir a mercados bem mais rentáveis. A comissão do eBay sobre as vendas é calculada conforme o tipo de negócio realizado.

O site é organizado em tópicos e permite que potenciais compradores procurem os itens que desejam comprar. Não há nenhum custo para olhar o site. Apenas os vendedores é que depositam uma pequena taxa, não reembolsável, quando anunciam um produto. Depois, dependendo do valor da venda, a comissão final do eBay poderá ser maior.

## 10.22 Palm Pilot – 1996

Em 1996 uma empresa chamada Palm Computing lançou o Pilot, considerado o primeiro PDA (ou Personal Digital Assistant) realmente útil.

A Palm foi fundada em 1992 por Jeff Hawkins (Estados Unidos, 1957). No início, a empresa fornecia software para um PDA chamado Zoomer da Cassio, que foi um fracasso de mercado. Ela também oferecia o software de leitura de escrita à mão para o Apple Newton, outro modelo de PDA lançado em 1993 e que, por ser custoso e nem sempre funcionar muito bem, ficou com fama de ser um caríssimo e frustrante bloco de notas.

Após ver as primeiras tentativas de produzir PDAs falharem, Jeff estava determinado a criar um modelo que realmente funcionasse. A primeira coisa, pensou, é considerar o seu tamanho e peso. Ele carregava consigo um bloco de madeira do tamanho e peso aproximado que ele acreditava que um PDA devia ter, só para se convencer de que realmente era possível.

Em 1995, com o projeto do Pilot pronto e sem dinheiro para colocá-lo em prática, Jeff concordou em deixar a empresa ser vendida para a fabricante de modems U.S. Robotics, que tinha os recursos físicos e financeiros para produzir o Pilot a um custo baixo ([Figura 10.12](#)).



Figura 10.12 PalmPilot fabricado pela U.S. Robotics.<sup>36</sup>

O que diferenciava o Pilot de outros PDAs da época era seu projeto minimalista. Ele não procurou colocar no aparelho o maior número possível de funções, mas sim aquelas que os usuários realmente queriam e as implementou da forma mais eficiente possível, minimizando, por exemplo, o número de ações necessárias para executar qualquer tarefa. Ele, portanto, buscava a forma mais eficiente de apresentar dados na diminuta tela e economizar bateria. As especificações do Pilot eram minimalistas também: tela monocromática de 160x160, CPU Motorola 68000 e 128 ou 512 kB de memória. Porém, o software impressionava pela facilidade de uso. Além de tudo, permitia sincronizar dados do PDA com um computador, evitando transtornos relacionados a uma eventual perda do aparelho ou falta de bateria. Aliás, a bateria podia durar por meses (consistia de duas pilhas AAA).

Outra decisão acertada da empresa foi abrir o código do seu SDK (Software Development Kit), de forma que desenvolvedores independentes pudessem inundar a plataforma do Pilot com aplicativos úteis e interessantes. Até a chegada do iPhone, o Pilot dominava o mercado.

O sistema operacional do Pilot foi desenvolvido por uma empresa especializada chamada Be. Em 1996 ela recusou uma oferta de 125 milhões de dólares da Apple para se tornar a desenvolvedora da nova geração de sistemas operacionais Mac OS. Assim, a Apple acabou comprando a NextStep e, junto com ela veio Steve Jobs, que assim retornava à Apple.

Cinco anos depois, a Be acabou comprada pela própria Palm por meros 11 milhões. Agora, imagine se tivessem sido vendida para a Apple em 1996... Jobs não teria retornado à empresa e a história provavelmente teria sido bem diferente para a Apple, pois atribui-se a Jobs o reerguimento da companhia, que estava em vias de falir.

Porém, quando o Pilot se tornou um hit de mercado, os advogados da indústria de canetas Pilot não ficaram felizes e processaram a Palm. A empresa concordou em rebatizar seu produto como PalmPilot em 1997. Porém, as canetas Pilot ainda não estavam contentes e quando a U.S. Robotics foi comprada pela 3Com em 1997, eles prometeram nunca mais usar o nome "Pilot" novamente em seus produtos.

Entretanto, a Palm aprendeu alguma coisa com os advogados da Pilot e em 1998 processou a Microsoft que precisou rebatizar o seu "PalmPC" para "PocketPC".

No ano 2000, a Palm tinha crescido tanto que foi transformada em empresa independente pela 3Com. Ela sozinha valia mais no mercado de ações do que a Ford e a GM juntas.

Consta que a inauguração da nova gigantesca sede da Palm em San Jose, na Califórnia, foi um evento para poucos, pois chovia muito. O prefeito da cidade cortou a fita inaugural e um pajé abençoou o local no qual tinham sido achados antigos artefatos indígenas durante as escavações. Há, porém, quem diga que o que ele fez mesmo foi amaldiçoar o local. Um ano depois, a bolha das .com estourou, e com ela, a empresa que era vista por todos como a "próxima Microsoft", começou a encolher e perder dinheiro.

A partir de 2006 a companhia, após muitas compras, vendas, unificações e divisões, deixou de produzir PDAs. Ela até tentou investir no mercado de smartphones, mas a competição com a Apple era muito acirrada. Finalmente, em 2010 o que sobrou da Palm foi adquirido pela HP, que estava mais interessada nas patentes da Palm do que propriamente na marca e, de fato, esta desapareceu.

## 10.23 Google – 1996

A nova Microsoft estava para surgir em 1996, e não era a Palm: era a Google. Em 1995, Larry Page (Estados Unidos, 1973) e Sergey Brin (Rússia, 1973) se conheceram quando Sergey estudava em Stanford e Larry apareceu para conhecer a universidade e possivelmente estudar ali. Sergey ficou responsável por mostrar o campus a ele e os dois se tornaram amigos. Logo depois, começaram a trabalhar em um artigo: “The Anatomy of a Large-Scale Hypertextual Web Search Engine”.<sup>37</sup> Terry Winograd (Estados Unidos, 1946), então orientador de doutorado de Page, incentivou a ideia; anos depois Page disse que foi a melhor orientação que ele já recebeu na vida.

Esse artigo se tornou a base para um projeto que eles iniciaram em 1996 para um mecanismo de busca na Internet ao estilo do Altavista, Lycos, Excite e outros. O mecanismo se chamava BackRub e foi instalado nos servidores de Stanford. Após um ano e crescentes acessos, ele começa a sobrecarregar a rede da universidade e é transferido para um domínio comercial em 1997, sendo registrado com o nome de google.com em 15 de setembro.

Mas o que tornava o mecanismo deles tão melhor do que os outros? Acontece que até então os buscadores baseavam a lista de resultados na quantidade de vezes que a palavra-chave buscada aparecia na página em questão. Isso levou muitas pessoas mal-intencionadas a colocarem milhares de palavras (as mais procuradas) em páginas que vendiam drogas ou pornografia, as

quais acabavam aparecendo em primeiro lugar em pesquisas totalmente não relacionadas. Essa característica irritava muito os usuários das ferramentas de busca da época.

Porém, o modelo de busca deles seria diferente: o que iria definir a relevância de uma página não seria apenas o conteúdo dela, mas o número de outras páginas relevantes que tivessem um “link” para ela (*page rank*). Esse modelo se assemelhava ao modelo de citações de trabalhos acadêmicos: até hoje, o que determina o impacto de uma publicação acadêmica é o número de vezes que essa publicação é citada por outras publicações. Assim, apesar de ainda ser possível, com algum trabalho, burlar o *page rank* e exibir páginas não relacionadas, o próprio comportamento dos usuários acaba eliminando essas páginas das pesquisas.

Porém, olhando uma página qualquer, é trivial saber quais links saem dela para outras páginas porque estão na sua própria estrutura; agora, saber quais links vêm na direção dela só é possível se você conhecer o conteúdo de todas as outras páginas que existem na Web. Isso tornava o trabalho de indexação totalmente não trivial, pois a estrutura inteira da Internet precisava ser pesquisada para que se conhecesse o valor de ranking de cada página.

O nome Google foi criado como uma referência (homófono) da palavra “Googol”, que é o nome que os matemáticos dão ao número que é igual a 1 seguido de 100 zeros, ou  $10^{100}$ . Esse número deveria refletir a quantidade inimaginável de informação na Internet, que o mecanismo de busca deles se propunha a descobrir e permitir acesso. O termo foi inventado por uma criança de 8 anos, Milton Sirotta, em 1938, o sobrinho do matemático Edward Kasner (Estados Unidos, 1878-1955). O tio pediu que ele desse um nome para um número muito grande, no caso  $10^{100}$ . Só para se ter uma ideia, o número de segundos que se passou desde o surgimento do universo há 14 bilhões de anos foi algo em torno de  $4,7 \times 10^{17}$ . O número de milionésimos de segundo que se passou desde a criação

do universo foi de  $4,7 \times 10^{23}$ . Tudo isso ainda está muito longe de um Googol. E por aí vai.

Em 1998, Larry e Sergey lançam o Google Friends Newsletter<sup>38</sup> para informar os usuários sobre novidades do site. Neste mesmo ano, o cofundador da Sun Andy Bechtolsheim (Alemanha, 1955) assina um cheque de 100 mil dólares para a Google Inc., uma empresa que ainda não existia. Assim, no mês seguinte, Larry e Sergey registraram-na na Califórnia e abrem uma conta para poder depositar o cheque.

O primeiro escritório da empresa, como não podia deixar de ser, foi em uma garagem. No caso, a de Susan Wojcicki (Estados Unidos, 1968), uma amiga de Larry, em Menlo Park.

O mecanismo de busca era realmente muito bom. Em dezembro de 1998, a *PC Magazine* reconhece essa qualidade e o elege seu buscador preferido. O Google tinha uma filosofia que ia totalmente na contramão de outras empresas interessadas em lucrar em cima dos usuários: enquanto a maioria dos sites procurava manter o usuário ali o maior tempo possível, o Google oferecia um sistema ultrarrápido no qual você entrava, fazia sua pesquisa, encontrava rapidamente o que precisava e saia. O fato de que o site não prendia o usuário tornou-se uma grande vantagem.

Em 2000, a Google iniciava sua tradicional campanha de trotes de 1º de abril com o anúncio do MentalPlex, um aplicativo que se dizia capaz de ler sua mente e adivinhar a busca que você queria fazer. As instruções do MentalPlex diziam para o usuário remover chapéu e óculos e olhar fixamente para a espiral que rodava no meio da página sem mexer a cabeça. Então devia criar uma imagem mental daquilo que estava buscando e clicar na espiral. O usuário seria então transferido para uma página onde, além dos resultados para a busca “1º de abril”, recebia alguma mensagem aleatória, por exemplo: “Você não pensou com convicção suficiente. Tente bater palmas três vezes enquanto diz ‘eu acredito’, e tente do novo.” O

MentalPlex ainda pode ser acessado em <https://archive.google.com/mentalplex/>.

Em outubro de 2000 é lançado o AdWords, o sistema de anúncio revolucionário que faria a Google construir sua fortuna. Nesse sistema, um usuário qualquer pode pagar um valor por um conjunto de palavras-chave que interessam ao seu produto e dependendo de aspectos dinâmicos gerenciados pela ferramenta, esse anúncio pode ser apresentado para usuários que pesquisam aquela palavra-chave específica. Assim, o anúncio do Google é totalmente direcionado para pessoas que estão efetivamente procurando por um determinado item ou serviço – ou seja, muito mais eficiente do que comerciais em TV, rádio, revistas ou outdoors. Além disso, ele só é cobrado se o usuário efetivamente clicar nele; a sua mera exibição não tem qualquer custo. Finalmente, o anunciante pode estabelecer um valor mensal máximo que está disposto a gastar com propaganda. Todas essas características fizeram desse modelo um verdadeiro sucesso e colocaram bilhões de dólares todo o ano na conta da empresa.

Em 2001, o Google já oferece buscas com resultados específicos para algumas dezenas de línguas. A própria interface da ferramenta podia ser configurada para uma língua preferida, inclusive algumas que não são oficiais. Consta que uma das favoritas naquela época foi a do Chefe Sueco, um personagem do *Muppet Show* que murmurava coisas praticamente incompreensíveis em uma linguagem que vagamente lembrava o sueco. Aparentemente, hoje essa língua já não está mais disponível, mas pode-se optar por “Elmer Fudd”, conhecido no Brasil como “Hortelino Trocaletra” (personagem do desenho do *Pernalonga*, famoso por sua incapacidade de pronunciar a letra “R”), ou ainda a linguagem Klingon (relativa a uma raça de alienígenas guerreiros da franquia *Jornada nas Estrelas*). Também existe hoje a opção da língua Hacker. Entre as línguas sérias, pode-se destacar o Guarani, língua originária da América do Sul e ainda falada por algumas tribos indígenas no Brasil, Argentina e Paraguai.

Em 2004 a empresa, já com 800 funcionários, muda sua sede para o GooglePlex. O nome é novamente uma referência matemática, visto que um GoogolPlex seria o nome dado ao número que é igual a  $10^{\text{googol}}$ , ou seja, 1 seguido de googol zeros.

Neste ano também foram lançados o Orkut, importante rede social que foi dominante no Brasil por anos e desbancada apenas pelo surgimento do Facebook, e o Gmail, hoje com cerca de 1 bilhão de usuários.

A Google desde o início e até hoje é extremamente pródiga em lançar produtos inovadores, como Maps, Earth, Transit, Acadêmico, Livros, Agenda, Tradutor, StreetView etc. Possivelmente uma das razões de tal criatividade seja o fato de que a Google incentiva seus funcionários a apresentar projetos ou ideias que possam ser transformados em produtos da empresa.

Enfim, assim como a IBM fez nos primeiros anos da computação e a Microsoft depois dela, hoje a Google é a grande pioneira e desenvolvedora de novas tecnologias, das quais ainda falaremos adiante.

## 10.24 Deep Blue – 1997

Desde os primórdios da computação, com os primeiros computadores sendo chamados de “cérebros eletrônicos” ou “cérebros gigantes”, os cientistas efetivamente quiseram comparar as capacidades de raciocínio automático realizáveis pela máquina com as de um ser humano.

Uma das maneiras mais populares de realizar esse tipo de comparação era o jogo de xadrez. Jogadores humanos são criativos, jogam com malícia, e embora não sejam capazes de prever todos os desdobramentos de uma jogada, conseguem ver longe o suficiente para escolher movimentos que podem levar à vitória. Computadores podem apenas usar algoritmos para analisar os resultados possíveis de cada movimento das peças e escolher o movimento que parece levar à vitória com maior probabilidade.

O problema com o jogo de xadrez é que a quantidade de movimentos possíveis cresce exponencialmente à medida que você tenta analisar mais e mais jogadas futuras. Em média, existem 35 possíveis jogadas a cada instante em uma partida de xadrez. Então se você quiser analisar todas as possibilidades de movimentos duas jogadas à frente, deverá analisar em média  $35^2$ , ou 1225 situações. Três jogadas à frente são  $35^3$ , ou 42.875 situações. Para 10 jogadas à frente são  $35^{10}$ , ou aproximadamente 2,758 quadrilhões de situações. Digamos que o seu computador consiga efetivamente analisar 10 jogadas à frente em, digamos, 10 minutos. Com o crescimento exponencial na ordem de 35 vezes, para conseguir analisar 11 jogadas à frente no mesmo tempo seria necessário um computador 35 vezes mais rápido. Para 12 jogadas à frente, ele teria que ser 35 vezes mais rápido que este último, e assim por diante.

Devido a estes números astronômicos, por muito tempo se acreditou que seria muito difícil, senão impossível, um computador ganhar de um grande mestre enxadrista. Entretanto, isso aconteceu em 1997, quando o Deep Blue ([Figura 10.13](#)), um computador para grandes desafios da IBM venceu Gary Kasparov, o campeão mundial de xadrez. Foram seis partidas contabilizando duas vitórias para o Deep Blue, uma para Kasparov e três empates.



Figura 10.13: Deep Blue.<sup>39</sup>

A história de Deep Blue começou em 1985, quando o aluno de mestrado Feng-Hsiung Hsu (Taiwan, 1959), do Carnegie Mellon, iniciou sua dissertação sobre xadrez computacional. Ele propôs uma máquina para jogar xadrez chamada ChipTest, depois seguido pelo Deep Thought,<sup>40</sup> batizado em homenagem ao computador ficcional de *O Guia do Mochileiro das Galáxias*, livro de autoria de Douglas Noel Adams (Reino Unido, 1952-2001), muito popular entre profissionais de computação. Um colega, Murray Campbell, também trabalhou no projeto e em 1989 ambos foram contratados pela IBM. Lá, junto com outros pesquisadores, eles continuaram o que passou a ser conhecido internamente como Deep Blue<sup>41</sup>, uma combinação de “Deep Thought” com a cor oficial dos ternos usados pelos vendedores da IBM, o azul.

Uma primeira versão do sistema foi vencida por Kasparov em 1996. Nessa ocasião, Deep Blue já impressionou por ter vencido uma partida, coisa que nunca tinha acontecido, mas Kasparov venceu outras três e empatou duas.

O jogo de 1997 era uma revanche e dessa vez a máquina, incrementada e informalmente rebatizada como “Deeper Blue”,<sup>42</sup> ganhou. As chances de ela vencer não eram certas, mas a equipe confiava em sua capacidade de analisar mais de 200 milhões de jogadas por segundo. Aos olhos de milhões de espectadores ao vivo, Kasparov venceu a primeira partida e Deep Blue a segunda. As três partidas seguintes foram empates. A partida final foi uma vitória esmagadora de Deep Blue.

Kasparov acusou a IBM de trapacear e exigiou outra revanche, mas a companhia recusou e aposentou a máquina. Há quem diga que um bug na programação do Deep Blue provocou um movimento aleatório na 44<sup>a</sup> jogada da primeira partida. Kasparov teria atribuído esse movimento totalmente inesperado a uma inteligência superior, possivelmente humana, interferindo na máquina. Na partida seguinte, a performance de Kasparov diminuiu, provavelmente devido à ansiedade, e ele perdeu.

Hsu posteriormente propôs a Kasparov que ele construiria uma máquina mais poderosa a partir do projeto do Deep Blue, independentemente da IBM, para a revanche. Desta vez foi Kasparov que recusou.

Entretanto, o legado de Deep Blue não foi apenas a humilhação do campeão humano e a desconfiança em relação à IBM. As novas arquiteturas, algoritmos e o grande poder computacional da máquina abriram caminho para outras pesquisas em modelagem financeira, mineração de dados e dinâmica molecular, ajudando, por exemplo, na criação de novos remédios.

O Deep Blue era um computador paralelo com 30 processadores e centenas de chips especialmente projetados para jogar xadrez. Seu programa foi escrito em C e empregava uma técnica conhecida

como “força bruta”, ou seja, a máquina simplesmente analisava o máximo de jogadas possível no tempo regulamentar e selecionava a mais promissora. Em média, ela conseguia avaliar de seis a oito jogadas à frente, mas em algumas situações conseguia avaliar até 20 ou mais.

Atualmente o Deep Blue está no museu Smithsonian em Washington, mas sua arquitetura altamente paralela foi reutilizada em várias outras máquinas da empresa nos anos seguintes e até hoje.

Em relação ao jogo de xadrez, hoje as pesquisas se concentram mais em software inteligente e não tanto em hardware. Nunca mais um computador como ele foi construído para jogar xadrez. Os atuais campeões computacionais são sistemas de software bem mais sofisticados do que os de 1997, mas que rodam em simples PCs.

## 10.25 Blog – 1997

O Blog é um fenômeno recente e cativante. Basicamente são diários que pessoas ou organizações escrevem e que são alimentados de tempos em tempos com novas informações. O próprio termo “Blog” vem da contração de “Web Log”, ou “diário na rede”, cunhado em 1997 por Jorn Barger (Estados Unidos, 1953), autor do famoso blog “Robot Wisdom”.<sup>43</sup>

Barger foi um dos primeiros nerds de que se tem notícia. Aos 11 anos (1964) ele já programava em um Minivac 601. Estudou matemática e ciências no Ensino Médio e foi várias vezes transferido de escola e de universidade até finalmente não conseguir obter nenhuma graduação. Ele abandonou a carreira em computação para se juntar a uma comunidade hippie por vários anos.

Retornou ao mundo nos anos 1980 para voltar a ser programador até se tornar pesquisador em inteligência artificial na Northwestern University em 1989. Nessa época, ele se interessou em analisar o comportamento humano a partir de simulações computacionais, o

que ele chamou de “Robot Wisdom”. Como a única comunidade da Internet digna desse nome na época era a USENET, Barger se juntou a ela de corpo e alma.

Porém, ele tinha personalidade forte e muita vontade de escrever. Assim, participava de grupos de discussão de todos os assuntos possíveis e no final acabou quebrando os pratos em alguns deles. Após algumas brigas, especialmente com os fãs da cantora Kate Bush (Reino Unido, 1958), ele resolveu não participar mais da USENET e acabou criando um espaço só seu: “Robot Wisdom”.

O conceito era diferente. Não mais se tratava de uma lista de discussão ou espaço para notícias, mas uma página pessoal, só dele, na qual ele podia expor suas ideias para quem quisesse ler. Cada vez que escrevia alguma coisa, o novo texto era colocado no início da página, de forma que os demais fossem movidos um pouco para baixo. Assim, descer na página era como voltar no tempo.

Barger decidiu que precisava dar um novo nome para isso e chamou o sistema de “Logging the Web”, e “Robot Wisdom” foi assim o primeiro blog. Em 17 de dezembro de 1997 a blogosfera foi criada quando seu site foi renomeado para “Robot Wisdom Weblog”.

Quando em 1999 Barger escreveu o “Blogging FAQ”, ele estava lançando as bases filosóficas de toda a cultura blog em pleno florescimento. Nele, Barger reconhecia que embora tivesse criado e divulgado o nome, não inventara o conceito. Ele atribui à página “What’s new” do Mosaic o título de primeiro blog, além de mencionar as iniciativas de Justin Hall (Estados Unidos, 1974) e William Gibson (Estados Unidos, 1948) em 1996, embora eles não usassem o nome nem o conceito de blog, referindo-se às suas páginas apenas como sites pessoais.

“Robot Wisdom” lançou um formato de publicação que foi depois copiado por ferramentas como WordPress, Blogger, Orkut, Facebook, Myspace e Twitter; todos seguem a mesma lógica de publicar e colocar no topo da “pilha” as publicações mais recentes.

Barger, claro, nunca recebeu um centavo por isso e anda sumido da Internet.

## 10.26 Clippy – 1997

Talvez nenhuma outra invenção tenha sido tão odiada na história da computação quanto o Clippy ou Clippit, o assistente automático em forma de clipe de papel usado no Office de 1997 a 2003. Ele costumava aparecer frequentemente se oferecendo para ajudar o usuário a escrever uma carta. Inclusive, dizem, fazia isso mesmo que você estivesse jogando.

O Clippy era comparado a um helicóptero voando baixo: chamava a atenção, ficava bem na sua cara e incomodava muito. O seu problema é que foi otimizado para ser o primeiro contato de um usuário totalmente leigo. Por isso, o usuário experiente logo ficava irritado com suas aparições não solicitadas e sugestões irrelevantes de tão básicas, ao estilo “verifique se o cabo de rede está conectado”.

O Clippy foi testado por grupos de trabalho na Microsoft antes de seu lançamento. Os resultados foram bastante negativos, mas mesmo assim a empresa prosseguiu com ele. As mulheres acharam que o personagem era muito masculino e os homens percorriam as listas de sugestões do Clippy e não encontravam a informação que precisavam. A revista *Time*<sup>44</sup> elegeu o Clippy como uma das 50 piores invenções de todos os tempos, juntamente com o agente laranja, o óleo hidrogenado e o jogo da fazendinha do Facebook (*FarmVille*).

## 10.27 iMac – 1998

Em 1998, os computadores pessoais eram caixas bege retangulares bastante feias e sem-graça. As plataformas PC eram dominantes, no entanto, e a Apple amargava o constante encolhimento do mercado dos seus Macintosh, além de não encontrar rumo para

seus outros produtos. Em 1997, ela estava a apenas seis meses da insolvência e o moral andava muito baixo.

Steve Jobs, que havia há pouco retornado à empresa, assumiu a gestão cortando na carne: projetos pequenos e irrelevantes foram eliminados; a Apple deveria se focar em uma linha de produtos que a colocasse de volta na vanguarda da informática. Jobs acreditava que apenas duas linhas tinham chance de sobrevivência, o Macintosh e o PowerBook, mas ambos sofriam perdas por não terem se concentrado em um mercado adequado.

Jobs apostou em uma ideia que tinha tudo para dar errado: um computador com design avançado, parecendo algo futurista saído do desenho *Os Jetsons*. O computador e todos os seus dispositivos, exceto teclado e mouse, ficariam em um gabinete transparente junto com o monitor CRT e, ao contrário dos PCs da época, o usuário não teria acesso ao interior da máquina para instalar placas ou outros dispositivos. O drive de disquete seria substituído por um drive de CD-ROM, que ainda não era o padrão para distribuição de software devido ao alto custo. A máquina foi toda projetada apostando nas mudanças antevistas para o novo milênio.

Surge assim o iMac, um computador com aparência futurista, construído em um gabinete arredondado e transparente ([Figura 10.14](#)). O design é obra de Jonathan Ive (Reino Unido, 1967). Para diferenciar sua máquina do padrão IBM, cujo slogan era “Think”, a Apple lançou a campanha “Think different”.



Figura 10.14: iMac G3.<sup>45</sup>

Os comerciais do iMac enfatizavam também sua facilidade de conexão com a Internet, coisa que não acontecia com outros computadores, sendo que às vezes usuários comuns precisavam chamar técnicos especializados apenas para ajudar a conectar. Um dos comerciais alegava: “Três passos simples para se conectar na Internet”, o passo 1 era conectar o plugue na tomada, o passo 2 conectar o cabo de Internet e o passo 3, o comercial dizia ironicamente, “não há passo 3”, enquanto mostrava uma panorâmica do iMac. Outro comercial destacava as diversas cores alegres do computador.

Jobs queria que a máquina se chamasse MacMan, mas Ken Segall, empregado da agência de propaganda que cuidava dos comerciais da Apple, não gostou – aliás, ficou horrorizado. Ele sugeriu iMac, no qual o “i” poderia significar “Internet”, mas também individualidade (“i” significa “eu” em inglês) e inovação. O “i” acabou depois disso se tornando marca reconhecida da Apple, com seus iPods, iBooks, iPhones e iPads. Mesmo Steve Jobs acabou sendo nomeado iCEO da Apple.

O iMac original, conhecido como modelo G3, foi o primeiro computador comercial a ser considerado “legacy free”<sup>46</sup> porque ele não tinha mais drive de disquete, apenas de CD, e também só possuía portas USB em detrimento das antigas RS 232.

A maior mudança de design veio só em 2002, com o lançamento do G4. O tubo de imagem CRT foi substituído por um monitor LCD bem mais fino. Assim, a CPU já não ficava mais junto da tela, mas em uma espécie de meia esfera que, como um pedestal, sustentava a tela LCD.



Figura 10.15: iMac G4.<sup>47</sup>

O iMac, especialmente seu design, teve grande impacto na indústria e logo todo tipo de equipamento, de notebooks a afiadores de facas, passaram a adotar designs inovadores na tentativa de repetir o sucesso do iMac. Além disso, ele salvou a Apple da falência e a colocou de volta entre as grandes indústrias de informática no século XXI.

## 10.28 Windows 98 – 1998

Em 1998 a Microsoft lança uma nova versão do Windows chamada Windows 98. Ela era um híbrido de 16/32 bits e ainda era uma casca sobre o antigo MS-DOS, embora para o usuário isso não mais aparecesse explicitamente.

O Windows 98 foi basicamente uma evolução do 95, incluindo algumas características que este último não tinha, como, por exemplo, a hibernação, que é a possibilidade de salvar o estado atual do computador antes de desligá-lo para que, ao ser religado, ele não precise executar toda a sequência de boot novamente e retorne com todos os aplicativos no estado em que foram deixados. Infelizmente essa característica, para funcionar bem, dependia de configurações de hardware e de BIOS que nem sempre estavam de acordo, o que causava problemas e fazia com que os usuários muitas vezes perdessem a confiança na habilidade de hibernação do 98.

Outra característica que diferenciava o 98 do 95 era o suporte que dava a dispositivos secundários que não existiam em 1995, como, por exemplo, o DVD-RAM, disponibilizado pela primeira vez em 1996.

Assim, o 98 foi uma melhoria significativa, mas nenhuma revolução em termos de sistema operacional. O motivo de ele ser mencionado neste capítulo é principalmente por um fato que ocorreu e que, pela “saia justa” que causou, foi muito marcante para a indústria da computação. A apresentação do Windows 98 ocorreu antes que ele estivesse disponível para compra, na COMDEX, ou Computer Dealer's Exhibition<sup>48</sup> em Nevada. Lá, com grande audiência, Bill Gates estava pessoalmente demonstrando a facilidade de uso do 98 em relação aos dispositivos Plug-And-Play (PnP). Porém, quando seu assistente Chris Capossela conectou um scanner USB no computador, este exibiu para uma plateia que riu às lágrimas a famosa BSOD, ou Blue Screen of Death<sup>49</sup> ([Figura 10.16](#)), que é o que aparecia quando o sistema Windows sofria um erro tão grave

que a única opção para evitar corrupção de dados era reiniciar o sistema.<sup>50</sup> Porém, Gates não se saiu mal. Após rir bastante, comentou: “*That must be why we’re not shipping Windows 98 yet.*”<sup>51</sup> Gates foi aplaudido possivelmente por sua presença de espírito. O evento pode ser conferido em vídeo em <https://www.youtube.com/watch?v=eKtGXPfabLQ>.

Se voltarmos uns 170 anos no tempo, veremos que esse conceito de travar um computador quando um erro ocorre foi criado por Charles Babbage, que incluiu no projeto da Máquina Diferencial um sistema de travamento quando as engrenagens paravam em posição inválida.

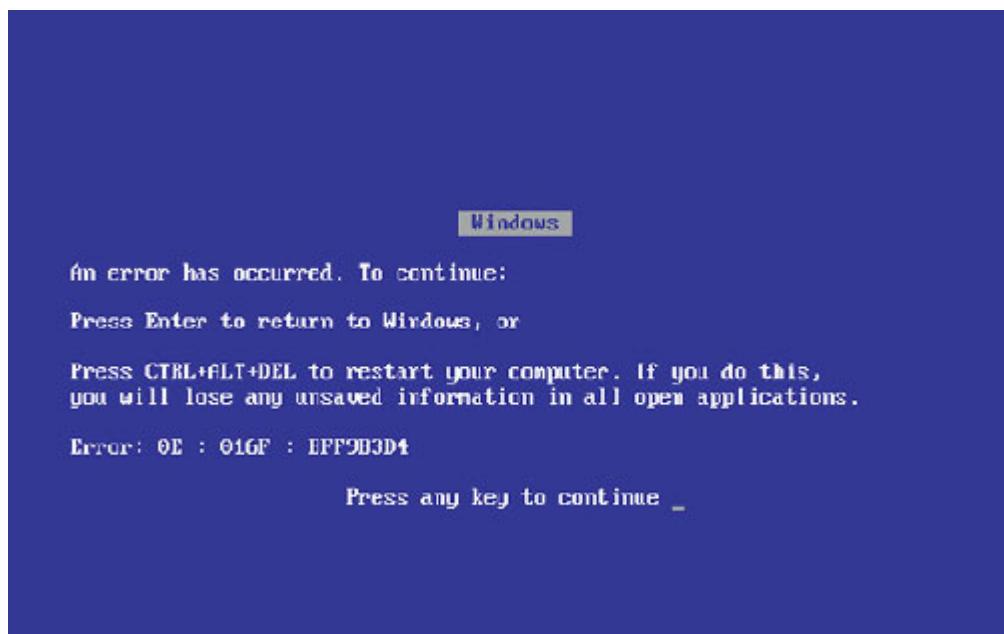


Figura 10.16: A tela azul da morte.<sup>52</sup>

Em 1999, a Microsoft lançou o Windows 98 Second Edition, uma versão bem mais estável do sistema, que também introduzia o conceito de compartilhamento de Internet com o qual um único computador poderia ser ligado à Internet e compartilhar essa conexão com outros. O Windows 98 foi descontinuado em 2002, e suporte ainda foi oferecido até 2006. Seu sucessor oficial é o

Windows ME, embora muitos usuários tenham atualizado diretamente do 98 para o XP, até porque o suporte ao ME foi descontinuado na mesma data que o do 98.

## 10.29 Wi-Fi – 1999

Hoje em dia, quando uma pessoa pergunta “Você tem Wi-Fi?”, isso soa tão natural que nem paramos para pensar que essa tecnologia só foi disponibilizada pela primeira vez em 1999. “Wi-Fi” hoje é praticamente sinônimo de “Internet”, mas até 1999 você só se conectava a ela se fosse por cabo. O mais interessante é que essa tecnologia não foi criada por inventores independentes ou empresas de tecnologia; foi o governo americano, especificamente a FCC (Federal Communications Commission), que definiu o padrão.

Em primeiro lugar é importante entender que para transmitir sinais de radiofrequência, na maioria dos países do mundo, você precisa de uma autorização legal de uma agência governamental. Então como é que podemos ter Wi-Fi em nossas casas se ele é uma rede baseada em radiofrequência? É que a FCC, depois seguida por outras agências mundo a fora, liberou determinadas frequências conhecidas como “banda lixo” para que esse tipo de rede de baixo alcance pudesse ser definido sem maior burocracia.

A decisão de liberar algumas bandas específicas de radiofrequência para uso sem autorização governamental foi feita em 1985 pela FCC. Michael Marcus escolheu três frequências do espectro industrial, científico e médico para serem liberadas ao público.

Essas frequências, de 900 MHz, 2,4 GHz e 5,8 GHz já eram usadas por equipamentos que não eram de comunicação, como por exemplo, fornos micro-ondas. É por isso que as empresas de Internet aconselham você a não instalar seu roteador Wi-Fi perto de um forno desse tipo: ele provavelmente vai interferir no sinal.

Nessa época, porém, pouca coisa aconteceu. Alguns fabricantes se arriscaram a produzir equipamentos de transmissão usando as bandas lixo, mas tiveram pouca receptividade. Como não havia

padronização, poucos se arriscaram nessa tecnologia. Porém, a padronização que havia nas redes a cabo (Ethernet) serviu de inspiração para que um padrão de rede sem fio fosse criado pela IEEE. Vic Hayes (Indonésia, 1941) coordenou o comitê de padronização e por esse motivo é considerado por muitos como o pai do Wi-Fi.

Demorou, entretanto, até que todos os membros do comitê concordassem com um padrão. Apenas em 1997 foram publicados os padrões IEEE 802.11a e IEEE 802.11.b, o primeiro na frequência de 5,8 GHz e o segundo na frequência de 2,4 GHz. A partir de então era possível construir equipamentos compatíveis. Porém, a especificação era exageradamente complexa: consistia de um manual de 400 páginas, demasiadamente complicado para as indústrias, e os problemas de compatibilidade continuaram.

Em 1999, seis empresas da área de tecnologia (Intersil, 3Com, Nokia, Aironet, Symbol e Lucent) criaram a WECA, ou Wireless Ethernet Compatibility Alliance, ou Wi-Fi Alliance ([Figura 10.17](#)), para certificar equipamentos que fossem realmente compatíveis com a norma.



Figura 10.17: Logotipo da Wi-Fi Alliance.<sup>53</sup>

Porém, em termos de nome, “WECA-Compatible” ou “IEEE 802.11b compatible” eram difíceis de pronunciar e não tinham apelo comercial. A tecnologia precisava de um nome mais simples. DragonFly e FlanckSpeed chegaram a ser sugeridos, mas o nome escolhido foi “Wi-Fi” porque era simples e lembrava “Hi-Fi” – termo dado a toda uma gama de equipamentos de som de alta qualidade

compatíveis entre si. A interpretação de Wi-Fi como “Wireless Fidelity” foi definida posteriormente à escolha do nome.

Agora que a tecnologia estava padronizada e tinha um título, faltava um equipamento. A Apple propôs à Lucent que se conseguissem fabricar um adaptador Wi-Fi por menos de 100 dólares eles colocariam um em todos os seus laptops dali por diante. A Lucent conseguiu. Assim surgiu a “AirPort” do iBook e demais equipamentos Apple. Logo os demais fabricantes seguiram o exemplo.

Inicialmente, os servidores Wi-Fi foram utilizados em residências, mas com o passar do tempo, vários tipos de empresas, como cafés, aeroportos, hotéis etc. passaram a disponibilizar Wi-Fi para seus clientes, com o nome de “hotspots”. No começo, isso se dava em troca de taxas de uso, mas, com o passar do tempo e o barateamento das linhas e equipamentos, isso passou a ser cortesia em muitos lugares.

Os padrões originais de Wi-fi permitiam velocidade de 2 Mbit/s. Porém, em 2003 foi lançado o 802.11g que permitia alcançar até 54 Mbit/s na frequência de 2,4 GHz.

## 10.30 Napster – 1999

Em 1999 um garoto de 18 anos chamado Shawn Fanning (Estados Unidos, 1980), juntamente com seu tio, John Fanning (Estados Unidos, 1963), e Sean Parker (Estados Unidos, 1979) criou um serviço de compartilhamento de arquivos mp3 na Internet chamado Napster. Após 60 horas de programação, Shawn tinha criado um buscador de arquivos de música na Internet, um sistema de compartilhamento de arquivos e um sistema de mensagens para facilitar a comunicação entre os usuários.

O nome Napster veio do apelido que Shawn usava em uma rede de hackers, a w00w00. O apelido foi dado por um falastrão que ficou gozando do cabelo dele em um jogo de basquete. Como bom hacker, Shawn estava acostumado a compartilhar arquivos com

outros. Quando queria ouvir uma música, ele buscava nos canais que conhecia. Porém, estava cansado desse processo; era difícil saber onde havia arquivos realmente bons e quando eram atualizados. Pior ainda, os downloads na época eram muito lentos e as conexões caiam frequentemente, obrigando o usuário a começar tudo de novo, mesmo que já estivesse há muitas horas baixando o arquivo. Assim, Shawn resolveu buscar uma solução para facilitar o compartilhamento de músicas na Internet e criou o Napster.

O diferencial do Napster em relação a outros serviços de compartilhamento de arquivos na Internet era o fato de que ele era dedicado a músicas. Isso foi tanto o que o levou às alturas quanto o que o derrubou definitivamente pouco tempo depois.

A tecnologia do Napster não criou o conceito de peer-to-peer, ou P2P, mas fez dele um conceito importante. Com essa tecnologia, em vez de se ter arquivos em um servidor centralizado, eles ficam nos computadores dos usuários. No peer-to-peer, eventuais servidores, se existirem, apenas mantêm tabelas com a localização destes arquivos.

Em seu pico no início de 2001, o Napster chegou a ter 70 milhões de usuários. Ele inclusive chegou a aparecer no Livro dos Recordes como o negócio que mais rápido cresceu em todos os tempos.

Entretanto, o que era disponibilizado no Napster ficava totalmente a critério dos usuários e logo arquivos com músicas comerciais começaram a ser disponibilizadas, inclusive, em alguns casos, meses antes de seu lançamento no circuito comercial. Comenta-se que o Napster foi o primeiro serviço a liberar totalmente o acesso a música. Seus criadores chegaram a consultar um advogado sobre a questão, mas ele analisou o caso de um ponto de vista otimista em relação à indústria de músicas – e não tardou a terem problemas.

Escolas e universidades em 2000 chegaram a bloquear o Napster em suas redes porque o fluxo de músicas de um lado para o outro ocupava a maior parte da banda da rede, sobrecarregando os demais serviços.

Neste mesmo ano, a banda Metallica descobriu que uma de suas músicas que ainda não havia sido lançada, "I Disappear", já estava sendo compartilhada no Napster. Em função disso, ela chegou até a tocar em estações de rádio antes do lançamento, e sem nenhum direito autoral repassado. Pior: todo o repertório da banda estava disponibilizado sem autorização. Isso, evidentemente, levou a um processo judicial por quebra de direitos autorais em março de 2000. Um mês depois, o rapper Dr. Dre (Estados Unidos, 1965) abriu um processo semelhante porque o Napster recusou seu pedido por escrito de remover seu material da rede. Esses processos foram resolvidos após o fechamento do serviço em 2001, desfecho de fato causado pela ação judicial movida pela Recording Industry Association of America (RIAA) em dezembro de 1999.

Durante o processo, o Napster chegou a informar à corte que tinha conseguido um mecanismo que poderia bloquear 99,4% de compartilhamento de materiais protegidos por direito autoral. Porém, o juiz alegou que não era suficiente; o único valor aceitável seria 100%. O Napster argumentou que se 99,4% não é suficiente então a briga não era contra a quebra de direitos autorais, mas contra a tecnologia de compartilhamento de arquivos. E, assim, o Napster teve que fechar as portas. Em julho de 2001, após um crescimento meteórico, o Napster termina abruptamente.

Muitas pessoas, porém, argumentam que divulgar material protegido por direito autoral não é tão ruim. Para as grandes gravadoras e músicos do Top 10, certamente é um pesadelo, mas para pequenas bandas, como foi provado logo depois, o Napster deu uma visibilidade que elas não teriam sem a estrutura das grandes gravadoras. Uma banda praticamente desconhecida até então, Radiohead, teve as músicas do seu álbum *Kid A* divulgadas pelo Napster meses antes do lançamento. Quando o CD foi finalmente lançado, ele se tornou um dos mais vendidos no mundo.

Após a sentença e o fechamento do Napster, a empresa foi comprada pela Roxio, que rebatizou o seu serviço de música online PressPlay para Napster 2.0.

## 10.31 Flash Drive – 1999

Vimos que em 1998 a Apple apostou no USB como o futuro da conexão entre computador e periféricos. Além disso, o iMac ainda tinha uma unidade de CD-ROM no lugar dos disquetes, o que também foi considerado um movimento ousado. Porém, o que talvez ninguém soubesse na época é que mesmo as tecnologias de CD e DVD-ROM já estavam com os dias contados, porque em 1999 foi criado o primeiro Flash Drive ou Pendrive, uma unidade de armazenamento de dados muito menor do que um CD e facilmente conectável a uma porta USB.

Na verdade, 1999 foi o ano da guerra de patentes, pois várias empresas alegaram quase que ao mesmo tempo terem inventado o Flash Drive. A primeira companhia a registrar patente nos EUA, em abril de 1999, foi a israelense M-Systems. Porém, o produto descrito por eles incluía um cabo flexível entre o circuito de memória propriamente dito e o conector USB. Um pouco mais tarde, no mesmo ano, a IBM registra uma patente que era exatamente o flash drive em uma única peça, como o conhecemos. A IBM e a M-Systems fizeram uma parceria para colocar o produto no mercado. Entretanto, outras empresas alegavam ter inventado o mesmo dispositivo, entre elas a chinesa Netac e a singapuriana Trek. As duas também processaram outras companhias que desenvolviam flash drives sem pagar *royalties* a elas.

O primeiro flash drive comercial, mostrado na [Figura 10.18](#), foi o DiskOnKey de 8 MB fabricado pela M-Systems e vendido sob a marca da IBM a partir de dezembro de 2000. Concomitantemente, a Trek desenvolveu e passou a vender o ThumbDrive.



**Figura 10.18: DiskOnKey da IBM.<sup>54</sup>**

Atualmente, Flash Drives não são usados somente para transportar dados de um computador para outro, mas também como extensões da memória, pois hoje são tão rápidos e tem tanta capacidade de armazenamento que podem rodar aplicações completas quando conectados a uma máquina.

Contudo, como qualquer outra tecnologia de armazenamento, os flash drives possuem um número limitado de vezes que podem ser regravados. Quando esse número é excedido ele pode falhar. Discos rígidos, DVDs e outras mídias mecânicas do tipo costumam avisar quando estão para falhar, tanto pelo ruído que passam a emitir, quanto pela possibilidade de apresentarem problemas em um ou outro setor. Porém, quando os flash drives falham, eles o fazem de vez e sem aviso prévio; esse é considerado um de seus pontos fracos.

A sua maior desvantagem, porém, é certamente a facilidade com que são perdidos, por serem tão pequenos. Quem nunca perdeu um pendrive?

## 10.32 Bug do Milênio – 1999

Como vimos, os primeiros computadores começaram a ser construídos na metade do século XX. Suas memórias eram pequenas, o que obrigava os programadores a economizar onde podiam. Uma dessas economias, que não era privilégio dos programadores, mas de toda a sociedade, era representar o ano com apenas dois dígitos. “Em que ano estamos?”, alguém perguntava. “76” era a resposta. Ninguém se dava ao trabalho de responder “1976”.

Assim, seja pela questão cultural, seja pela necessidade de economizar memória, os computadores costumeiramente representavam os anos com apenas dois dígitos. Quando iam exibir na tela, eles normalmente escreviam um valor fixo “19”, seguido do valor armazenado para o ano.

Os programadores até sabiam que isso daria algum tipo de problema quando chegasse a virada do século, pois, dessa forma, quando o calendário passasse de 1999 para 2000, os programas iam mostrar 1900 ou 19100 – o 19 era constante e o ano era representado apenas pelos valores de 00 a 99. Porém, nos anos 1970 e mesmo na década seguinte, o ano 2000 era considerado uma coisa muito distante, um futuro de ficção científica com colônias na lua e empregados robôs. Então ninguém estava preocupado com isso. Além do mais, como a tecnologia evoluía rapidamente, ninguém imaginava que sistemas desenvolvidos naquela época ainda estariam em uso no ano 2000. Todos pensavam que esses sistemas teriam sido já substituídos.

Não foi assim. Os sistemas抗igos apenas serviram de base e novos desenvolvimentos foram feitos sobre eles e não no lugar deles. O ano 1999 chegou e os sistemas não estavam preparados. Esse problema foi conhecido como “Bug do Milênio” ou “Y2K”. A [Figura 10.19](#) mostra o flagrante de uma das consequências do Y2K em uma estação de trens na França no dia 3 de janeiro de 2000.



Figura 10.19: Bug do Milênio flagrado em uma estação de trem na França.<sup>55</sup>

No entanto, os problemas causados pelo Y2K não seriam necessariamente tão inofensivos. Sistemas que lidavam com datas, como folhas de pagamento, segurança social etc. poderiam deixar de funcionar. Por exemplo, se um homem nascido em 1945 registrasse um filho em 2003 o sistema poderia pensar que o filho é 42 anos mais velho do que o pai, porque internamente o pai teria nascido em 45 e o filho em 03. Não ia funcionar.

A primeira pessoa a perceber o problema foi Bob Bemer (Estados Unidos, 1920-2004), e isso já em 1958. É que ele estava desenvolvendo software para lidar com genealogias e, neste caso, datas eram representadas ao longo dos séculos. Ele tentou avisar às empresas e publicou artigos alertando para o problema, mas, como num típico filme de cinema catástrofe, ninguém lhe deu ouvidos.

A maioria das empresas do mundo, no final do século, então mobilizou programadores para ajudar a corrigir o Bug e passar a representar os anos com quatro e não mais dois dígitos. Em especial, programadores COBOL foram procurados como se fossem

diamantes, porque nessa época a linguagem já era considerada obsoleta e poucos se interessavam em aprendê-la – no entanto, ainda havia uma infinidade de sistemas em COBOL sendo usados mundo afora.

Outro problema menos conhecido foi que o ano 2000 seria bissexto. Muitos sistemas de calendário eletrônico não consideraram essa regra que, na verdade, era a exceção da exceção. Quando o número do ano é divisível por quatro ele é bissexto. Porém, essa regra tem uma exceção: se o ano for terminado em 00, como, por exemplo, 1800 ou 1900, ele, embora divisível por quatro, não é bissexto. Essa regra tem a exceção da exceção, que acabou não sendo implementada em muitos sistemas: quando o número do ano é divisível por 400, como 1600 e 2000, então ele é bissexto, embora terminado em 00. Esse problema chegou a existir nas versões iniciais do Microsoft Excel.

Outra questão não tão importante, mas que merece nota, é que o milênio de fato não iniciou no ano 2000, mas em 2001. Mesmo assim, o termo “Bug do Milênio” foi mantido, embora a forma Y2K, derivada da expressão em inglês “year two thousand”, fosse mais correta.

Agora estamos livres do problema. Pelo menos até o ano 9.999, quando ele deverá ocorrer novamente. Porém, como pensávamos em 1970, temos muito tempo até lá.<sup>56</sup>

### **10.33 Até Aqui...**

Vimos que os anos 1990 foram de franco desenvolvimento na computação. A World Wide Web surgiu e rapidamente se estabeleceu como o principal serviço da Internet, a ponto de que hoje frequentemente as pessoas pensam que são sinônimos. Juntamente com a Web, surgiram também as linguagens de programação adaptadas para ela como Java, PHP e JavaScript.

Novos modelos de negócio baseados na Web surgiram, como a Amazon, o Google, o eBay e o Napster. Usuários começaram a

escrever seus blogs.

A miniaturização dos dispositivos seguia em frente, com a criação de computadores tão pequenos que podiam ser guardados no bolso. Em termos de comunicação, merece destaque a criação do padrão Wi-Fi, uma forma barata e eficiente de acessar a rede.

Estava tudo pronto para o próximo passo na evolução tecnológica, com o surgimento dos smartphones, coisa que aconteceria após a virada de século. A evolução cultural desse novo período seria encabeçada pelas redes sociais, que se tornaram possíveis com a maior conectividade entre as pessoas conquistada nos anos 1990.

---

<sup>1</sup> © Microsoft. Usada com permissão. Disponível em: <http://www.free-power-point-templates.com/articles/history-of-powerpoint-the-amazing-facts-you-did-not-know/>

<sup>2</sup> Caro leitor, você deve ter percebido que existe “;” no final da linha anterior, mas não no final desta aqui. Se você é programador (Pascal) sabe por que; se não é, provavelmente não tem a menor ideia. A questão é: por que você deveria saber isso? Novamente, esse tipo de detalhe que não aumenta sua capacidade de produzir programas é o tipo de coisa que Python tenta eliminar.

<sup>3</sup> Tradução: Livre

<sup>4</sup> Tradução: Aberrações.

<sup>5</sup> “Linux logo” by Larry Ewing <[lewing@isc.tamu.edu](mailto:lewing@isc.tamu.edu)>. Licenced under Attribution via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Linux\\_logo.jpg#/media/File:Linux\\_logo.jpg](https://commons.wikimedia.org/wiki/File:Linux_logo.jpg#/media/File:Linux_logo.jpg)

<sup>6</sup> By Daewoo - M;l lok8ln9 disk., CC BY-SA 3.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=6222846>

<sup>7</sup> Um protocolo para acessar, distribuir e procurar arquivos na Internet desenvolvido na Universidade de Minnesota em 1991

<sup>8</sup> Tradução: “Companhias ruins são destruídas por crises; companhias boas sobrevivem a elas; as grandes companhias são melhoradas por elas.”

<sup>9</sup> Thomas Johansson. Cortesia. Disponível em: <http://blogs.mathworks.com/cleve/2013/05/13/pentium-division-bug-affair/>

<sup>10</sup> Fonte: <http://history-computer.com/Internet/Conquering/Wanderer.html>

<sup>11</sup> Tradução: Mais um outro...

<sup>12</sup> Tradução: Mais um Outro Oráculo Oficioso Hierárquico

<sup>13</sup> “Yahoo Headquarters” by Coolcaesar at the English language Wikipedia. Licensed under CC BY-SA 3.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Yahoo\\_Headquarters.jpg#/media/File:Yahoo\\_Headquarters.jpg](https://commons.wikimedia.org/wiki/File:Yahoo_Headquarters.jpg#/media/File:Yahoo_Headquarters.jpg)

<sup>14</sup> “IBM Simon Personal Communicator” by Bcos47 - Licensed under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:IBM\\_Simon\\_Personal\\_Communicator.png#/media/File:IBM\\_Simon\\_Personal\\_Communicator.png](https://commons.wikimedia.org/wiki/File:IBM_Simon_Personal_Communicator.png#/media/File:IBM_Simon_Personal_Communicator.png)

<sup>15</sup> “Netscape Navigator 1.22 Screenshot” by Indolering - Own work. Licensed under CC0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Netscape\\_Navigator\\_1.22\\_Screenshot.png#/media/File:Netscape\\_Navigator\\_1.22\\_Screenshot.png](https://commons.wikimedia.org/wiki/File:Netscape_Navigator_1.22_Screenshot.png#/media/File:Netscape_Navigator_1.22_Screenshot.png)

<sup>16</sup> Um registro fotográfico deste evento pode ser visto em: <http://ww4.hdnux.com/photos/10/33/46/2210179/5/920x920.jpg>

<sup>17</sup> Tradução: Implacável.

<sup>18</sup> O trocadilho não fica perfeito em português. Em inglês, a frase de duplo sentido era “It will assume the Lotus position”, ou o “lugar” do Lotus.

<sup>19</sup> Tradução: Desenvolvimento Rápido de Aplicações.

<sup>20</sup> Tradução: Carvalho.

<sup>21</sup> Tradução: Escreva uma vez, rode em qualquer lugar.

<sup>22</sup> “HNL Wiki Wiki Bus” by Andrew Laing - WikiWiki!!!. Licensed under CC BY-SA 2.0 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:HNL\\_Wiki\\_Wiki\\_Bus.jpg#/media/File:HNL\\_Wiki\\_Wiki\\_Bus.jpg](https://commons.wikimedia.org/wiki/File:HNL_Wiki_Wiki_Bus.jpg#/media/File:HNL_Wiki_Wiki_Bus.jpg)

<sup>23</sup> Tradução: Zona livre de XP.

<sup>24</sup> Tradução: “Para mim chega daqui!”

<sup>25</sup> “USB Male Plug Type A” by André Karwath aka Aka - Own work. Licensed under CC BY-SA 2.5 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:USB\\_Male\\_Plug\\_Type\\_A.jpg#/media/File:USB\\_Male\\_Plug\\_Type\\_A.jpg](https://commons.wikimedia.org/wiki/File:USB_Male_Plug_Type_A.jpg#/media/File:USB_Male_Plug_Type_A.jpg)

<sup>26</sup> Tradução: Preprocessador de Hipertexto PHP.

<sup>27</sup> Tradução: Me dê a partida!

<sup>28</sup> Tradução: Conecte e use.

<sup>29</sup> Tradução: Conecte e reze.

<sup>30</sup> Tradução: A Linguagem de Scripts Orientada a Objetos Ruby.

- <sup>31</sup> Tradução: Programando Ruby.
- <sup>32</sup> “Ruby logo 64x64” by Yukihiro Matsumoto - <http://rubyidentity.org/>. Licensed under CC BY-SA 2.5 via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Ruby\\_logo\\_64x64.png#/media/File:Ruby\\_logo\\_64x64.png](https://commons.wikimedia.org/wiki/File:Ruby_logo_64x64.png#/media/File:Ruby_logo_64x64.png)
- <sup>33</sup> Tradução: Princípio do Menor Espanto.
- <sup>34</sup> Tradução: Planilhas de Estilo em Cascata.
- <sup>35</sup> “Ebayheadquarters” by Coolcaesar at English Wikipedia - Originally from en.wikipedia; description page is/was here.. Licensed under CC BY-SA 3.0 via Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:Ebayheadquarters.jpg#/media/File:Ebayheadquarters.jpg>
- <sup>36</sup> “Palmpilot5000 eu” by Channel R at English Wikipedia. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Palmpilot5000\\_eu.png#/media/File:Palmpilot5000\\_eu.png](https://commons.wikimedia.org/wiki/File:Palmpilot5000_eu.png#/media/File:Palmpilot5000_eu.png)
- <sup>37</sup> Tradução: “A anatomia de um motor de busca de hipertexto em grande escala na Web.”
- <sup>38</sup> Tradução: Boletim de notícias dos amigos do Google.
- <sup>39</sup> By James the photographer - Disponível em: <http://flickr.com/photos/22453761@N00/592436598/>, CC BY 2.0. <https://commons.wikimedia.org/w/index.php?curid=3511068>
- <sup>40</sup> Tradução: Pensador Profundo.
- <sup>41</sup> Tradução: Azul Profundo.
- <sup>42</sup> Tradução: Azul mais Profundo Ainda.
- <sup>43</sup> Tradução: Sabedoria de Robô.
- <sup>44</sup> Publicação online de 27 de maio de 2010, por Dan Fletcher. Disponível em: [http://content.time.com/time/specials/packages/article/0,28804,1991915\\_1991909,00.html](http://content.time.com/time/specials/packages/article/0,28804,1991915_1991909,00.html)
- <sup>45</sup> “Nafija55” by Nafija.shabani - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:Nafija55.jpg#/media/File:Nafija55.jpg>
- <sup>46</sup> Tradução: Livre de legado.
- <sup>47</sup> “IMac G4 sunflower7” by No machine-readable author provided. Bishonen assumed (based on copyright claims). - No machine-readable source provided. Own work assumed (based on copyright claims). Licenced under Public Domain via Wikimedia Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:IMac\\_G4\\_sunflower7.png#/media/File:IMac\\_G4\\_sunflower7.png](https://commons.wikimedia.org/wiki/File:IMac_G4_sunflower7.png#/media/File:IMac_G4_sunflower7.png)

<sup>48</sup> Tradução: Feira dos Vendedores de Computadores.

<sup>49</sup> Tradução: Tela azul da morte.

<sup>50</sup> Nota do autor: No momento em que eu escrevia esta frase e pesquisava sobre o assunto, o Chrome sofreu uma falha geral e reiniciou. Pelo menos a tela azul da morte não apareceu.

<sup>51</sup> Tradução: “Deve ser por isso que ainda não estamos distribuindo o Windows 98.”

<sup>52</sup> “Windows 9X BSOD” by The original uploader was Akhristov at English Wikipedia - Transferred from en.wikipedia to Commons by TFCforever.. Licensed under Public Domain via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Windows\\_9X\\_BSOD.png#/media/File:Windows\\_9X\\_BSOD.png](https://commons.wikimedia.org/wiki/File:Windows_9X_BSOD.png#/media/File:Windows_9X_BSOD.png)

<sup>53</sup> “Wi-Fi Alliance Logo” by Nathan Smith - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Wi-Fi\\_Alliance\\_Logo.svg#/media/File:Wi-Fi\\_Alliance\\_Logo.svg](https://commons.wikimedia.org/wiki/File:Wi-Fi_Alliance_Logo.svg#/media/File:Wi-Fi_Alliance_Logo.svg)

<sup>54</sup> “8mb ibm disk on key” by Gbuchana - Photographed in Ottawa Canada, 19-Aug-2015. This is my thumb drive that I own.. Licensed under CC BY-SA 3.0 via Wikipedia. Disponível em: [https://en.wikipedia.org/wiki/File:8mb\\_ibm\\_disk\\_on\\_key.jpg#/media/File:8mb\\_ibm\\_disk\\_on\\_key.jpg](https://en.wikipedia.org/wiki/File:8mb_ibm_disk_on_key.jpg#/media/File:8mb_ibm_disk_on_key.jpg)

<sup>55</sup> “Bug de l'an 2000” by Bug de l'an 2000 - Own work. Licensed under CC BY-SA 3.0 via Commons. Disponível em: [https://commons.wikimedia.org/wiki/File:Bug\\_de\\_l%27an\\_2000.jpg#/media/File:Bug\\_de\\_l%27an\\_2000.jpg](https://commons.wikimedia.org/wiki/File:Bug_de_l%27an_2000.jpg#/media/File:Bug_de_l%27an_2000.jpg)

<sup>56</sup> A rigor, os sistemas baseados em Unix terão um problema sério em 2038, porque o sistema de representação de tempo no Unix conta os segundos transcorridos desde 1º de janeiro de 1970. Sistemas de 32 bits conseguem representar um número de segundos suficiente para chegar até as 3h14m08s do dia 1º de janeiro de 2038. Neste momento, o número que representa o tempo passará a ser negativo, devido ao *overflow* da representação de 32 bits e, com isso, a data representada voltará a ser interpretada como 1970 ou 1901, dependendo do sistema.

# **PARTE XI**

## **Computação Móvel**

O mundo sobreviveu ao Bug do Milênio e iniciaram os anos 2000. Os computadores já eram pessoais e já estavam ligados em uma rede mundial. Agora era o momento de miniaturizá-los ainda mais para que pudessem ser carregados no bolso. O pioneirismo da IBM com o Simon na década anterior rendeu frutos para outras empresas. Os anos 2000 foram os anos da guerra dos smartphones. Inicialmente, a Apple lança o iPhone, uma cartada de gênio de Jobs, que com isso salvou a empresa da insolvência e a colocou de novo entre as grandes. Logo em seguida vem o Android da Google, que por ser um sistema livre acabou ganhando uma enorme fatia do mercado de smartphones, deixando a Apple em segundo e a Microsoft em terceiro.

Em termos de desenvolvimento de software, esse período viu a consolidação e adoção cada vez em maior escala dos modelos ágeis, um contraponto aos processos de desenvolvimento mais burocráticos definidos nas décadas anteriores. A sociedade também começou a sofrer mudanças importantíssimas causadas pela tecnologia nos anos 2000, seja com a disponibilização de smartphones exponencialmente mais poderosos e baratos, seja com a criação das redes sociais como o Facebook, dos meios de comunicação global sem custo como o Skype e a organização do conhecimento humano na Wikipédia. As transformações ocorrem cada vez mais rápido porque cada vez temos mais tecnologia para promovê-las.

## 11.1 C# – 2000

Anders Hejlsberg (Dinamarca, 1960) é reputado como criador do Turbo Pascal, o qual ele licenciou para a Borland. Posteriormente, ele também participou ativamente do projeto de criação do ambiente Delphi. Finalmente veio a trabalhar na Microsoft, especificamente no projeto .NET, que seria a resposta da empresa ao Java da SUN.

Anders criou uma linguagem especificamente para este projeto chamada C#, ou C “sharp”. A ideia do nome vem da notação musical, na qual o símbolo “#” indica que uma nota deve ser tocada meio tom acima do usual. Como em inglês as notas musicais são representadas por letras, C# seria literalmente “dó sustenido”. Porém, o termo também é um trocadilho e pode significar “C afiado”. Outra interpretação vem do fato de que o símbolo “#” é formado por quatro cruzes “+”. Assim, C# seria igual a (C++)++, ou seja, mais do que C++. De fato, C# foi fortemente inspirado em C++, mas também em outras linguagens.

Houve críticas, especialmente por parte de funcionários da SUN de que C# teria sido uma cópia de Java, mas Anders disse que embora, Java tenha sido sim uma das inspirações, C++ foi a principal, e que Smalltalk, Modula 2 e C também foram modelos importantes. C#, por exemplo, reintroduz na linguagem elementos que Java eliminou, como o uso de enumerações, que são tipos de dados que consistem em listas de nomes válidos.

Embora a Microsoft não confirme essa informação, há testemunhos de que originalmente a linguagem deveria se chamar “COOL”, significando “C-like Object Oriented Language”. Entretanto o nome já era marca registrada e seria difícil de usar. Assim, quando a plataforma .NET foi lançada em 2000, ela já veio com uma nova linguagem de programação, a C#.

Porém, C# foi uma ideia que aconteceu bem no final do projeto .NET, praticamente no último ano, pois iniciou em janeiro de 1999. Antes disso, apesar de várias linguagens serem usadas no projeto, a maioria das bibliotecas de classes de .NET estavam sendo

escritas em uma linguagem que se chamava SMC ou Simple Managed C, uma versão modificada de C++ desenvolvida pelo mesmo grupo dentro da Microsoft. Entretanto, quando .NET foi lançado em 2000 as bibliotecas já haviam sido reescritas de SMC para C#.

Desde o lançamento de C# 2.0 em 2005, ela tem ficado cada vez menos parecida com Java. Já a versão 3.0 incluiu várias características de linguagens funcionais, tornando C# assim uma linguagem multiparadigma. A versão atual, 5.0, além da programação funcional e imperativa, admite os paradigmas orientado a objetos e declarativo.

Em relação a outras linguagens semelhantes, considera-se C# mais fácil de trabalhar do que C ou C++ e, ao mesmo tempo, mais flexível do que Java. Por exemplo, ela não exige que o cabeçalho de uma classe seja declarado em um arquivo diferente de sua implementação e também não exige que os métodos de uma classe tenham que ser declarados em uma ordem específica, como acontece com outras linguagens. Um arquivo C# pode conter qualquer quantidade de classes, interfaces, *struts*<sup>1</sup> e eventos.

Assim como no caso de Java e Smalltalk, o código C# é compilado em uma linguagem intermediária independente de processador, chamada IL ou Intermediate Language. Como IL atende ao padrão internacional CLI ou Common Language Infrastructure, o código gerado por C# pode interagir com código gerado por outras linguagens no framework .NET, como por exemplo, Visual Basic, Visual C++ ou qualquer outra das mais de 20 linguagens compatíveis com o framework.

C# é uma linguagem em franco desenvolvimento. Versões com novas características ainda estão em desenvolvimento ou sendo planejadas para os próximos anos. Na tabela Tiobe, ela aparecia em 3º lugar em fevereiro de 2016.

## 11.2 Manifesto Ágil – 2001

O desenvolvimento da engenharia de software desde os anos 1960 até o final do século XX foi bastante focado em definição de processos. Com o passar do tempo e as dificuldades inerentes dessa indústria, processos cada vez mais detalhados foram definidos, normalmente com a justificativa de reduzir riscos relacionados a incertezas. Porém, o excesso de detalhe também levou a uma burocratização do processo de desenvolvimento, que acabou minando a produtividade dos desenvolvedores. Mais do que uma ajuda, o Modelo Cascata e suas variações transformaram-se num fardo para os desenvolvedores.

Porém, nem todos os metodologistas concordavam com a visão de que os processos tinham que ser cada vez mais detalhados para serem melhores. Assim, alguns modelos de processo chamados “processos leves” passaram a ser desenvolvidos principalmente a partir dos anos 1980, quando as ideias relacionadas à prototipação de software começaram a ganhar força.

Nos dias 11 a 13 de fevereiro de 2001, no Resort Snowbird nas montanhas de Utah, um grupo de 17 metodologistas se reuniu para esquiar, apreciar boa comida, beber, relaxar, e lançar as bases de um novo movimento na área de engenharia de software: o desenvolvimento ágil, que foi fundando a partir de um documento gerado nessa reunião, que ficou conhecido como “Agile Manifesto”, ou “Manifesto Ágil”. A ideia surgiu em 2000, quando Kent Back convidou um grupo de interessados em XP e alguns outros metodologistas não relacionados com XP, mas com ideias convergentes, para uma reunião no interior do Oregon. Lá decidiram que um encontro mais amplo seria necessário, e foi o que aconteceu em 2001.

O nome “ágil” passou a ser usado no lugar de “processo leve”, basicamente porque já não mais se tratava de processos, mas também porque a palavra “leve” pode soar como pejorativa, indicando processos que não são suficientemente completos.

Processos são como receitas de bolo: são instruções passo a passo sobre como fazer as coisas. Porém, o desenvolvimento ágil não seria necessariamente baseado em processos, mas em valores e práticas. A diferença pode ser explicada através de uma metáfora: o desenvolvimento baseado em processos é como ter a receita de bolo que qualquer pessoa pode seguir. Porém, as vezes a receita não é suficientemente clara, detalhada ou correta e as pessoas podem errar, especialmente se não estiverem acostumadas a fazer bolos. Um metodologista dessa linha vai detalhar a receita cada vez mais à medida que perceber que alguém não entendeu alguma coisa nela. Já o desenvolvimento ágil parte do pressuposto de que você vai contratar bons confeiteiros e dar a eles uma cozinha muito bem equipada. Você não precisa se preocupar com a receita porque eles sabem fazer bolos. Em algum momento, podem até escrever uma receita ou um conjunto de práticas que dão certo, como por exemplo, a melhor forma de mexer a massa ou os tempos ideais para cozinhar a massa dependendo de suas características, mas essas instruções e regras emergem a partir de decisões do grupo de confeiteiros – elas não são impostas por manuais. Bom, agora com essa imagem na mente, substitua os confeiteiros por programadores e você terá uma ideia um pouco mais clara sobre as diferenças entre desenvolvimento baseado em processos e desenvolvimento ágil.

Martin Fowler (Reino Unido, 1963), que estava na reunião, foi quem cunhou o termo “*agile*”. Ele apostava que metade da população dos Estados Unidos também não sabia como pronunciar essa palavra.

Já Alistair Cockburn apostou que eles não sairiam dali com nenhum acordo substancial visto que muitos modelos diferentes estavam ali representados: SCRUM, XP, FDD, Crystal, DSDM, ASD, Pragmatic Programming etc. Porém, ao final da reunião, ele e todos os presentes ficaram bastante satisfeitos porque conseguiram concordar com vários princípios que passaram a ser observados e defendidos por todos. Ward Cunningham, inventor do Wiki, se prontificou a colocar o documento na Internet. Ele rapidamente

adquiriu o domínio agilealliance.org e assim o fez. Posteriormente, o site ficou dedicado à Agile Alliance, organização que eles criaram, e o manifesto foi movido para um site dedicado: agilemanifesto.org. Qualquer pessoa pode assinar o manifesto eletronicamente e ver a lista das pessoas que já o assinaram desde 2002.

Entre outras coisas, houve concordância de que a indústria de software não era como outras indústrias, onde cada funcionário faz a sua parte e se reporta a um chefe que também é aquele que distribui as tarefas e frequentemente o que menos sabe o que precisa ser feito; esse tipo de organização é referenciada por eles como sendo “Dilbertesca”, em referência ao personagem Dilbert de Scott Adams (Estados Unidos, 1957). O software é uma construção complexa e que muda com frequência; assim, pessoas precisam colaborar fortemente para atingir objetivos comuns. Colocar, por exemplo, metas de produtividade em grupos de software é coisa muito perigosa, pois se os desenvolvedores competirem uns contra os outros, eles vão minar o projeto. Portanto, valores como respeito, comunicação franca, segurança pessoal, coragem e outros foram estabelecidos como necessários para o bom andamento de projetos de software. Isso mudou a indústria.

Hoje em dia, muitas empresas de software, especialmente aquelas orientadas para o desenvolvimento de projetos de pequeno ou médio porte utilizam, ou pelo menos dizem utilizar, alguma forma de desenvolvimento ágil. Em grande parte, isso pode ser apenas a palavra da moda, como foi “orientação a objetos” nos anos 1990, mas em diversas companhias é fato e tendência.

O movimento ágil não deve ser entendido como algo que é “contra” metodologia. Pelo contrário, o objetivo é restaurar o significado da palavra “metodologia”. Ele não é contra documentação, mas acredita que produzir centenas de páginas que não serão mantidas atualizadas em relação ao produto é inútil. Ele não é contra modelagem de sistemas, mas acredita que simplesmente arquivar diagramas feitos não se sabe por que, não é ser produtivo. Ele é a favor de planejamento, mas tem a clareza de

perceber que o ambiente de desenvolvimento é mutável e que a capacidade de adaptação às mudanças é muito mais importante do que seguir cegamente a um plano traçado. Finalmente, a autorreflexão, ou capacidade de avaliar coletivamente o que se pode melhorar, é uma peça chave nesse tipo de indústria.

O manifesto em si é conhecido por qualquer pessoa que já tenha lido um livro, artigo ou participado de algum curso ou palestra sobre modelos ágeis. Por isso, seu texto, embora curto, não será repetido aqui. O leitor que por desventura nunca tenha lido ou ouvido sobre ele pode consultar a fonte original em: <http://agilemanifesto.org/>. Por incrível que pareça, tratam-se de apenas quatro frases.

## 11.3 Windows XP – 2001

Após o lançamento do Windows 98 a Microsoft introduziu o Windows ME (Millenium Edition) em 2000 como uma atualização do 98 e no mesmo ano lançou o Windows 2000, que era uma atualização do Windows NT. Enquanto o NT e o 2000 eram especialmente voltados para o mercado empresarial, o 98 e ME eram versões para computadores domésticos.

Porém, o ME e o 2000 tiveram vida curta porque em 2001 a Microsoft já lançou o Windows XP, que veio para substituir tanto a família do 95/98/ME quanto a família NT/2000. O XP foi baseado no código do Windows 2000 e no design de interface do 98. Ele tinha duas versões, doméstica e profissional, mas era um sistema só. A ideia era trazer a confiabilidade do sistema profissional para as residências e a facilidade de uso do sistema doméstico para as empresas.

O XP foi o primeiro sistema a permitir a troca rápida de usuários no computador, sem precisar sair da conta. Ele também introduziu um sistema que prioriza os elementos no menu Iniciar em função da frequência de seu uso, o qual pode ser visto na coluna da esquerda na [Figura 11.1](#). Entre os aplicativos introduzidos no XP estavam o Messenger e o Movie Maker. Uma característica controversa era a

assistência remota, que permitiria a um técnico operar a máquina a distância para diagnosticar e resolver problemas, mas que também podia permitir a uma pessoa mal-intencionada tomar conta da máquina para fins nefastos.

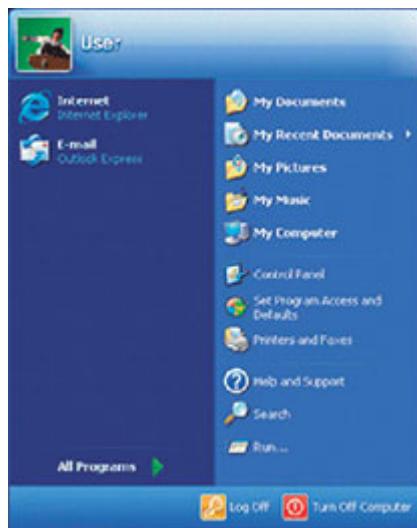


Figura 11.1: Menu iniciar no Windows XP.<sup>2</sup>

O System Restore ou Restauração do Sistema também foi uma nova característica importante porque permitia retornar o sistema a encarnações anteriores, desfazendo assim instalações ou mudanças de configuração que pudessem estar causando problemas.

A partir do Windows XP os usuários domésticos finalmente estavam livres do DOS, pois ele era um sistema completo e não apenas uma interface gráfica para o DOS.

No seu auge, estima-se que 80% dos PCs do mundo usavam Windows XP. Ele é considerado também o mais longevo sistema operacional de PCs que já existiu. Em 2004 foi lançado o Service Pack 2, que deixava o sistema mais seguro e mais adaptado a dispositivos criados depois de 2001.

O XP não foi desbancado pelo Windows Vista, lançado em 2006, porque este exigia muito do hardware e ficou com fama de “pesado”.

Foi comum usuários instalarem o Vista e depois retornarem ao XP para recuperar a eficiência de seus computadores. Muitos preferiram continuar usando-o mesmo após o lançamento do Windows 7. Apenas com o Windows 8 é que o XP foi realmente perdendo sua força. Mesmo assim, foi um sistema suficientemente adequado para sobreviver por mais de 10 anos no mercado, coisa que nenhum outro foi capaz.

Embora as vendas do XP tenham encerrado oficialmente em 2008, ele continuou sendo vendido pré-instalado em PCs até 2010. A Microsoft manteve o suporte ao produto até abril de 2014. Estima-se que nada menos do que 1 bilhão de cópias tenham sido vendidas até essa data. Em janeiro de 2016 ainda se estimava que 8% dos PCs no mundo usassem Windows XP e a taxa cresce para 26% quando se fala em PCs na China.

## 11.4 BitTorrent – 2001

Em 2001, Bram Cohen (Estados Unidos, 1975) criou um protocolo de comunicação peer-to-peer bastante inovador. Irritado com a baixa velocidade dos sistemas de compartilhamento de arquivos entre usuários, ele imaginou uma solução. O problema era o seguinte: se você queria baixar um arquivo que estava no computador de outra pessoa, então do seu ponto de vista estaria fazendo download, mas do ponto de vista do outro computador seria um upload. Só que, para a grande maioria dos usuários da internet, as velocidades de upload são muito menores do que as de download. Assim, mesmo que você tivesse uma grande velocidade de download, não conseguiria baixar o arquivo tão rapidamente porque o upload da fonte dele seria um limitador, uma espécie de gargalo.

A solução que Cohen imaginou foi construir um sistema em Python que carrega um arquivo a partir de várias cópias dele espalhadas em diversos computadores. Assim, carregando várias partes ao

mesmo tempo, as baixas velocidades de upload eram compensadas pela quantidade de fontes das quais você faz o download.

Esse novo protocolo foi batizado de BitTorrent, ou simplesmente Torrent. Ele era de fato centenas – talvez milhares – de vezes mais rápido do que os outros protocolos de compartilhamento de arquivos existentes como KaZaA e eDonkey.

A ideia veio da empresa onde Cohen trabalhava, uma startup chamada MojoNation, que desenvolvia um sistema que permitia que arquivos criptografados confidenciais fossem quebrados em pedaços e divididos entre vários computadores. Se alguém quisesse baixar o arquivo, precisaria acessar os vários computadores ao mesmo tempo.

O BitTorrent ganhou fama por permitir o compartilhamento rápido de música e filmes. Estima-se que hoje (2016) cerca de 50% do tráfego de rede na Internet seja devido a algum tipo de Torrent.

Para funcionar, ele precisa que cada arquivo tenha várias fontes. Quanto mais popular um arquivo, mais fontes ele tem. Versões completas de um arquivo são chamadas de “seed” ou “semente”, enquanto que versões incompletas, normalmente formadas por arquivos que ainda estão sendo baixados por algum usuário são aproveitados no “swarm” ou “enxame”.

Para garantir que arquivos sejam efetivamente disponibilizados, o BitTorrent usa um algoritmo baseado no princípio de *tit-for-tat*, ou seja, quanto mais um usuário compartilha, mais ele consegue baixar. Assim, usuários que não compartilham nada terão velocidades de download bem menores.

Em 2004, Cohen, seu irmão Ross e o parceiro de negócios Ashwin Navin criaram a empresa BitTorrent, Inc. O modelo de negócio deles é baseado em anúncios. O código desde o início era livre e gratuito, mas a partir da versão 6.0 o sistema foi rebatizado como µTorrent e tornou-se código proprietário da BitTorrent Inc.

O problema de quebra de direitos autorais, porém, sempre representou dor de cabeça para a empresa. Embora ela não

armazene nem disponibilize material protegido diretamente, muitos tribunais consideram criminoso o fato de disponibilizarem uma ferramenta que permite que os usuários façam isso.

## 11.5 Wikipédia – 2001

Durante o século XX o conceito de enciclopédia foi bastante valorizado, especialmente entre as famílias que queriam disponibilizar a seus filhos conhecimento sobre todas as coisas – e assim compravam caríssimos volumes em papel, muitas vezes em fascículos semanais. Porém, a computação surgiu, e as ideias de Vannevar Bush sobre organização e acesso ao conhecimento deram origem aos hipertextos e, com eles, novas formas de construir e organizar enciclopédias.

A Microsoft foi uma das pioneiras das enciclopédias eletrônicas ao lançar a *Encarta*, uma enciclopédia digital publicada em CD-ROM que foi disponibilizada entre 1993 e 2009. Antes dela ainda houve as iniciativas da *Compton's Multimedia Encyclopedia* (1989), a *The New Grolier Multimedia Encyclopedia* (1992) e a *Funk & Wagnalls Encyclopedia*, que já existia em papel desde a década de 1930 e que foi adquirida pela Microsoft para dar origem à *Encarta*.

Entretanto, elas, assim como praticamente todas as outras enciclopédias da época, tinham seus artigos escritos por especialistas especialmente contratados, e revisados por um corpo editorial. Ou seja, a única novidade era a forma de disponibilização do material, agora em meio eletrônico e com hiperlinks ligando um assunto a outro.

A Wikipédia nasceu com um conceito diferente e desde o início bastante controverso: liberdade total. Com certeza era mais uma iniciativa que tinha tudo para dar errado; mas não deu.

A ideia de uma enciclopédia livre é atribuída a Richard Stallman (criador do GNU), em um artigo publicado em 2000, no qual apresentava o conceito e as vantagens da enciclopédia livre como recurso educacional e definia as formas de liberdade que poderiam

ser utilizadas. Ele previa, logo no início do artigo, que a construção da enciclopédia livre e universal aconteceria sem maior esforço desde que não houvesse interferência das organizações que já estavam organizando conhecimento para vendê-lo a quem quisesse obter educação.

Entre outros princípios, ele mencionava a necessidade de não existir um controle ou editor central, mas um sistema de revisão coletivo, no qual qualquer pessoa poderia publicar qualquer coisa, mas também qualquer outra pessoa poderia criticar e corrigir certas informações. Artigos passariam a ser avaliados pela comunidade e não por organizações, que possivelmente teriam seus próprios vieses. Como era de seu feitio, Stallman publicou suas ideias para quem quisesse colocá-las em prática e essas pessoas foram Jimmy Wales (Estados Unidos, 1966) e Larry Sanger (Estados Unidos, 1968) em 2001.

Eles estavam envolvidos com o projeto da Nupedia, iniciado em 1999 e que seria uma enciclopédia na Web de conteúdo aberto e contribuições voluntárias. Porém, o processo de publicação na Nupedia seria mais ou menos o mesmo de uma revista científica, com revisões por pares e avaliação editorial. No total, eram pelo menos sete passos que tinham que ocorrer até que um artigo pudesse ser finalmente publicado. Apenas 21 artigos foram efetivamente publicados no primeiro ano de funcionamento da Nupedia e, nessa marcha, ela nunca seria uma enciclopédia suficientemente abrangente sob nenhum ponto de vista.

Wales e Sanger discutiram formas de aumentar a quantidade de contribuições para a Nupedia e, após um encontro com Ward Cunningham, criador da WikiWikiWeb, e de um jantar no qual Ward fez uma preleção sobre como funcionavam as Wikis, surgiu a ideia de adicionar uma publicação lateral à Nupedia, ou seja, ela seria uma publicação de entrada, na qual artigos ainda não avaliados seriam publicados e que posteriormente, se fossem aprovados, passariam a compor a enciclopédia principal.

Muitas pessoas acharam a ideia questionável: “Qualquer pessoa publicar o que quiser, sem passar pelo crivo de um comitê avaliador? Absurdo!” Os editores e revisores da Nupedia sequer queriam associar o nome da enciclopédia com esse tipo de iniciativa.

Assim, Wales decidiu transformá-la em um projeto à parte, o qual ele chamou de Wikipédia, a enciclopédia organizada em Wikis. O domínio wikipedia.com e wikipedia.org foram registrados e, em 15 de janeiro de 2001, a Wikipédia entrou no ar. O domínio oficialmente era wikipedia.org, visto que a organização nunca teve fins lucrativos, mas para evitar que empresas usassem o termo, eles registraram também o domínio wikipedia.com, o qual simplesmente redireciona para wikipedia.org. O primeiro logotipo de 2001, quando o projeto ainda era de certa forma ligado à Nupedia, é apresentado na [Figura 11.2](#).



Figura 11.2: Primeiro logotipo da Wikipédia.<sup>3</sup>

O primeiro artigo foi “Hello World!”, um teste publicado pelo próprio Wales em 2001. O mais antigo ainda em exibição é “UuU”, de 16 de janeiro de 2001, cujo conteúdo é a lista com os nomes “United Kingdom, United States, Uruguay”.

O projeto começou a ser divulgado e, de forma totalmente inesperada, provocou grande interesse na comunidade desde o

início. O milésimo artigo foi publicado em 12 de fevereiro do mesmo ano (menos de um mês após a criação). No primeiro ano de existência, cerca de 20 mil artigos foram publicados, contra 21 da Nupedia.

Em 2003 foi criada a Fundação Wikipédia, para dar suporte à iniciativa. Esse projeto nunca usou de propagandas ou financiamentos para se manter, apenas doações de seus usuários.

Porém, a liberdade exige responsabilidade e isso não ficou claro para a Wikipédia até o primeiro escândalo acontecer em 2005, o Incidente Seigenthaler. Em um artigo publicado anonimamente, era apresentada uma bibliografia do jornalista John Seigenthaler, que, entre outras notícias falsas, afirmava que ele era suspeito pela morte de John Kennedy e Robert Kennedy.

O incidente passou despercebido por meses até que Seigenthaler escreveu sobre isso para o jornal *USA Today*. Mesmo que o artigo tenha sido apagado, isso manchou a credibilidade da Wikipédia como fonte de informações confiáveis.

Wales alegou que a possibilidade de publicação anônima era importante porque pessoas em países onde não havia liberdade de imprensa, como China e Irã, precisavam ter sua identidade preservada para que escrevessem livremente. Entretanto, como isso levou a abusos desse tipo, passariam a controlar melhor as publicações anônimas, que não seriam mais permitidas da mesma forma como as publicações assinadas por usuários identificados.

A Wikipédia precisou, a partir daí, tomar outras providências para que páginas fossem protegidas contra vandalismos e mesmo banidas, caso seu conteúdo fosse considerado inapropriado.

Em relação à confiabilidade, um estudo publicado em 2005 pela *Nature* afirma que a confiabilidade da Wikipédia estava no mesmo nível da Britannica e que ambas eventualmente continham erros. Outros estudos apontam que a Wikipédia é uma fonte de informação bastante completa e de alta qualidade e que erros ou vandalismos são rapidamente corrigidos pela comunidade.

Em 2015, a Wikipédia era o sétimo site mais popular da internet e a maior enciclopédia de todos os tempos, com 36 milhões de artigos em 291 línguas. Nesse ano, ela recebeu mensalmente 10 bilhões de visitas de 495 milhões de pessoas.

O projeto da Wikipédia impressa foi realizado por Michael Mandiberg (Estados Unidos, 1977) em 2015, o qual imprimiu todo o seu conteúdo em 7.473 volumes de 700 páginas cada. Agora imagine quanto isso ia custar se fosse vendido de porta em porta como as antigas enciclopédias em papel, e quem teria espaço para armazenar tal volume de informação em casa...

Em 2012, a *Britannica* anunciou que não mais iria imprimir suas enciclopédias – hoje só acessíveis pela Internet ou nas estantes de algumas avós.

## 11.6 Skype – 2003

O Skype foi criado em 2003 por Niklas Zennström (Suécia, 1966) e Janus Friis (Dinamarca, 1976). A tecnologia originou-se com o KaZaA, um sistema de compartilhamento de arquivos peer-to-peer, mas como esse tipo de iniciativa ao estilo do Napster estava tendo problemas com a justiça, eles resolveram usar a tecnologia peer-to-peer para transmissão de voz, como se fosse uma linha telefônica sobre a Internet. Assim, não haveria problemas com direitos autorais.

A tecnologia P2P do KaZaA foi desenvolvida por Ahti Heinla (Estônia, 1972), Priit Kasesalu (Estônia, 1972) e Jaan Tallinn (Estônia, 1972) e foi comprada por Niklas e Janus para a criação do Skype.

O nome Skype deriva de “*sky peer-to-peer*”, que foi abreviado para Skyper. Quando perceberam que o domínio skyper não estava totalmente disponível, abreviaram para Skype e não tiveram mais problemas em registrar.

Com o Skype, quaisquer pessoas com um computador e acesso decente à Internet podem se comunicar a custo zero, desde que

ambas rodem uma versão do software. A tecnologia peer-to-peer foi fundamental para o sucesso deste sistema. Por exemplo, duas pessoas no interior podem se comunicar entre si sem que sua comunicação precise ser roteada por servidores em grandes centros. Isso eliminava gargalos.

Além disso, era grátis. Empresas telefônicas costumam tarifar ligações em função da distância. Uma ligação de São Paulo para o Japão, por exemplo, pode custar uma verdadeira fortuna. Porém, com o Skype seria de graça. Isso foi uma verdadeira revolução no modelo de negócios de comunicação.

Essa tecnologia de capturar voz, converter em dados binários, transmitir pela internet e reconstruir os sons no destino também é conhecida atualmente como VoIP, ou Voz sobre IP (Internet Protocol).

Mas como o Skype faturava? Também de uma forma inovadora. Se você quisesse ligar para um telefone fixo ou celular, podia usar o Skype também. Qual a vantagem? Se você vai ligar para o Japão usando a telefonia normal, vai gastar uma fortuna por minuto, mas se ligar para lá usando o Skype, sua voz será transmitida pela Internet sem nenhum custo para você até o mais próximo possível do lugar onde está o telefone de destino. A partir de lá, uma chamada, normalmente local, será feita e você será tarifado de acordo com a chamada local, com o acréscimo de uma pequena margem de lucro para o Skype. Ainda assim, trata-se de um valor muito mais barato do que usar a telefonia normal. Ou ainda, se você morasse nos Estados Unidos, pagando uma pequena taxa mensal (2,95 dólares em 2010), você poderia ligar de graça para qualquer telefone dentro do país.

Com o passar do tempo outras funcionalidades foram implantadas, como a videoconferência, secretaria eletrônica, serviço de mensagens por texto, dentre outros.

Em 2005, a eBay comprou o Skype por 2,6 bilhões de dólares. Em maio de 2011, a Microsoft comprou o Skype do eBay por 8,5

bilhões. Ela estava interessada na imensa base de usuários, cerca de 700 milhões, além da possibilidade de integrar o serviço nos seus produtos, como o Windows, Windows Phone, Internet Explorer, Outlook, Xbox etc. A Microsoft chegou mesmo a desativar seu serviço de comunicação, o Messenger, pouco depois da aquisição do Skype, transferindo todos os seus usuários para o recém-adquirido serviço.

## 11.7 Fundação Eclipse – 2004

Em 2001 a IBM criou internamente o Projeto Eclipse para o desenvolvimento de ferramentas para suporte a atividades de programação em várias linguagens. Porém, o projeto foi lançado de forma pouco usual: as ferramentas produzidas seriam todas de domínio público e um consórcio foi criado para efetuar seu desenvolvimento. O consórcio inicialmente foi formado por IBM, Borland, Rational, Red Hat, TogtherSoft, Webgain, SuSE, QNX e MERANT, algumas das quais eram concorrentes entre si.

Uma das principais motivações da IBM para lançar o projeto dessa forma foi o fato de que vários ambientes de programação já estavam em uso e que, para um ambiente se tornar efetivamente líder, e com isso criar um ecossistema benéfico para todas as empresas envolvidas, era através da disponibilização gratuita e aberta dessas ferramentas.

Para participar do consórcio era necessário firmar um compromisso de boa-fé, mas não obrigatório, de usar as ferramentas da plataforma Eclipse internamente, promover seu uso e produzir pelo menos um produto baseado nela.

O consórcio promoveria as relações comerciais e de marketing, enquanto que a comunidade cuidaria do código. Era um modelo inovador de desenvolvimento de software livre. Dentro dessa filosofia, como já vimos, também não seria proibido que quaisquer empresas produzissem e vendessem sistemas comerciais construídos ou baseados na plataforma.

Em 2003, o Consórcio Eclipse já contava com 80 membros. No início do ano seguinte foi anunciado que ele seria reorganizado em uma fundação sem fins lucrativos chamada “Fundação Eclipse”. Isso ocorreu porque muitas empresas ainda viam o projeto Eclipse como subordinado à IBM. Afinal, apesar de existir o consórcio, a maioria dos recursos financeiros e de software provinham da empresa. Criar uma organização independente da IBM era necessário para melhorar a adoção desses produtos em um mercado mais amplo.

A fundação tem, desde o seu início em 2004, ativamente produzido e incentivado comunidades a se formarem, no sentido de construir as mais diversas ferramentas livres e de código aberto para desenvolvimento de software. Todo ano, no final de junho, são lançadas oficialmente novas versões das várias ferramentas sendo desenvolvidas pela iniciativa. A fundação é mantida pelas anuidades pagas pelos seus membros e por doações.

No ano em que este livro foi publicado (2016), a Fundação Eclipse possuía mais de 250 projetos simultâneos, dos quais, os mais conhecidos são os ambientes de programação (IDE) para Java e outras linguagens. Porém, também ferramentas de modelagem, geração de relatórios e teste de software, entre muitas outras, são disponibilizadas gratuitamente através da licença EPL (Eclipse Public License).

## **11.8 Facebook – 2004**

O Facebook de Mark Zuckerberg (Estados Unidos, 1984) e Eduardo Saverin (Brasil, 1982) certamente não foi a primeira rede social, mas merece destaque por ter sido a pioneira em alcançar a marca de mais de 1 bilhão de membros. No início, Zuckerberg era o rapaz da computação e Saverin o rapaz das finanças: ele tinha 40 dólares.

As primeiras iniciativas de redes sociais na Internet possivelmente foram as comunidades USENET. Em meados dos anos 1990, com a popularização da Web, algumas comunidades como TheGlobe, Geocities e Tripod foram formadas com o objetivo de reunir pessoas

para interagir a partir de mensagens instantâneas e publicações de páginas pessoais. No início do século XXI, as redes sociais mais populares eram o MySpace, Friendster, Linkedin, Orkut (especialmente no Brasil) e Bebo.

Antes do Facebook, porém, Zuckerberg criou o Facemash, lançado em 28 de outubro de 2003, juntamente com Andrew McColum (Estados Unidos, 1979), Chris Hughes (Estados Unidos, 1983) e Dustin Moskovitz (Estados Unidos, 1984). O sistema permitia aos usuários comparar duas fotos e escolher a mais bonita. As fotografias foram hackeadas principalmente a partir de acesso aos arquivos de estudantes (os “facebook”s) mantidos pela Universidade de Harvard para que os estudantes pudessem conhecer uns aos outros.

O site foi fechado pela universidade poucos dias depois e Zuckerberg, acusado de violação de copyright, quebra de segurança e violação de privacidade individual, o que poderia levar à sua expulsão. Isso acabou não acontecendo e, no final, as acusações foram retiradas.

Em janeiro de 2004, ele começa a escrever o código para uma nova página, a “Thefacebook”, que foi lançada em fevereiro no endereço thefacebook.com. A página de abertura original ainda pode ser vista em

<https://en.wikipedia.org/wiki/Facebook#/media/File:Thefacebook.png>

Após alguma divulgação em uma lista de discussão com 300 alunos, ele e seus colegas de quarto acompanharam o processo de registro, e após 24 horas entre 1.200 e 1.500 pessoas já haviam criado seus perfis no Thefacebook.

Seis dias depois, ele é acusado por outros estudantes, Cameron Winklevoss (Reino Unido, 1981), Tyler Winklevoss (Reino Unido, 1981) e Divya Narendra (Estados Unidos, 1982), de ter roubado suas ideias de construir uma rede social chamada HarvardConnection.com ao fingir trabalhar para eles na produção do

código desse projeto enquanto secretamente criava a própria rede social. O caso foi arquivado em 2007 por questões técnicas.

Inicialmente, apenas alunos de Harvard poderiam ser membros do site. Em um mês, mais da metade dos alunos tinha criado um perfil. Em março as Universidades de Stanford, Columbia e Yale também foram autorizadas a participar. Logo, alunos de várias universidades nos Estados Unidos e Canadá estavam participando da iniciativa. Uma empresa foi criada pela metade do ano e Sean Parker, um dos fundadores do Napster, que era conselheiro informal de Zuckerberg, acabou se tornando o primeiro presidente da Facebook, enquanto Saverin era colocado de escanteio.

O primeiro aporte financeiro fora as economias de Zuckerberg e Saverin foi um cheque de 500 mil dólares do investidor Peter Thiel (Alemanha, 1967), que, com isso, comprou 10,2% da empresa e assumiu uma posição no conselho.

Ainda em 2004 a companhia muda sua base para Palo Alto, na Califórnia, e compra por 200 mil dólares o domínio facebook.com, eliminando portanto o “the” em 2005.

Nesse ano, o Facebook já era usado por universidades em todo o mundo e uma versão para escolas estava em andamento. O ingresso ainda era feito por convite. Finalmente, em setembro de 2006, o Facebook torna possível que qualquer pessoa com 13 anos ou mais e um e-mail válido possa criar um perfil.

Neste mesmo ano, a Yahoo! fez uma oferta de 1 bilhão para comprar o Facebook, que foi recusada por Zuckerberg, alegando que não tinha interesse em vender a empresa. Seu sonho, desde os primeiros dias na universidade, era ser dono do site mais popular do mundo e ele estava a caminho disso. Ele teria dito que não saberia o que fazer com o dinheiro, provavelmente acabaria criando outra rede social; mas ele já estava contente com a que tinha.

Porém, até 2008 a empresa valia bilhões, mas não sabia como lucrar com esse patrimônio. Com a contratação de Shery Sandberg (Estados Unidos, 1969) para cuidar desse assunto, o modelo de

anúncios foi modificado e, em setembro de 2009, a companhia anunciou que pela primeira vez na história foi lucrativa.

O Facebook sempre foi alvo de boatos. Dentre eles, um que assustou muita gente em 2010 foi o de que a empresa ia mudar as diretrizes de privacidade e abrir os dados de todos seus usuários para empresas. A história gerou muito barulho e falou-se até no fim da rede social, mas no final pouquíssimas pessoas realmente saíram.

Em 2010 também surgiu o botão “curtir”, que hoje, apenas seis anos depois, já faz parte definitiva de nossa cultura. Nesse mesmo ano também foi lançado o filme *A Rede Social* (*The Social Network*), a partir do qual Zuckerberg lamentou não ter sido muito preciso em relação aos fatos e motivações reais para a criação do Facebook.

Zuckerberg foi a pessoa mais jovem a se tornar bilionária na história e seu site mudou a forma como seres humanos se relacionam no mundo inteiro. Apenas para citar um exemplo, as manifestações populares de 2013 no Brasil foram organizadas a partir da rede social. Que se saiba, não havia um comando central do movimento, como usualmente acontecia até então: a coisa toda emergiu a partir do comportamento coletivo de indivíduos na rede social.

## 11.9 YouTube – 2005

A história do YouTube começou graças à PayPal, mas não teve nada a ver com a empresa em si. Na verdade, três empregados da PayPal, Chad Hurley (Estados Unidos, 1977), Steve Chen (Taiwan, 1978) e Jawed Karim (Alemanha, 1979) tiveram a ideia e iniciaram o serviço. O site YouTube.com foi lançado no dia 14 de fevereiro de 2005, dia dos namorados nos Estados Unidos.

A ideia, segundo eles, surgiu em um jantar festivo um ano antes. Além disso, dois eventos que ocorreram em 2004 os incentivaram ainda mais a criarem um site para divulgação de vídeos. O primeiro foi o caso que ficou conhecido como Nipplegate – quando um dos seios da cantora Janet Jackson (Estados Unidos, 1966) ficou

exposto por meio segundo durante o show do intervalo em uma partida esportiva –, também lembrado como o “mal-funcionamento do guarda-roupa de Janet Jackson”. Naquele dia o nome da cantora foi o mais procurado na Internet por pessoas que não viram a cena ao vivo e que jamais a veriam porque as TVs não a exibiram novamente. O evento também contribuiu para atrair milhares de novos usuários para o TiVo, um aparelho que permite gravar e reproduzir a programação da TV.

Outro evento do final de 2004 que causou grande comoção e cuja procura por filmes do fato também motivou a criação do YouTube foi o tsunami que ocorreu na Indonésia. Pela primeira vez na história uma catástrofe deste porte foi amplamente filmada devido à disseminação das câmeras digitais. E as pessoas buscavam pelos vídeos.

Hurley era o responsável pelo design, enquanto Chen e Karim eram os rapazes da computação. Basicamente, perceberam que nos últimos anos tinha crescido muito a facilidade de criar vídeos, especialmente digitais, mas ainda não existia um lugar na Internet específico para divulgá-los. Naquela época era relativamente difícil transferir arquivos de vídeo pela Internet, mas a velocidade da rede não parava de aumentar, e eles apostaram que rapidamente isso se tornaria trivial. Assim, mais uma vez, quem saiu na frente acertou.

O primeiro escritório do YouTube foi em uma sala simples sobre uma pizzaria em São Mateo, na Califórnia. O registro do site em fevereiro de 2005, porém, não correspondeu à data de lançamento, porque ainda faltava escrever o software. Uma primeira versão, ou protótipo, foi apresentada ao público em maio do mesmo ano e o lançamento oficial ocorreu em novembro, quando também a empresa recebeu seu primeiro investimento: 3,5 milhões de dólares da Sequoia Capital.

O primeiro vídeo carregado no YouTube foi “Me at the zoo”, mostrando Karim no Zoológico de San Diego em frente a alguns elefantes. O vídeo publicado em 23 de abril de 2005 ainda pode ser

visto em <https://www.YouTube.com/watch?v=jNQXAC9IVRw>.

Tratava-se de uma produção extremamente simples e amadora. Karim apenas comentava que estava em frente dos “uh... elefantes” e que a coisa legal sobre eles é que têm trombas realmente, realmente, realmente longas. Comenta-se que esse vídeo marcou o início de uma revolução cultural em relação a como as pessoas produzem e consomem mídia. Toda uma nova linguagem nunca antes vista nas redes de televisão começou a surgir e evoluir. Atualmente há pessoas que fizeram fortuna, digamos, filmando a si mesmas jogando videogames e os comentando – como o caso de Felix Kjellberg (Suécia, 1989), mais conhecido como PewDiePie.

Em 2006, o YouTube cresceu vertiginosamente, chegando a se tornar o 5º site mais popular da Internet. O público dominante era de adolescentes. Em setembro, o primeiro vídeo a atingir 1 milhão de visualizações foi um comercial da Nike estreado pelo jogador brasileiro Ronaldinho Gaúcho. A Nike logo se tornou uma das grandes investidoras em propaganda no YouTube.

Ainda em 2006 houve o primeiro incidente, quando a rede de televisão NBC pediu ao YouTube para remover um vídeo que era um clipe do programa *Saturday Night Live*. O site atendeu à solicitação e aproveitou para criar seu programa de verificação de conteúdo, que permite que os donos dos direitos sobre vídeos não públicos possam localizar e remover qualquer vídeo que viole seus direitos autorais. Mais tarde, a NBC tornou-se parceira do YouTube na produção de conteúdo específico para ele.

Em outubro de 2006 a Google fez uma proposta de 1,65 bilhão de dólares pelo YouTube, que foi aceita. A empresa que tocava o YouTube então tinha apenas 65 funcionários, mas a Google entendia que esse site seria o próximo passo na evolução da Internet.

Em 2007, agora nas mãos da Google, o YouTube inovou mais uma vez. A partir desse ano, eles possibilitaram a usuários que tivessem vídeos muito assistidos e que, portanto, atraíssem e divulgassesem

propaganda no site, pudessem receber parte do lucro que a empresa obtinha com essas propagandas. A partir de então, muitas pessoas talentosas puderam transformar seus hobbies em profissão. Pessoas comuns, saídas da multidão, passaram a ter programas que atraíam muitos visualizadores, e várias enriqueceram com o dinheiro que entrava em suas contas, chegando muitas vezes a abandonar ou desistir de terem outro tipo de emprego. Uma família que postou o vídeo de dois irmãos pequenos, no qual um, Charlie, morde o dedo do outro, teve meio bilhão de visualizações – esse simples vídeo rendeu à família mais de 150 mil dólares.

A partir de 2010 passou a ser possível comprar filmes ou alugá-los no YouTube. Assim, filmes completos em alta qualidade passaram a estar disponíveis via Internet a um custo muito baixo e sem violar nenhuma lei de direitos autorais. Não é para menos que o mercado das videolocadoras despencou vertiginosamente nos últimos anos, sendo hoje uma sombra do que já foi.

Em 2011, o YouTube começa a transmitir ao vivo, e em 2011 lança seus próprios canais com conteúdo original, passando a funcionar também praticamente como uma rede de TV – uma rede que tem mais audiência do que o total de espectadores do horário nobre de todas as redes norte-americanas juntas.

## 11.10 Twitter – 2006

O Twitter foi uma ideia de Jack Dorsey (Estados Unidos, 1976). Trata-se de um sistema simples para envio de mensagens curtas (nunca mais de 140 caracteres) que no início pensava-se que fosse uma rede social ou um sistema de mensagens instantâneas, mas que acabou tornando-se algo único, e altamente viciante: um sistema de mensagens curtas com gramática não convencional. Você selecionava as pessoas que queria “seguir” e passava a receber todas as mensagens que elas colocassem no Twitter.

O sistema era um projeto que Jack realizava nas horas de folga. Porém, quando a empresa em que ele trabalhava, a Odeo, viu-se enrascada, o Twitter acabou se tornando um projeto dela. O problema com a Odeo é que eles estavam desenvolvendo uma plataforma para distribuição de músicas, mas quando a Apple lançou o iTunes, a plataforma deles tornou-se irrelevante da noite para o dia.

Houve várias tempestades cerebrais, grupos de trabalho e incentivo à proposição de projetos envolvendo os funcionários da Odeo. Os cofundadores Evan Williams (Estados Unidos, 1972) e Christopher “Biz” Stone (Estados Unidos, 1974), ex-funcionários da Google, gostaram da ideia de Jack, e seu projeto foi a tábua de salvação da Odeo. Porém, os investidores da companhia não gostaram da ideia, e assim, Evan criou a Obvious Corp., que comprou a Odeo de volta dos investidores. Estes, na época, ficaram agradecidos por Evan os ter tão amavelmente livrado de uma possível perda financeira, mas cinco anos depois, quando o valor do Twitter tinha se multiplicado por 1.000, eles certamente ficaram bastante chateados, para dizer o mínimo.

O nome Twitter foi sugerido por Noah Glass, outro cofundador da Odeo, e um dos grandes incentivadores iniciais da ideia. O termo significa o canto do pássaro ou um surto de mensagens rápidas e inconsequentes, e era esse o significado que eles queriam associado à plataforma. O projeto inicialmente foi batizado de “twtr” porque o domínio twitter já era utilizado e naquele tempo era comum empresas batizarem seus projetos sem algumas vogais, para justamente evitar que eventuais vazamentos de informação fizessem os domínios serem registrados por outros que depois os venderiam bem caro. Uma das empresas que manteve o nome com menos vogais foi a Flickr. Outros nomes que foram cogitados são “FriendStalker” e “DodgeBall”.

Logo após a recompra da empresa por Evan, ele demitiu Noah Glass, por razões que ainda não ficaram totalmente claras, mas que provavelmente tinham a ver com luta por poder. Glass ficou

chocado, visto que era um dos grandes incentivadores e executores do projeto Twitter. Anos mais tarde, comentou que se sentira traído e, referindo-se às redes sociais formadas com ajuda do aplicativo, questionou: “*Wait... what's the value in building these relationships if this is the result?*”<sup>4</sup>

O primeiro twitt foi dado por Jack no dia 21 de março de 2006 às 9h50. A mensagem era simplesmente “*just setting up my twtr*”,<sup>5</sup> que ainda pode ser conferida em <https://twitter.com/jack/status/20>.

Como o sistema era inicialmente baseado em mensagens SMS enviadas por celular, os membros da equipe chegaram a gastar várias centenas de dólares em suas contas telefônicas durante o desenvolvimento. Isso não era porque realmente precisassem enviar tantas mensagens, mas porque já estavam viciados no fenômeno. No final, a empresa concordou em pagar as contas.

O usuário do Twitter enviava mensagens de texto a um número telefônico que então a distribuía para seus seguidores. A limitação de 140 caracteres por twitt veio do fato de que naquela época mensagens de SMS eram limitadas a 160 caracteres e eles quiseram deixar 20 caracteres para o nome de usuário.

O sistema foi lançado ao público em julho de 2006. Em 2007, Twitter saiu da Obvious e tornou-se uma empresa independente. Nessa época, ele teve seu primeiro grande sucesso que foi durante a conferência South by Southwest Interactive (SXSWi). As pessoas ali podiam acompanhar posts no Twitter em tempo real em dois televisores de plasma de 60 polegadas colocados na área da conferência. Nesse dia, o número de mensagens tuitadas diariamente subiu de 20 mil para 60 mil.

O Twitter cresceu e ele era mais do que apenas as pessoas falando sobre o que tinham comido no café da manhã ou onde estavam passando as férias; ele se tornou um fenômeno em relação a assuntos políticos, comerciais e principalmente relacionados ao entretenimento: imagine receber diariamente mensagens pessoais escritas pela sua cantora favorita, ou seu ator de Hollywood? Sim,

você também poderia seguir seu senador, vereador, o mercado próximo da sua casa, que anunciava uma oferta imperdível só naquele dia, e assim por diante. A ideia era “quente”!

Porém, se o Twitter se alegrava ao ver o número de usuários dobrar de um dia para outro (literalmente), isso não era tão favorável para os servidores que tinham que dar conta de tantas mensagens. A empresa chegou a criar um símbolo que foi usado até recentemente para pedir desculpas aos usuários quando o site estava fora do ar devido ao grande fluxo: uma baleia bem descansada sendo carregada por oito pequenos pássaros. A imagem ficou conhecida como “Fail Whale”.<sup>6</sup>

Uma das vezes em que os servidores do Twitter caíram foi quando morreu o cantor Michael Jackson, em junho de 2009. Outra ocasião foi quando Ellen Degeneres postou uma *selfie* com vários atores durante a entrega do Oscar em março de 2014: o Twitter foi derrubado por pelo menos 20 minutos em função da quantidade de acessos.

Entre 2007 e 2008, o número de mensagens tuitadas por trimestre subiu de 400 mil para 100 milhões. Em 2010 eram 50 milhões de mensagens por dia e em 2011, 140 milhões. Em maio de 2015, o Twitter tinha cerca de 500 milhões de usuários registrados.

Em 2009 aconteceu o Twitpocalypse, porque cada mensagem postada no Twitter desde seu início recebeu um identificador numérico único. Acontece que no dia 12 de junho deste ano, o número de mensagens excedeu 2.147.483.647, que é o maior número inteiro que se pode representar com 32 bits. Correções logo foram feitas e apenas os usuários dos iPhones tiveram que esperar um pouco mais pela solução porque a correção do aplicativo primeiro precisava ser aprovada pela Apple Store.

A primeira mensagem tuitada do espaço foi do astronauta T. J. Creamer da Estação Internacional em janeiro de 2010, logo depois que a NASA liberou pela primeira vez o acesso à internet no espaço. A mensagem foi: “Hello Twitterverse! We’re now LIVE

*tweeting from the International Space Station - the 1st live tweet from Space! :) More soon, send your ?s*<sup>7</sup>

Antes disso, mensagens tinham que ser encaminhadas à Terra por e-mail e então tuitadas por lá.

Atualmente, o Twitter funciona também como site na internet e aplicativo móvel. Usuários registrados podem publicar e ler mensagens e usuários anônimos podem apenas ler.

Quando a empresa foi premiada com o Web Award, eles receberam o prêmio com o seguinte comentário: “*We'd like to thank you in 140 characters or less. And we just did!*”<sup>8</sup>

## 1.11 Computador Quântico – 2007

Desde Konrad Zuse, os computadores binários são capazes de representar informação usando bits. Cada bit, como já vimos, pode valer 0 ou 1. Palavras são formadas por sequências de bits. Por exemplo, com três bits podemos formar as palavras 000, 001, 010, 011, 100, 101, 110 e 111, correspondendo, aos números decimais 0, 1, 2, 3, 4, 5, 6 e 7. Então, mesmo os mais modernos computadores que temos hoje, mesmo os supercomputadores, continuam representando a informação exatamente dessa forma, com números binários, só que atualmente o mais usual são palavras de 64 bits e não de apenas 3 bits, o que permite representar uma quantidade de números muito maior:  $2^{64}$  que é aproximadamente igual a  $1,84467 \times 10^{19}$ , ou seja, um número inteiro positivo com 19 dígitos; muito maior, portanto, do que o número de segundos que se passou desde o Big Bang.

Portanto, houve apenas uma evolução nessas máquinas em relação ao tamanho da memória e velocidade do processador, mas o princípio de operação ainda é o mesmo desde o primeiro computador digital. Além disso, esses computadores estão sendo também cada vez mais miniaturizados e é possível que em breve seus componentes cheguem próximos ao tamanho de átomos.

Por falar em átomos, o início do século XX viu surgir uma teoria estranha, que previa que em nível subatômico as partículas não se comportariam como estamos acostumados a pensar em nosso dia a dia. Essa teoria previa, por exemplo, a existência de determinadas partículas que poderiam ter dois estados possíveis, os quais podem ser interpretados, se quisermos, por “0” e “1”, e que poderiam estar tanto no estado 0 quanto no estado 1, mas também poderiam estar nos dois ao mesmo tempo.

Essa teoria inicialmente foi bastante controversa, mas, com o passar do tempo, começou a ser comprovada experimentalmente e hoje, embora muitas de suas partes ainda sejam desconhecidas, ela é aceita como uma explicação coerente com as observações que são feitas nestes elementos subatômicos.

Depois do surgimento e consolidação do computador digital, naturalmente surgiu a hipótese sobre se seria possível usar essas características da mecânica quântica para efetuar computações. Criou-se assim o conceito de qubit ou “quantum bit”, que seria um bit de um hipotético computador quântico capaz de ser 0, 1 ou os dois valores ao mesmo tempo.

Qual a utilidade de um computador assim? Certamente não seria o caso de usá-lo para fazer a folha de pagamento da sua empresa; imagine se o salário pudesse ter sido pago ou não pago ou pago e não pago ao mesmo tempo... Com certeza criaria muita confusão.

Assim, embora a ideia de computador quântico fosse interessante, em princípio, do ponto de vista de curiosidade científica, ainda se está descobrindo quais problemas esse tipo de computador realmente pode resolver melhor do que os computadores digitais.

Em relação à Tese de Turing-Church, foi comprovado que um computador quântico pode ser simulado por uma Máquina de Turing. Assim, um computador quântico não será capaz de computar funções que um computador digital não computa. Ambos têm o mesmo poder de computação. Porém, para alguns problemas, possivelmente, o computador quântico poderá ser muito mais rápido

do que um digital. Os cientistas estão até hoje estudando e descobrindo várias classes de problemas que se encaixam nessa definição.

Genericamente, entende-se que um problema que pode ser mais eficientemente resolvido por um computador quântico do que um digital precisa ter todas as características a seguir:

- A única forma de resolver o problema é adivinhando uma possível resposta e verificando se ela é a verdadeira.
- O número de possíveis respostas para serem verificadas é igual ao número de entradas possível para o problema.
- Toda resposta possível leva o mesmo tempo para ser analisada que as outras.
- Não há nenhum método para estimar respostas que eventualmente são mais promissoras do que as outras, tentar cada uma delas em ordem aleatória é tão bom quanto qualquer outra sequência que se possa imaginar.

Um problema típico que se encaixa nessas características é o de adivinhar a senha de um sistema ou arquivo. A única forma de fazer isso, normalmente, é tentando adivinhar e testando cada uma das senhas possíveis até acertar por acaso. Assim, acredita-se que se os computadores quânticos forem realmente desenvolvidos até o nível suficiente para fazer isso, os sistemas de segurança terão que ser totalmente repensados. Já existem iniciativas nesse sentido. Um computador digital precisaria testar diferentes senhas uma de cada vez, até adivinhar a correta. Isso leva tempo, possivelmente mais tempo do que uma vida humana, mesmo com os computadores mais rápidos. Entretanto, um computador quântico poderia testar praticamente todas as combinações de uma só vez, portanto, em um tempo muito mais reduzido. Na prática, porém, ainda restam dúvidas em relação a essa capacidade realmente poder ser obtida de forma tão simples, visto que os computadores quânticos existentes ainda não foram capazes de obter tal velocidade de processamento.

Os primeiros desenvolvimentos teóricos da computação quântica são atribuídos a Charles H. Bennet (Estados Unidos, 1943), Paul A. Benioff, David Deutsch (Israel, 1953) e Richard P. Feynman. A ideia surgiu quando perceberam que a miniaturização crescente dos microprocessadores acabaria levando a um produto no qual os elementos não seriam maiores do que uns poucos átomos. Esses elementos estariam, portanto, sujeitos às leis da física quântica e teriam que ser entendidos e usados de forma diferente do que os transistores da física clássica.

Feynman, o mesmo que trabalhou no projeto da Connection Machine em 1986, havia produzido um modelo teórico desse computador quântico em 1982. Ele também propôs que uma classe de problemas que seria especialmente adequada para esse tipo de computador consistiria exatamente de modelos de experimentos envolvendo física quântica.

Deutsch, em 1985, baseado no trabalho de Feynman, percebeu que não apenas processos quânticos, mas qualquer processo físico, poderia ser modelado por essa máquina. Ele criou assim o conceito de máquina quântica universal, similar à Máquina de Turing universal.

Pensando em criar um computador desse tipo, Haig Farris, Geordie Rose, Bob Wiens e Alexandre Zagorskin formaram a empresa D-Wave, à qual são atribuídos vários projetos reais de computadores quânticos.

O primeiro deles foi o protótipo Orion, apresentado em 13 de fevereiro de 2007 no Museu de História da Computação em Mountain View, Califórnia. Essa apresentação é considerada como a primeira vez que um computador quântico foi demonstrado ao público.

Porém, a recepção por parte da Academia não foi boa. A ideia de que o computador fosse uma fraude foi levantada por muitos que duvidavam que a empresa realmente tivesse tecnologia para fazer o que alegava. As maiores críticas vieram dos teóricos da física

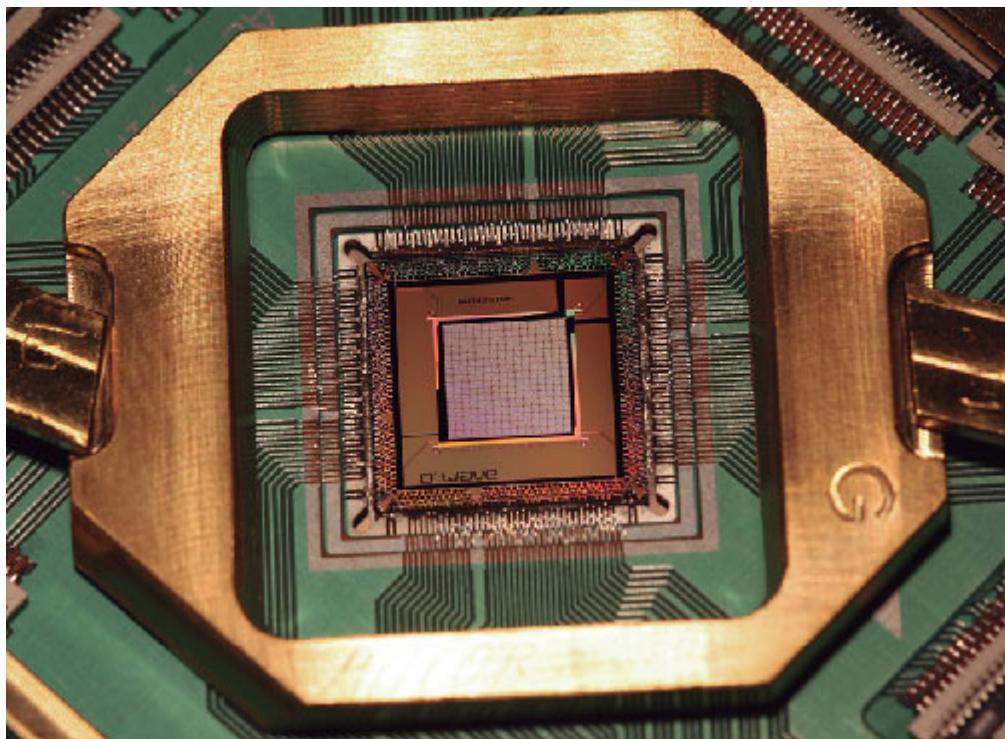
quântica, que chegaram a dizer que o computador era o resultado de um mal-entendido sobre o que seria física quântica.

De fato, não se podia “ver” o que o computador fazia. Ele recebia dados e dava respostas e a empresa garantia que essas respostas eram obtidas através de um algoritmo quântico rodando sobre um hardware de 16 qubits. Infelizmente, na época a empresa também não apresentou detalhes técnicos suficientes para convencer os críticos de que aquilo não era um embuste.

Porém, com o tempo, os funcionários da D-Wave passaram a publicar artigos científicos em periódicos importantes e, com eles, a base científica e tecnológica desse computador apareceram mais claramente, acalmando as críticas.

Além disso, a empresa passou a desenvolver outros modelos do computador. Em 2011 foi anunciado o Rainier, um modelo com 128 qubits, o primeiro computador quântico disponível comercialmente por 10 milhões de dólares.

Em 2012 é lançado o Vesuvius, com 512 qubits e em 2015 o D-Wave 2X, ou Washington, com 1.152 qubits, cujo processador quântico é mostrado na [Figura 11.3](#).



**Figura 11.3: Processador quântico D-Wave de 1.152 qubits.<sup>9</sup>**

A Google e a NASA têm investido bastante dinheiro nos projetos da D-Wave e, apesar das dúvidas que ainda permanecem, a área está em franca expansão, com muitos problemas em aberto aguardando pesquisadores capazes de resolvê-los. A Google afirma que o D-Wave 2X é 100 milhões de vezes mais rápido do que qualquer outro computador atual, pelo menos no que se refere às classes de problemas que descrevemos anteriormente. Isso significa que esse computador pode calcular em alguns segundos respostas para problemas que o computador digital mais rápido levaria 10 mil anos. Esse tipo de máquina talvez possa lançar também uma nova era para a área de inteligência artificial. De fato, a Google, juntamente com a NASA lançaram recentemente um projeto de inteligência artificial quântica.

Talvez possamos comparar o estado da computação quântica hoje mais ou menos ao da computação digital no final da década de 1930. Contudo, espera-se que o crescimento dessa nova tecnologia

seja muito mais rápido, pois, afinal, hoje temos computadores digitais para nos ajudar a projetar computadores quânticos, enquanto que na década de 1930 o que os cientistas tinham eram calculadoras mecânicas e réguas de cálculo.

## 11.12 iPhone – 2007

A Apple, sob o comando de Jobs, mais uma vez surpreendeu o mundo e criou uma revolução tecnológica possivelmente mais impactante do que o lançamento do PC pela IBM. Em 2007, os telefones celulares já tinham algumas características de smartphones, com possibilidade de acesso a e-mail, editor de textos e outras coisas, mas possuíam telas pequenas que dividiam espaço com teclados difíceis de operar e, mesmo quando a tela era sensível ao toque, usava tecnologia resistiva, que exigia pressão e possivelmente o uso de um dispositivo apontador.

Porém, neste ano, Jobs apresenta na conferência e exposição MacWorld em São Francisco o primeiro iPhone. Previamente foi anunciado que Jobs iria apresentar um iPod com tela ampla e sensível ao toque, um telefone móvel revolucionário e um dispositivo de internet totalmente inovador. Entretanto, apenas durante a palestra ele informou que não seriam três dispositivos, mas um único: o iPhone ([Figura 11.4](#)).



Figura 11.4: O primeiro iPhone.<sup>10</sup>

A ideia de um smartphone, como já vimos, foi implementada pela primeira vez pela IBM 12 anos antes, com o Simon Personal Communicator. Entranto, enquanto ele não vingou, o iPhone foi um sucesso. A grande mudança do iPhone em relação a outros smartphones da época, como o BlackBerry, era a integração efetiva da rede móvel em um dispositivo tão pequeno. Passou a ser muito mais fácil acessar a internet e instalar novos aplicativos no telefone. Ele era um computador completo e de grande capacidade na palma da mão. Além disso, acabou definitivamente com o teclado e a caneta apontadora que os concorrentes, BlackBerry, Motorola e Palm usavam em seus smartphones. Agora tudo seria feito com um toque e/ou movimento de um ou dois dedos. Segundo Jobs: “*We are all born with the ultimate pointing device—our fingers—and iPhone*

*uses them to create the most revolutionary user interface since the mouse.”<sup>11</sup>*

Desde então, cada lançamento de um novo modelo de iPhone tem sido um evento cultural, com filas se formando na porta das lojas – algumas vezes dias antes do equipamento estar disponível.

O primeiro modelo do iPhone funcionava exclusivamente com uma linha de dados e voz da AT&T (American Telephone and Telegraph) e usava um sistema de acesso de dados chamado EDGE – bastante inferior ao 3G, que na época já estava disponível. Porém, como o aparelho podia ser conectado em redes Wi-Fi, compensava-se os problemas de velocidade caso uma rede estivesse disponível.

Um dos problemas com o novo sistema para a AT&T foi tornado público quando Justine Ezarik (Estados Unidos, 1984), uma designer gráfica da Pensilvânia, publicou em seu blog um filme de um minuto no qual mostra uma conta telefônica da AT&T com 300 páginas. A empresa de telecomunicações simplesmente não estava preparada para cobrar pelos novos serviços fornecidos pelo iPhone. Depois que o vídeo viralizou na Internet e várias outras pessoas também apresentaram contas com centenas de páginas, a AT&T anunciou que iria criar modelos de conta mais resumidos.

No início (junho de 2007), o iPhone vinha em duas versões, uma com 4 GB de memória vendida a 499 dólares e uma com 8 GB de memória por 599 dólares. Porém, em setembro a Apple resolveu descontinuar o modelo de 4 GB e baixar o preço da versão de 8 GB para 399 dólares. Os compradores dos primeiros iPhones, claro, ficaram furiosos, e a Apple ofereceu a todos um crédito de 100 dólares em produtos de suas lojas e, aos que tinham comprado nos últimos 14 dias, uma devolução de 200 dólares nas suas contas telefônicas.

Em fevereiro de 2008 é introduzido o modelo com 16 GB por 499 dólares e em junho do mesmo ano todos os modelos são descontinuados em função do lançamento do novo iPhone 3G. O 3G foi seguido pelos modelos 3GS em 2009, 4 em 2010, 4S em

2011, 5 em 2012 (que em 2013 passou a ter duas versões), a 5C (mais barata) e a 5S e, finalmente, os mais recentes 6 e 6-plus em 2014, este último praticamente do tamanho de um tablet e 6S e 6S-plus em 2015.

Depois que a Apple lançou o iPhone, todas as demais fabricantes de celulares, palmtops e assemelhados tiveram que se adaptar porque, embora de início elas acreditassesem que a ideia não pegaria ou que o preço era muito alto, o fato é que o público adorou o novo dispositivo e ele, juntamente com seus clones, é hoje parte fundamental de nossas vidas. Há mais smartphones do que pessoas no mundo desde 2014.

No Brasil, o iPhone só foi lançado em 2008 em parceria com a TIM. O mais recente lançamento em um país foi na Sérvia, em outubro de 2014.

## **11.13 Android – 2008**

A guerra do final dos anos 2000 seria a guerra dos smartphones. Assim, apenas cinco meses após o anúncio do iPhone pela Apple, a Google anunciava o Android em novembro de 2007.

Porém, o Android não começou com a Google, pelo menos não bem no início. Em 2003, a empresa Android Inc. foi criada por Andy Rubin (Estados Unidos, 1963), Rich Miner (Estados Unidos, 1964), Nick Sears e Chris White para desenvolver dispositivos móveis que fossem mais amigáveis, por exemplo, sendo capazes de saber a localização do seu portador e suas preferências.

O primeiro produto da empresa teria sido um sistema operacional para câmeras digitais, mas os investidores perceberam que o mercado para esse tipo de produto seria muito pequeno e não aprovaram a ideia. Assim, eles resolveram criar um sistema operacional para o mercado de smartphones que estava crescendo. O projeto todo foi tocado com grande segredo; a única coisa que se sabia fora da Android é que estavam desenvolvendo software para telefones móveis, mas ninguém sabia exatamente o quê.

Em 2004, a empresa estava em maus lençóis porque, sem ter um produto para vender, as contas se acumulavam; inclusive o aluguel do espaço ocupado por ela, que já estava ameaçada de despejo. Em função disso, Rubin ligou para seu amigo dos tempos em que trabalhara na Apple, Steve Perlman e, muito sem jeito, perguntou se ele podia emprestar alguma coisa. Perlman disse que iria transferir algum dinheiro o mais cedo possível, ao que Rubin respondeu que seria melhor se fosse ainda mais cedo do que isso. Perlman foi ao banco e sacou 100 notas de 100 dólares, que entregou a Rubin dentro de um envelope. No dia seguinte, transferiu uma quantia que não foi revelada e a empresa pôde continuar funcionando. Perguntado anos depois do porquê ter emprestado dinheiro a Rubin nessas condições, Perlman alegou que acreditava no negócio e queria ajudar.

Entretanto, a ideia de Rubin talvez nem parecesse tão boa assim; ele estava querendo fazer um sistema operacional aberto para competir em um mercado dominado pelas monolíticas e fechadíssimas empresas de telefonia. Além disso, havia os gigantes da informática como Apple e Microsoft, que abocanhariam qualquer oportunidade de lucrar nesse mercado. E imagine se a IBM resolvesse voltar à briga... O projeto, olhando dessa forma, parecia um beco sem saída: um sistema operacional sem aparelhos para ser instalado. Porém, a Google deve ter descoberto o que eles faziam, porque em 2005 ofereceu 50 milhões de dólares para comprar a empresa.

O caminho para Rubin chegar lá foi longo. Após seu primeiro emprego em uma empresa de microscopia, ele se mudou para a Suíça, onde trabalhou em uma companhia de robótica. Quando estava de férias nas ilhas Cayman encontrou um engenheiro da Apple, Bill Caswell, que havia sido expulso de sua cabana pela namorada após uma briga e não tinha onde ficar. Rubin lhe ofereceu lugar, embora mal o conhecesse. Em agradecimento, Caswell arrumou um emprego para Rubin na Apple.

Ele trabalhou lá de 1989 a 1992, onde sua paixão por robótica ficou tão evidente que lhe rendeu o apelido de “Android”. Arrumou confusão quando, apenas por piada, programou o sistema de telefonia da empresa para mandar mensagens aos seus colegas como se fosse o chefe da Apple, John Sculley (Estados Unidos, 1939), presenteando-os com ações. Após passar por outras companhias, inclusive a Microsoft, Rubin iniciou a sua própria, a Danger, que inventou um sistema que chamou a atenção da Google: o T-Mobile Sidekick. Por um lado, a empresa de Rubin ficou totalmente esquecida nesta história porque, para os usuários, o sistema era vendido pela T-Mobile, e levava o nome dela; Danger sequer existia. A Google sabia, entretanto.



Figura 11.5: T-Mobile Sidekick, fabricado pela Danger.<sup>12</sup>

Larry Page, fundador da Google, marcou um encontro com Rubin quando ele já trabalhava no projeto do Android e ofereceu “ajuda”. Rubin deve ter ficado com a mesma sensação de um rato que recebia oferecimento de amparo de um leão, mas, apesar do pânico, fizeram a reunião.

Rubin e Sears foram até os escritórios da Google e se encontraram com Page e Sergey Brin, cofundadores da Google. Page estava de camiseta e Brin com um relógio do Mickey, se empanturando com as balas de duas jarras sobre a mesa. Porém, não perderam tempo: felicitaram Rubin pelo Sidekick, o qual Page disse ser o melhor telefone que já havia visto.

Brin contou algumas piadas e depois começou a testar Rubin, perguntando detalhes sobre a tecnologia do Sidekick e o que ele faria diferente se fosse melhorar o aparelho, e por que o fizera daquele jeito. A conversa não era agressiva, mas Rubin provavelmente sentiu receio de entregar o jogo dessa forma. De qualquer maneira, era uma aposta: sem novos aportes de recursos, sua empresa não iria muito longe – e recursos era o que a Google tinha de sobra.

Quarenta e dois dias depois, uma nova reunião foi marcada. Dessa vez, os quatro fundadores da Android foram e souberam que a Google queria comprar a sua empresa e contratá-los para trabalhar para eles.

Em julho de 2005, 50 milhões de dólares mais ricos, os antigos donos da Android mudam seus escritórios para dentro do GooglePlex. Porém, a Android não foi absorvida pela Google; era mais como uma startup incubada dentro dela. Tinha seu próprio espaço, o qual era guardado por um boneco em tamanho real de um androide Cylon da série *Galáctica, Astronave de Combate*.

Agora a ideia do Android, que parecia maluca no início tinha alguma chance de dar certo. Só faltava achar um telefone para rodar o sistema e uma empresa de telefonia para operar suas linhas.

O modelo de negócio aberto da Android desde o início consistiria em fornecer o sistema gratuitamente para as fabricantes de smartphones e depois cobrar por serviços de adaptação ou produção de aplicativos. Além disso, as empresas de telefonia poderiam vender propaganda através do software, mas, neste caso, a Android queria uma fatia do lucro.

Porém, antes de terem tudo isso eles precisavam de uma prova de conceito. Assim, a Google produziu seu primeiro protótipo de smartphone, o G1. Ainda em 2007, nenhuma empresa de telefonia queria operar com o aparelho, porém. Queriam manter controle absoluto sobre tudo o que estava relacionado com a telefonia.



Figura 11.6: T-Mobile G1 com Android.<sup>13</sup>

A melhor aposta era a T-Mobile, mas por mais de seis meses eles resistiram. Finalmente, após intervenções de Nick Sears, cofundador da Android, que havia trabalhado para a T-Mobile como executivo de marketing, a empresa aceitou o contrato com a Google.

Finalmente, tinham tudo o que precisavam para lançar o G1, mas então veio uma bomba: Steve Jobs anuncia o iPhone. Rubin, apavorado, mandou o motorista parar o carro para que ele pudesse assistir até o fim o webcast da apresentação de Jobs. Ele então teria dito a seus colegas no carro que a Google provavelmente não iria mais lançar o G1.

A equipe voltou às “pranchetas de desenho”, contudo. O G1 seria um telefone muito mais parecido com um BlackBerry, com teclado deslizante e sem tela sensível ao toque. Com o revolucionário iPhone no mercado, teriam que mudar tudo isso. Felizmente, a Google já tinha projetos em andamento relacionados com telas sensíveis ao toque e não precisou começar do zero neste ponto. Eles até acreditavam que um dia o Android seria adaptado para esse tipo de interface, mas a Apple obrigou esse futuro a vir mais cedo.

Além disso, outro fenômeno muito interessante ocorreu. Como a Apple tinha exclusividade com a AT&T para operar o iPhone, muitas pessoas começaram a trocar de operadora e passar para ela, a única opção de quem quisesse ter um smartphone até então. Por conta disso, outras operadoras começaram a entrar em pânico: não era apenas concorrência, era uma ameaça seríssima ao próprio modelo de negócio. Elas precisavam de uma opção. Essa opção era o Android da Google e todas as empresas de telefonia correram para ele. Como era uma plataforma de software livre, as companhias não ficariam reféns da Google da mesma forma como a IBM ficou da Microsoft durante a guerra dos clones. Possivelmente por este motivo, o Windows Mobile, por outro lado, teve pouquíssima influência no mercado, amargando hoje um terceiro lugar entre os sistemas operacionais mobile.

Porém, foi em 2009 que a primeira ameaça séria ao iPhone surgiu, quando a Motorola anunciou o Droid, um smartphone baseado em Android com teclado deslizante. Tiveram que licenciar o nome do aparelho, pois pertencia a George Lucas, da franquia *Guerra nas Estrelas*.

A primeira versão do Android, 1.0, permitia vários aplicativos rodarem ao mesmo tempo, mas tinha apenas 35 aplicativos disponíveis aos usuários. Tecnicamente, o Android foi baseado no kernel do Linux.

Talvez em função do gosto de Brin por doces (ou não), todas as versões do Android foram apelidadas alfabeticamente com nomes de doces ou sobremesas:

- 1.5 – Cupcake (abril de 2009)
- 1.6 – Donut (setembro de 2009)
- 2.0 – Éclair (outubro de 2009)
- 2.2 – Froyo (maio de 2010)
- 2.3 – Gingerbread (dezembro de 2010)
- 3.0 – Honeycomb (fevereiro de 2011)
- 4.0 – Ice Cream Sandwich (outubro de 2011)
- 4.1 – Jelly Bean (julho de 2012)
- 4.4 – Kitkat (outubro de 2013)
- 5.0 – Lollipop (novembro de 2014)
- 6.0 – Marshmallow (outubro de 2015)

Em 2013, a Google anunciou que a quantidade de dispositivos com Android ativado no mundo atingira a marca de 1 bilhão.

## 11.14 WhatsApp – 2009

A história do WhatsApp começa com um imigrante ucraniano que chegou aos Estados Unidos com 16 anos, Jan Koum (Ucrânia, 1976), nascido Joyce Kikeru. Jan migrou com a mãe e a avó; seu pai teve que permanecer na Ucrânia. Uma vez nos Estados Unidos, sua mãe foi trabalhar como babá e ele, como faxineiro em um mercado. Eles sobreviviam à base de vale-alimentação.

Aos 18 anos, ele ingressou em um curso superior de computação para aprender a programar e iniciou sua carreira como testador de segurança. Em 1997, começou a trabalhar para a Yahoo! como

engenheiro de infraestrutura. Lá ele conheceu Brian Acton (Estados Unidos, 1972), que posteriormente seria seu sócio na fundação do WhatsApp.

Em 2007 ambos saíram de seus empregos na Yahoo! para gastar suas economias viajando pela América do Sul. Posteriormente, em 2009 tentaram um emprego no Facebook, mas ambos foram rejeitados. Ironicamente, 5 anos depois, o Facebook comprou o WhatsApp por 19 bilhões de dólares. Quem sabe se a companhia os tivesse contratado antes, não teria economizado esse dinheiro, mas enfim...

No início de 2009, Koum comprou um dos recém-lançados iPhones e imediatamente percebeu que o modelo de compras da Apple Store seria uma grande revolução na forma como as pessoas passariam a adquirir software. Ele colocou na cabeça que ia fazer um aplicativo importante e procurou seu amigo Alex Fishman para trocar ideias. A concepção do aplicativo inicialmente seria de que usuários de iPhone poderiam escrever um “status” que passaria a ser visto pelos seus contatos. O nome do aplicativo já surgiu nessa reunião: WhatsApp, um trocadilho com a frase “What’s up” e “App”, que é a forma como os aplicativos para smartphones passaram a ser conhecidos. Assim, cada um poderia dizer o que estava pensando, que é a resposta usual para a pergunta: “What’s up?”<sup>14</sup>

Um mês depois, ele cria a empresa com esse nome, mas ainda não tem nenhum aplicativo para vender. Nenhum deles sabia como desenvolver software para o iPhone. Assim, encontraram no site rentacoder.com, atual freelancer.com, um programador russo chamado, Igor Solomennikov, que produziu a primeira versão do WhatsApp.

Tratava-se de uma versão muito instável e Koum estava quase desistindo e arrumando um novo emprego quando Acton o convenceu a tentar por mais alguns meses. Assim, em meados de 2009 a Apple disponibilizou uma funcionalidade aos programadores que permitia enviar notificações aos usuários mesmo que eles não

estivessem usando nenhum aplicativo. Koum aproveitou essa característica nova para fazer com que cada vez que um usuário do WhatsApp mudasse seu status uma notificação (mensagem) fosse enviada a todos os seus contatos. Essa característica foi uma das que começou a chamar a atenção das pessoas, e o número de usuários cresceu bastante depois disso, atingindo rapidamente a marca de 250 mil.

Assim, após vários meses como versão de teste (beta), o WhatsApp é lançado oficialmente na Apple Store em novembro de 2009. Ao contrário de muitos outros aplicativos, ele precisou mudar seu status de gratuito para pago para evitar que o número de usuários crescesse rápido demais. A empresa tinha despesas com envio de mensagens para validar cada novo usuário e não conseguiria pagar a conta de um crescimento muito forte.



Figura 11.7: Logotipo do WhatsApp.<sup>15</sup>

Outro ponto que motivou o crescimento do WhatsApp foi o fato de que desde o início Koum era totalmente contra a veiculação de propaganda e joguinhos no aplicativo. Uma nota escrita à mão sobre sua escrivaninha estabelece: “*No Ads! No Games! No Gimmicks!*”<sup>16</sup>

Ainda outro ponto atrativo é o fato de que o aplicativo não exige informações do usuário como nome, idade, endereço etc. Basta ter um número de telefone autenticado. Essa abordagem, completamente na contramão de outros aplicativos, é atribuída aos anos de repressão que Koum passou na Ucrânia comunista, onde telefones frequentemente eram grampeados e as pessoas

precisavam cuidar muito de sua privacidade por causa da polícia secreta.

No final de 2009, o aplicativo passou a permitir o envio de fotos e em 2011 já era um dos 20 mais baixados na Apple Store. Em 2011, a Sequoia Capital estava procurando os autores do WhatsApp para oferecer um financiamento de 7 milhões de dólares. Só que o app não disponibilizava o endereço de contato dos autores, apenas a informação de que a sede da empresa ficava em Mountain View na Califórnia. Assim, vários funcionários da Sequoia foram literalmente vasculhar a cidade em busca da empresa e de seus criadores para oferecer o investimento. O fato de não terem uma placa na frente da empresa também não ajudou, mas no final houve o encontro e o financiamento.

A partir de 2013, o WhatsApp já era o aplicativo de comunicação mais popular do mundo. Entre outras coisas, isso aconteceu porque ele permitia enviar mensagens curtas a contatos sem pagar nada a mais, como acontecia no caso dos SMS. Nessa época, Mark Zuckerberg, dono do Facebook, começou a marcar conversas com Koum, inicialmente ventilando alguma forma de cooperação, mas finalmente dando a conhecer que ele queria, de fato, comprar a empresa. Assim, no início de 2014 o WhatsApp foi comprado pelo Facebook pela quantia de 19 bilhões de dólares. Isso fez com que muitos usuários procurassem outros aplicativos de comunicação, como o Telegram, por exemplo.

Dois meses depois, o Irã baniu o WhatsApp com a seguinte alegação “*The reason for this is the assumption of WhatsApp by the Facebook founder Mark Zuckerberg, who is an American Zionist.*”<sup>17</sup> Entretanto, o regime totalitário possivelmente está mesmo é preocupado com a possibilidade de seus cidadãos, especialmente os mais jovens, trocarem informações livremente com pessoas de outros países.

Em janeiro de 2016 o WhatsApp contava com 990 milhões de usuários, e o Facebook alegava planejar chegar a 3 bilhões nos

próximos anos. Para isso, neste mesmo mês, o Facebook anunciou que não ia mais cobrar a taxa anual de 1 dólar para evitar que usuários sem cartão de crédito tivessem que abandonar o aplicativo. Assim, ele voltava a ser totalmente gratuito, como no início.

## 11.15 Waze – 2009

Existem algumas tecnologias que surgem e são bastante usadas por um tempo até que alguma nova tecnologia as torne totalmente obsoletas. Foi isso o que aconteceu com os aparelhos de navegação por GPS a partir do surgimento do aplicativo Waze para smartphone. Os aparelhos de GPS são capazes de indicar caminhos e podem até estimar a velocidade média de cada estrada, mas como não são capazes obter e enviar informações através da internet, eles não avisam, por exemplo, se há um congestionamento à frente.

Já o Waze ([Figura 11.8](#)) é um app que foi pensado para ter, além das informações tradicionais das estradas, outras que são enviadas pelos próprios usuários – tanto explicitamente, como quando um usuário reporta um acidente ou via fechada; quanto implicitamente, pois o Waze acompanha a velocidade de cada usuário e usa essa informação para decidir se uma determinada via está sofrendo de condições anormais de trânsito.



Figura 11.8: Waze.<sup>18</sup>

Assim, esse aplicativo acaba sendo muito mais útil do que um aparelho de GPS. Ele é gratuito e pode ser instalado em qualquer smartphone.

A história do Waze começa em 2006 em Israel, quando Ehud Shabtai criou o projeto FreeMap Israel. O objetivo era produzir um mapa preciso do país a partir de dados enviados por uma comunidade de usuários. Essa base ficaria à disposição do público de forma livre e sem qualquer custo. Em 2008 o projeto muda seu

nome para “Waze”, e em 2009 a Waze Mobile Ltd. se torna uma empresa.

A empresa conseguiu crescer com investimentos até 2011, quando iniciaram a venda de anúncios. O seu modelo de propaganda é localizado, ou seja, quando o usuário passa fisicamente perto de um anunciante ele é informado sobre o estabelecimento. Por exemplo, ao se aproximar de uma pizzaria que é anunciante do Waze, o aplicativo vai mostrar no mapa a localização do lugar e perguntar se você não quer dar uma passada lá. Se você concordar, basta dar um clique na tela e o aplicativo instruirá você sobre como chegar lá.

O Facebook e outras empresas manifestaram interesse em comprar o Waze, mas foi a Google que chegou no preço: 1,1 bilhão de dólares em 2013. A Google então passou a adicionar a informação coletada com a comunidade de wazers em seu sistema de mapas.

Mais recentemente outro aplicativo muito interessante surgiu, o Moovit, que já é chamado por alguns de “Waze do transporte público”. Com o Moovit, você diz aonde quer ir e recebe em resposta alguns itinerários de ônibus, indicando local da parada, horário, onde fazer baldeação, se for o caso, e que horas vai chegar no seu destino. O Moovit também foi criado em Israel. Uri Levine, um dos fundadores do Waze, faz parte do conselho de diretores. Segundo informações, o país com o maior número de usuários do Moovit atualmente é o Brasil.

## **11.16 Até Aqui...**

Os primeiros anos do século XXI viram surgir uma computação com a qual Babbage, von Neumann e mesmo Vannevar Bush possivelmente nem sonhavam: máquinas minúsculas, que cabem em um bolso, com mais capacidade de processamento do que um supercomputador Cray-2.

A área de produção de software nesse período dá uma guinada na direção de métodos ágeis, afastando de muitos projetos o

engessamento de estruturas rígidas de processos que são comuns em outras indústrias.

Alguns produtos também mudaram paradigmas: a Wikipédia dominou as enciclopédias, o Skype mudou a telefonia, o YouTube acabou com as videolocadoras, o Facebook criou uma gigantesca comunidade global e o WhatsApp liquidou com as mensagens tarifadas de SMS.

Foram muitas mudanças nessa época, mas o progresso continua. Talvez o leitor até se surpreenda por perceber que determinadas tecnologias que parecem estar por aí há tanto tempo são tão recentes quanto a década passada.

---

<sup>1</sup> Definições sobre como os dados são apresentados na interface de um sistema.

<sup>2</sup> © Microsoft. Used with permission. Disponível em: <https://en.wikipedia.org/w/index.php?curid=12215927>

<sup>3</sup> By Bjørn Smestad according to this archived page of submitted logos from Nupedia site - as is seen in that page at the bottom “Logo ideas posted with permission of their designers. Click on a logo to go to its designer’s website.” Image link in that page is pointing to this archived page: “Assistant Professor Bjørn Smestad”. previously attributed to Stephen Gilbert. Actually, User Stephen Gilbert is not the author of the logo, as he himself stated on 17 July 2007 (see statement) (\*) - Uploaded to english wikipedia 11:00, 28 March 2005 by Mark, CC BY-SA 3.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=1115422>

<sup>4</sup> Tradução: “Espere... de que vale construir esses relacionamentos se este é o resultado?”

<sup>5</sup> Tradução: “apenas configurando meu twtr.”

<sup>6</sup> Tradução: a Baleia da Falha.

<sup>7</sup> Tradução: “Olá, Twitterverso! Nós estamos AO VIVO da Estação Espacial Internacional – o primeiro tweet ao vivo do espaço! :) Mais adiante, mande suas perguntas.”

<sup>8</sup> Tradução: “Nós gostaríamos de agradecer com 140 caracteres ou menos. E nós acabamos de fazer isso!”

<sup>9</sup> “*D-Wave-Washington-1000Q*” by *Mwjohson0* - Own work. Licenced under CC BY-SA 4.0 via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:D-Wave-Washington-1000Q.jpg#/media/File:D-Wave-Washington-1000Q.jpg>

<sup>10</sup> By *DanielZanetti* - Own work, CC BY-SA 3.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=8731337>

<sup>11</sup> Tradução: “Nós todos nascemos com o dispositivo apontador definitivo – nossos dedos – e o iPhone os usa para criar a mais revolucionária interface com o usuário desde o mouse.”

<sup>12</sup> © David Mueller. CC BY-SA 2.5. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=782298>

<sup>13</sup> By *Luis Alberto Arjona Chin* - originally posted to Flickr as *HTC Android T-Mobile G1*, CC BY 2.0. Disponível em: <https://commons.wikimedia.org/w/index.php?curid=4856009>

<sup>14</sup> Tradução: “E aí?”

<sup>15</sup> “WhatsApp” by WhatsApp - Derivative of WhatsApp\_logo.svg. Licenced under Public Domain via Wikimedia Commons. Disponível em: <https://commons.wikimedia.org/wiki/File:WhatsApp.svg#/media/File:WhatsApp.svg>

<sup>16</sup> Tradução: “Nada de anúncios! Nada de jogos! Nada de truques!”

<sup>17</sup> Tradução: “A razão para isto é o fato de que o WhatsApp foi assumido pelo fundador do Facebook, Mark Zuckerberg, que é um sionista americano.”

<sup>18</sup> © Waze Mobile. Cortesia.

## **PARTE XII**

### **O Presente e o Futuro**

Finalmente chegamos ao presente. O mundo atual é dominado pela tecnologia. Os smartphones são cada vez mais onipresentes e seus aplicativos têm mudado muito a forma como fazemos as coisas. Hoje, você não vai mais a videolocadoras nem assiste aos canais de TV, nem mesmo à TV por assinatura; hoje você assiste filmes diretamente na internet em alta qualidade. Hoje você não compra mais um PC; você coloca um dispositivo do tamanho de um pendrive na sua TV em alta definição e ela vira um computador instantaneamente.

As empresas de telefonia estão precisando mudar seus modelos de negócio porque cada vez menos as pessoas se telefonam; mensagens de texto e voz sobre internet são dominantes. Mesmo os provedores de internet precisam repensar seriamente seu modelo porque a ameaça de uma rede wireless mundial e gratuita vai destruir suas atuais fontes de renda.

Os aparelhos de GPS da década passada, que tanto ajudaram os motoristas a se locomover, se tornaram obsoletos devido à adoção maciça de aplicativos como o Waze, que, pelas informações coletadas de uma comunidade de usuários, permite saber em tempo real quais as condições do trânsito. O EasyTaxi e o 99Taxis localiza o táxi mais próximo de você e o Uber proporciona um serviço diferenciado (quando os taxistas deixam).

O iFood lhe permite comprar comida pronta sem falar com um atendente, que muitas vezes anotava errado seu pedido. Hoje, as interfaces de voz como Siri e Google Now permitem que as pessoas conversem com seus smartphones pedindo, recebendo instruções e realizando atividades.

Além disso, a inteligência artificial começa a tomar forma com novas máquinas e softwares como os implementados no Watson da IBM, que venceu os maiores campeões de Jeopardy! de todos os tempos.

A década mal iniciou e tecnologias promissoras como o Google Glass já falharam, enquanto que outras como as tecnologias vestíveis no estilo do Apple Watch estão apenas no início. É difícil saber em tão pouco tempo o que ainda será história nessa década. Talvez ao final dela descubramos coisas do início da década que foram importantes, mas hoje ainda não entendemos, como novas linguagens de programação, por exemplo, que poderão se tornar dominantes em 5 ou 10 anos.

Enfim, o capítulo termina com alguns exercícios de futurologia nos quais, baseados nas previsões de um site especializado, apresentamos um panorama das possíveis evoluções da computação nos próximos anos, décadas e séculos.

## 12.1 Chrome OS – 2010

Em setembro de 2008, a Google lançou seu próprio browser: o Google Chrome. Isso aconteceu logo após o vazamento de uma história em quadrinhos<sup>1</sup> que apresentava o navegador e toda a motivação para sua construção. Entre outras coisas, a Google argumentava que o que as pessoas mais faziam na Web era assistir vídeos, conversar e jogar, e que nenhuma dessas atividades era usual quando os primeiros navegadores foram construídos. Eles foram feitos para navegar em páginas com textos e eventualmente imagens. Assim, a ideia do Google Chrome era de um navegador muito leve, rápido e adequado para essas novas atividades.

No ano seguinte, a Google anunciou um projeto ainda mais ousado, o Chrome OS, um sistema operacional diferente, no qual tanto as aplicações quanto os dados do usuário ficariam na rede, ou nuvem, e não mais no computador do usuário. Assim, o Chrome OS seria um sistema operacional especialmente dedicado a aplicações que rodam na Web e seu hardware alvo seriam inicialmente os netbooks, ou seja, notebooks de baixo custo com baixa capacidade de armazenamento local, mas com boa conectividade em rede. A aparência do sistema também seria a de um browser e não mais de desktop. A [Figura 12.1](#) mostra dois computadores rodando Chrome OS.



**Figura 12.1: Computadores rodando Chrome Os.<sup>2</sup>**

O sistema, alega-se usualmente, foi baseado no kernel do Linux, embora haja quem conteste essa afirmação. O projeto foi conduzido por Sundar Pichai (Índia, 1972). A data de lançamento oficial foi 11 de novembro de 2010.

Os primeiros notebooks baseados em Chrome OS, os Chromebooks, foram lançados em 2011. O Chrome OS não é vendido isoladamente: ele apenas vem instalado no hardware de produtos feitos por parceiros da Google. Em 2014, a Samsung lançou um desktop baseado no sistema chamado “Chromebox”. Também em 2014, a LG integrou o hardware do Chromebox em um monitor com câmera e alto-falantes embutidos. O novo dispositivo é chamado de Chromebase. A Acer lançou uma versão do Chromebase com tela sensível ao toque.

Em 2015 foi anunciado o Chromebit, um dispositivo do tamanho de um pendrive que pode ser ligado à entrada HDMI de uma TV e com isso a TV se transforma em um computador, com um custo de apenas 85 dólares. Teclado e mouse sem fio podem ser adicionados. Os Chromebits passaram a ser vendidos em novembro de 2015.

Há uma versão livre do sistema operacional chamada Chromium OS que pode ser baixada e compilada livremente. Assim, Chromium OS é mantido por uma comunidade aberta enquanto que o Chrome OS é suportado apenas pela Google.

A partir de 2014 foi oferecida a possibilidade de rodar aplicativos Android no Chrome OS, permitindo, assim, alguma compatibilidade entre as duas plataformas. O fato de a Google ter dois sistemas operacionais foi criticado por algumas pessoas, entre elas, Steve Ballmer (Estados Unidos, 1956), CEO da Microsoft, que alegou que a Google não era capaz de se decidir. A Google afirmou que ter dois sistemas operacionais competitivos no mercado era algo que muitas outras empresas iam querer. Sergey Brin, cofundador da Google, também afirmou que em determinado momento os dois sistemas acabariam convergindo. E aparentemente isso é mesmo uma decisão da empresa porque em 2013, Andy Rubin foi removido do controle do Android e ambos os sistemas passaram a estar sob a direção de Pichai. Já foi anunciado que até 2017 os dois devem ser combinados em um só, possivelmente com o Chrome OS embutido

no Android. O sistema resultante deverá ser um Android que também roda em notebooks.

## 12.2 Instagram – 2010

O Instagram é um aplicativo gratuito criado em 2010 por Kevin Systrom (Estados Unidos, 1983) e Mike Krieger (Brasil, 1986). Ele permite que usuários tirem fotos, apliquem filtros e as disponibilizem em redes sociais, inclusive a rede própria do Instagram. Em homenagem às antigas máquinas Kodak Instamatic e Polaroid, eles inicialmente mantinham as fotos dentro do mesmo formato quadrado que elas usavam.

Uma das características especiais do Instagram que o tornou um grande sucesso embora outros sistemas semelhantes já existissem antes, foi o fato de que os filtros aplicáveis às fotos fazem com que elas pareçam profissionais, mesmo quando tiradas por amadores.

Outra característica é a ênfase na possibilidade de compartilhar as fotografias no exato momento em que são batidas com o smartphone. Além disso, é possível compartilhá-las, além de no próprio Instagram, nas seguintes redes: Facebook, Twitter, Tumblr, Flickr e Foursquare, tudo isso sem precisar entrar diretamente nelas.

O nome “Instagram” veio da conjunção das expressões “instant camera” e “telegram”, ou seja, é uma aplicação para envio de fotos como se fossem telegramas. O mote da empresa é “Fast beautiful photo sharing”.<sup>3</sup>

Iniciado em 2010, em menos de dois meses o Instagram atingiu 1 milhão de usuários. Ele cresceu tanto que em 2011 já contava com 10 milhões de utilizadores e mais de 100 milhões de fotos carregadas. Em 2012, o aplicativo já contava com 30 milhões de usuários e chamou a atenção do Facebook, que o comprou por 1 bilhão de dólares. A partir de então, seu crescimento tornou-se ainda mais intenso.

Porém, muitos viram essa compra como um mau negócio, afinal o Instagram ainda não tinha um modelo definido e, apenas sete anos antes, a Yahoo! havia comprado o Flickr, um sistema com objetivos similares, por apenas 35 milhões de dólares.

No final de 2012, o Instagram perdeu boa parte de seus usuários quando mudou os termos de uso do sistema, atribuindo a si o direito de vender para terceiros as imagens dos usuários sem aviso nem notificação. A reação negativa dessa iniciativa fez com que a empresa voltasse atrás.

Em 2013, o Facebook começou uma campanha de anúncios com aparência natural. Como os usuários do Instagram não estavam acostumados com anúncios chamativos e piscantes, como acontece em outros aplicativos, eles experimentaram adicionar fotos de alta qualidade e beleza com referências discretas às marcas que queriam divulgar. A iniciativa provou ser um sucesso porque mais de 5% dessas fotos receberam “curtidas” dos usuários.

Em 2014, o Instagram foi a rede social que mais rapidamente cresceu no mundo, com um aumento de 23% em apenas seis meses, superando em muito a expansão do próprio Facebook, que ficou em 3%. Nesse ano, ele também ultrapassou o Twitter em número de usuários mobile.

Desde 2015 o aplicativo permite a captura de fotos de quaisquer dimensões, não apenas quadradas. Além disso, agora ele possibilita o compartilhamento de vídeos de até 15 segundos.

## 12.3 PewDiePie – 2010

O que você faria se seu filho anunciasse que ia largar o curso de Economia e Engenharia Industrial em uma universidade de primeira linha para se dedicar a jogar videogame, gravar vídeos de si mesmo jogando e postar esses vídeos na Internet? Certamente cortaria a mesada dele, não? Pois foi o que aconteceu quando Felix A. U. Kjellberg (Suécia, 1989) disse exatamente isso aos seus pais.

Um ano antes, ele estava entediado e começou a jogar e gravar esses vídeos, criando uma conta no Youtube chamada “PewDie”. “Pew” era referência ao barulho que fazem as armas laser nos jogos e “die” é “morrer” em inglês, coisa que acontecia com muita frequência nos jogos preferidos dele. Algum tempo depois, ele esqueceu a senha da conta e criou uma nova, que dessa vez chamou de “PewDiePie” (pronuncia-se “piu-di-pai”). O “pie” pode ter vindo do fato de gostar de tortas, mas ele alega que apenas usou a palavra sem nenhuma razão particular.

Os pais, como qualquer um de nós, cortaram a mesada do garoto e ele passou a trabalhar em uma banca de cachorros-quentes para se manter enquanto gravava e disponibilizava seus vídeos.

Seu humor e autenticidade fizeram dele um sucesso no Youtube. Os fãs comentam que ele se comporta nos vídeos como se estivesse na sala com eles, como velhos amigos, jogando videogames e fazendo brincadeiras uns com os outros. Nas palavras do rapaz: *“YouTube breaks the barrier between the audience and the creator. They feel a connection to the one they’re watching.”*<sup>4</sup>

Em 2012, ele chegou a 1 milhão de assinantes no seu canal Youtube e, desde 15 de agosto de 2013, trata-se do canal com o maior número de assinantes, tendo sido ultrapassado por menos de dois meses apenas por um do próprio Youtube, o Spotlight. Logo retomou a liderança e hoje não há artista, banda, ator, franquia, nada que alcance a sua popularidade no Youtube. É o canal mais visto na história, tendo passado de 10 bilhões de visualizações em dezembro de 2015.

Ele trata seus fãs como “bros”<sup>5</sup> ou “bro army”<sup>6</sup>, e seu cumprimento característico é aproximar o punho fechado da câmera como que dando o cumprimento do soco no punho. Assim, ele escolheu como logotipo um “P” dentro do qual está a imagem de um punho, o “profist”, azul.

Em termos monetários, PewDiePie se deu muito bem para um jovem de vinte e poucos anos, tendo faturado 4 milhões de dólares

em 2013, 7 milhões em 2014 e 12 milhões em 2015, muito do que ele converte em ajuda a instituições de caridade. Quando seus rendimentos de 2014 se tornaram conhecidos por toda a mídia, ele sentiu a necessidade de fazer uma justificativa aos seus fãs. Em linhas gerais, diz que muitos consideram injusto ele passar o dia gritando para o monitor e ganhar milhões com isso – e que, de fato, era verdade. Como ele possui bilhões de visualizações e o Youtube vende propaganda durante seus vídeos, isso acabou retornando para ele como consequência. Porém, desde o início ele nunca tinha pensado nisso como carreira, até porque ninguém nunca havia ganhado dinheiro decente jogando vídeo games – mas aconteceu.

Atualmente ele vive na Inglaterra porque, segundo ele, a velocidade da rede lá é melhor. Em relação aos games que ele joga, muitos poderiam pensar que estaria talvez infringindo direitos autorais das empresas produtoras ou ainda oferecendo “spoilers” (ou seja, entregando o final dos games e com isso estragando a experiência das pessoas que fossem jogar). Porém, de forma totalmente imprevista, aqueles que assistem aos seus vídeos (a maioria muito jovem) adquirem interesse em procurar pelos mesmos jogos, mesmo havendo os spoilers. Assim, muitas empresas produtoras de games fazem fila na porta dele para que jogue e, portanto divulgue, os seus próprios produtos.

Sobre ter largado a faculdade, ele disse o seguinte recentemente: *“Thinking about it now, it was utterly absurd.”*<sup>7</sup>

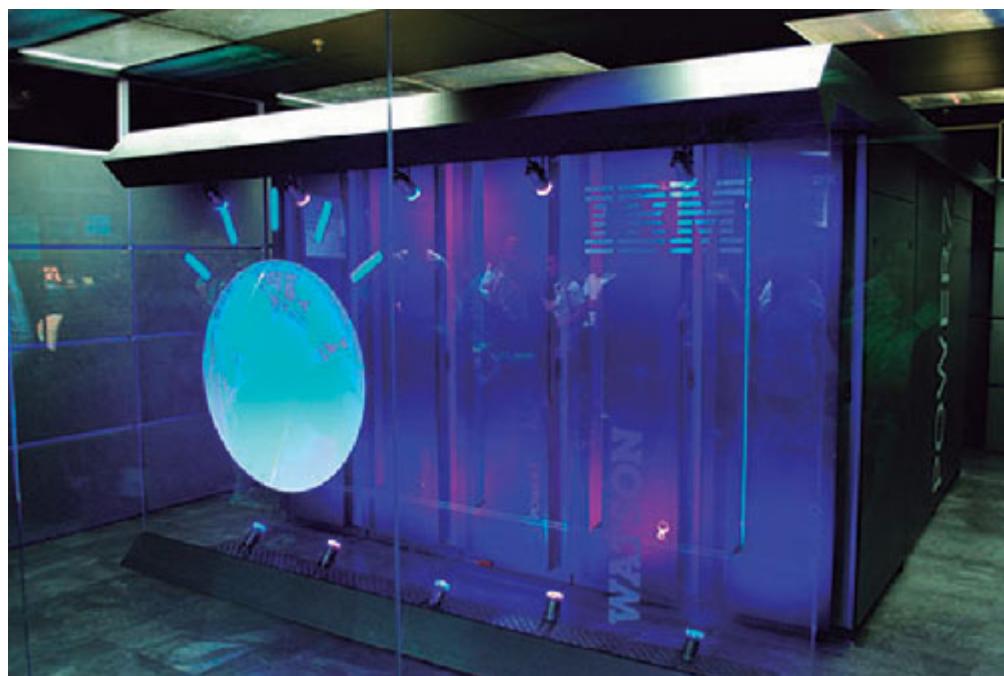
## 12.4 Watson – 2011

Já faz algum tempo que não falamos de computadores de grande porte, nem da IBM. Ela pode ter saído do mercado de computadores pessoais e mal ter molhado os pés no mercado de smartphones, mas no que se refere a grandes máquinas e aos limites da computação, a empresa continuava vencendo desafios.

Um destes desafios foi completado em 2011, quando uma máquina chamada “Watson”, em homenagem ao pai fundador Thomas John

Watson Sênior, venceu dois campeões de Jeopardy! ao vivo na TV. Jeopardy! é considerado um jogo que exige muito conhecimento, cultura e raciocínio. Os participantes são apresentados a pistas na forma de informações, e sua resposta deve ser dada na forma de uma pergunta que corresponda às respostas. Por exemplo, se uma das pistas for “Descobriu o Brasil em 1500” o jogador, para ganhar, deveria responder com a pergunta “quem foi Pedro Álvares Cabral?”.

O Watson, mostrado na [Figura 12.2](#) é um computador especializado em perguntas e respostas, uma máquina ao estilo *Jornada nas Estrelas*: você pergunta em língua natural e ele responde na mesma linguagem. Foi desenvolvido como parte do projeto DeepQA da IBM, liderado por David Ferrucci.



**Figura 12.2: IBM Watson.<sup>8</sup>**

O Watson foi alimentado com cerca de 200 milhões de páginas de conhecimento estruturado e não estruturado, contendo, por

exemplo, todo o conteúdo da Wikipédia. Esse material fica armazenado em 4 TB (terabytes) de disco.

A máquina começou a ser projetada cinco anos antes, em resposta a um dos grandes desafios lançado pela IBM, que tinha por objetivo lidar com problemas complexos, mas compreensíveis pelo ser humano comum, como, por exemplo, jogar Jeopardy!. O objetivo da empresa com esse tipo de projeto era atrair mais jovens para a área de tecnologia.

Watson é um computador maciçamente paralelo, com um cluster de 90 processadores IBM Power7 de 3,5 GHz, cada qual com oito núcleos de processamento e cada núcleo com quatro processos paralelos, totalizando uma capacidade de lidar com 2.880 processos em paralelismo. Sua memória RAM é de 16 TB, o que permitiu que todo o conhecimento necessário para jogar Jeopardy! fosse colocado em memória principal, pois se ficasse em disco, os tempos de acesso mais lentos poderiam ter feito a máquina perder para os jogadores humanos.

Para quem viu o Sr. Data folheando livros a uma velocidade incrível nos filmes de *Jornada nas Estrelas – A Nova Geração*, saiba que se considera que o Watson é capaz de processar a informação contida em 1 milhão de livros em um único segundo.

No Jeopardy!, Watson bateu Ken Jennings, maior campeão de todos os tempos com 74 vitórias, e Brad Rutter, que ganhou o maior prêmio individual de 3,25 milhões. Jennings estava confiante antes do jogo, porque havia assistido a algumas aulas sobre inteligência artificial e tinha certeza de que nenhum computador teria as habilidades necessárias para vencer nesse desafio. Porém, após o programa ele comentou o seguinte: “*I felt obsolete. I felt like a Detroit factory worker in the ‘80s seeing a robot that could now do his job on the assembly line. I felt like ‘Quiz Show Contestant’ was now the first job that had become obsolete under this new regime of thinking computers.*”<sup>9</sup>

Como Watson era muito grande e barulhento para ficar no estúdio com os demais jogadores, ele foi substituído por um avatar, um globo terrestre cortado por 42 linhas lembrando órbitas de satélites. O uso do número 42 aqui também foi proposital e uma referência ao livro *O Guia do Mochileiro das Galáxias*, no qual um computador gigantesco, Deep Thought,<sup>10</sup> apresentava “42” como a resposta à questão fundamental da vida, universo e tudo o mais. Como no jogo de Jeopardy!, o que faltava para a resposta fazer sentido era saber qual era a pergunta. Essa, infelizmente, Deep Thought não sabia informar.

O logotipo, normalmente azul, ficava verde quando o Watson recebia um problema espinhoso e laranja quando dava uma resposta errada.

Nem sempre, claro, suas respostas foram corretas. Quando deram a pista “*Its largest airport is named for a World War II hero; its second largest for a World War II battle*”,<sup>11</sup> ele cometeu alguns erros de interpretação, pois embora a pista em si não deixasse claro, o tópico da questão era “Cidades dos EUA”. Ele não levou isso em consideração e acabou respondendo “*What is Toronto?????*”, uma cidade canadense. A quantidade de pontos de interrogação deve-se ao fato de que não tinha muita certeza de que essa era a resposta correta; seu nível de confiança era de apenas 14%. Logo abaixo disso, a segunda resposta, com 12%, era a correta: “Chicago”. Infelizmente, só valia a primeira.

O período de negociação entre a IBM e a produção de Jeopardy! não foi tranquilo. Uma das maiores barreiras, que no final foi quebrada, foi o fato de que a IBM desconfiava que os produtores de desafios para o programa poderiam explorar o fato de a máquina não ter o conhecimento do senso comum dos seres humanos, mas apenas trabalhar com símbolos. Se questões envolvendo senso comum fossem usadas, o programa seria mais semelhante a um Teste de Turing, e esse era ainda um passo muito grande para um computador. Para evitar esse problema, as partes concordaram que

os desafios usados seriam sorteados dentre aqueles que foram escritos para programas anteriores, porém não usados no ar.

Após perder para Watson, Jennings fez piada com uma cena de *Os Simpsons*, escrevendo junto com sua resposta final “*I, for one, welcome our new computer overlords*”<sup>12</sup>. Na cena original do desenho animado, o âncora do jornal, Kent Brockman, diz uma frase parecida referindo-se aos insetos gigantes que ele pensava estarem invadindo a Terra: “*I, for one, welcome our new insect overlords*.”<sup>13</sup> Porém, no final Jennings foi mais filosófico: “*I had a great time and I would do it again in a heartbeat. It's not about the results; this is about being part of the future.*”<sup>14</sup>

Porém, o Watson não foi construído apenas para ganhar 1 milhão de dólares (que foram doados à caridade) em um jogo de TV. Ele é hoje aplicado em áreas onde um amplo conhecimento ajuda a responder certas questões. Uma dessas áreas é o auxílio a profissionais que precisam planejar tratamentos médicos, especialmente no caso de câncer. Não se trata de um sistema de diagnóstico, mas de um sistema que propõe os melhores tratamentos para pessoas com doenças já diagnosticadas. Em relação à versão do Watson que foi usada no Jeopardy! em 2011, a mais atual é 240% mais rápida e ocupa apenas 1/16 do seu espaço: indo do tamanho de uma suíte-máster para as dimensões de uma gaveta de meias.

O Watson voltado para a medicina foi ensinado e treinado com inúmeras informações sobre tratamentos médicos, especialmente na área de oncologia, e reproveitou muita da informação usada no Jeopardy!, como, por exemplo, a Wikipédia. Porém, nem toda a base de conhecimento usada no programa pôde ser aproveitada. O Dicionário Urbano (Urban Dictionary), por exemplo, foi retirado da base de conhecimento do Watson quando ele respondeu a uma pergunta de um pesquisador com “bullshit!” (forma pouco respeitosa de dizer “besteira!”). Assim, ele tem sido alimentado com publicações médicas e outras revistas científicas.

A IBM afirma que o Watson tem atualmente o conhecimento equivalente ao de um estudante de Medicina do primeiro ano e espera em alguns anos que o computador seja aprovado no exame geral de licenciamento médico, equivalente ao nosso exame da Ordem dos Advogados, mas para a área de medicina, que existe nos Estados Unidos. Outra área em que o computador está sendo testado é a de investimentos financeiros.

## 12.5 Siri – 2011

Siri (Speech Interpretation and Recognition Interface<sup>15</sup>) é um aplicativo para iPhone que permite que um usuário faça perguntas em voz alta e receba respostas na forma de voz ou através do acionamento dos aplicativos do smartphone. Não é um sistema semelhante ao Watson, que procura ser capaz de responder quaisquer questões de conhecimento geral ou específico, mas um sistema prático, voltado para ajudar usuários de smartphones a localizar restaurantes, fazer ligações, marcar eventos na agenda ou obter informações simples usualmente disponíveis em aplicativos. Seus objetivos primários são ser uma interface de conversação com consciência de contexto pessoal e capaz de delegar serviços. Por exemplo, você pode dizer a Siri: “Minha esposa é fulana de tal.” Dias depois, você diz a Siri: “Ligue para minha esposa.” Siri vai olhar na sua agenda telefônica, qual o telefone de “fulana de tal” e vai ativar o aplicativo adequado que vai ligar para este telefone. Se você perguntar por um hotel, ela vai consultar um serviço de reservas, se pedir instruções de direção, vai buscar um aplicativo de navegação, e assim por diante. Em seu lançamento, Siri já se comunicava com 42 diferentes serviços via Web.

O projeto Siri originalmente surgiu no centro de inteligência artificial do SRI International como projeto CALO (Cognitive Assistant that Learns and Organizes<sup>16</sup>) financiado pelo departamento de defesa norte-americano. O conceito foi estabelecido numa época em que muitos diziam que inteligência artificial era uma perda de tempo,

visto que muitas iniciativas nessa área tinham falhado miseravelmente nos anos anteriores. Assistentes pessoais baseados em interpretação de voz eram considerados o canto da sereia no Vale do Silício, pois atraíram várias empresas para o naufrágio.

A iniciativa reuniu várias áreas independentes da inteligência artificial que isoladamente já eram extremamente complexas e, em fazendo isso, criou novas tecnologias de aprendizagem de máquina, enfatizando inclusive a aprendizagem “*in vivo*” no lugar da aprendizagem “*in vitro*”, ou seja, a máquina que aprende à medida que realiza suas tarefas. Anteriormente, ela normalmente só aprendia quando um conjunto de dados especialmente preparados eram apresentados a ela em grandes lotes.

Durante o projeto, o assistente que estava sendo criado foi apelidado de HAL, em referência ao computador de 2001: *Uma Odisseia no Espaço*. No entanto, o seguinte lema procurava esclarecer as coisas: “*HAL’s back – but this time he’s good.*”<sup>17</sup>

Inspirada nesse projeto, foi criada a empresa Siri, Inc., fundada em 2007 por Dag Kittlaus (Noruega, 1967), Adam Cheyer e Tom Gruber (Estados Unidos, 1959). O aplicativo foi inicialmente disponibilizado na Apple Store, e versões para BlackBerry e Android foram anunciadas. O nome Siri foi dado por Kittlaus e, apesar da semelhança com a sigla SRI, não se deveu a ela. Quando sua esposa ficou grávida, ele quis batizar a criança, se fosse menina, como “Siri”, que era o nome de uma antiga colega de trabalho. Em norueguês, “Siri” é abreviação de “Sigrid” e significa “bela vitória”. Ele até registrou o domínio Siri.com. Quando sua esposa deu à luz um menino, ele esqueceu a ideia por um tempo. Porém, quando resolveu criar uma empresa para desenvolver a tecnologia de assistente pessoal, o domínio estava registrado, o nome era fácil de pronunciar e a Siri Inc. foi criada.

A Apple foi rápida e comprou a empresa em 2010 por 200 milhões de dólares, cancelando todos os projetos dela que não fossem

relacionados ao iPhone, inclusive um contrato que já estava assinado para transformar Siri em um aplicativo padrão dos sistemas Android. Na verdade, levou apenas três semanas entre o lançamento do Siri na Apple Store e o telefonema de Steve Jobs, informando que queria se encontrar com os criadores de Siri no dia seguinte.

Em outubro de 2011, ela foi lançada como parte integral do sistema operacional iOS do iPhone 4S. Para ativar o sistema basta o usuário falar “Hey Siri!”. Em seguida era só fazer as perguntas.

Siri implementa algumas respostas criativas. Uma delas também faz referência ao computador Deep Thought, aquele do livro *O Guia do Mochileiro das Galáxias*, que disse que a resposta para a questão fundamental da vida, universo e tudo o mais era 42. Se você perguntar à Siri “Qual o significado da vida?”, a resposta poderá ser “42”. Porém, você nem sempre receberá a mesma resposta. Ela poderá dizer que é “chocolate”, ou ainda, a frase final do filme *O Sentido da Vida (The Meaning of Life, 1983)* do grupo Monty Python, que sugere que é: “*Uh, try and be nice to people, avoid eating fat, read a good book every now and then, get some walking in, and try and live together in peace and harmony with people of all creeds and nations.*”<sup>18</sup>

Siri faz referências à sua avó ELIZA. Se você perguntar em quem ela vai votar, ela diz que é apenas um programa e que por isso não vota, mas que se pudesse votaria em ELIZA. Se você insistir que lhe conte uma história, ela vai contar sobre como foi trabalhar na Apple e mencionará ELIZA algumas vezes.

Em sua versão original, ela era bem mais irreverente e tinha até a palavra “*Fuck*” em seu vocabulário. Entretanto, a Apple lhe passou algumas barras de sabão na boca.

Várias empresas criaram sistemas semelhantes a Siri nos últimos anos. Entre outras, a Google, criou uma interface de voz para o Google Now, que tem poder bastante similar a Siri. Além disso, a IBM está trabalhando para permitir que o Watson possa ser

acessado como um serviço em smartphones, e promete que será uma Siri turbinada, capaz de ajudar médicos, enfermeiros, fazendeiros, corretores da bolsa e estudantes de nível superior com conhecimento especializado. Isso, porém, já vai um pouco além dos objetivos de um assistente pessoal.

## 12.6 Google Glasses – 2012

Quem já viu em filmes de ficção científica humanos do futuro ou alienígenas usando óculos de realidade aumentada que permitem conectar sua visão com dados obtidos de computadores deve ter sentido que o futuro chegou quando, em 2012, a Google anunciou o Projeto Glass.

Um dispositivo pensado para ser um simples par de óculos transparente, mas que pode projetar na lente imagens geradas por computador, como por exemplo, eventos da sua agenda, comunicação visual com outras pessoas, dados sobre o clima, dados de navegação, informações sobre estabelecimentos comerciais etc. Além disso, ele permite que você grave ou envie vídeos do que está vendo. Tudo isso controlado por voz. Uma versão do dispositivo para ser usada com lentes de grau é mostrada na [Figura 12.3](#).



Figura 12.3: Um modelo de Google Glasses.<sup>19</sup>

Na conferência da Google, a “Google I/O”, Sergey Brin interrompeu a palestra sobre o Google+ para anunciar um evento: ele ia demonstrar os Glasses. A plateia foi à loucura. Brin então chamou dois paraquedistas que estavam em um dirigível sobre o local da conferência e conectou a tela da apresentação aos Glasses deles. Assim, a plateia passou a ver as imagens que eles viam à medida que se jogavam de paraquedas para cair exatamente sobre o prédio da conferência, cujas paredes eles desceram por rapel e depois entraram de bicicleta no auditório para delírio de todos. Parecia que um novo mundo de possibilidades estava se abrindo.

Então as coisas ficaram um pouco estranhas, para dizer o mínimo, quando Robert Scoble (Estados Unidos, 1965), um dos entusiastas do Glass, postou no Google+ uma foto dele mesmo no chuveiro usando os Glasses e comentando que eles não estragaram, resistiriam até à chuva e nunca deveriam ser tirados. Larry Page, na época CEO da Google, comentou com Scoble que não gostou nem um pouco da foto.

A Google lançou então um programa piloto, no qual ela esperava que apenas desenvolvedores de software com interesse em desenvolver aplicativos para o Glass pudessem adquirir um aparelho por 1.500 dólares. Um preço propositalmente alto para afastar o comprador comum. Porém, o tiro saiu pela culatra: o alto preço atraiu mais ainda a atenção das pessoas, que queriam ter um desses. Os primeiros Glasses foram entreguem em 2014. Mais ou menos nessa época, surgiu o termo pouco apreciativo “glasshole”,<sup>20</sup> para se referir a pessoas que usam os Glasses em situações socialmente inaceitáveis. O termo chegou a entrar para o Dicionário Urbano no mesmo ano.

Ciente das preocupações dos usuários, a Google lançou um manual dos “Do’s and Don’ts”,<sup>21</sup> na esperança de colocar limites no uso do equipamento e reduzir um pouco a sensação das pessoas de que estariam o tempo todo sendo vigiadas por óculos que transmitiram imagens pela internet. A Google também prometeu que não

instalaria aplicativos de reconhecimento facial nos Glasses, mas o efeito dessas iniciativas foi o inverso do que a Google esperava. Em 2014, restaurantes e cinemas começaram a banir o uso de Glasses nas suas instalações.

Pior ainda foi o fato de que o alto preço da tecnologia acabou criando uma imagem para usuários de Glasses de que eles seriam pessoas abastadas obcecadas por tecnologia e, logo, todos os seus usuários passaram a ser chamados de glassholes. Poucos gostaram disso e muitos pararam de usar o equipamento. No final de 2014, o Twitter anunciou que não iria mais manter seu aplicativo para o Glass.

Em 2015, a Google suspendeu o projeto para repensá-lo. No final de dezembro de 2015, a empresa apresentou um novo projeto à comissão federal de comunicações. É possível que uma nova versão possa aparecer em 2016, mas nada confirmado ainda.

## 12.7 Michigan Micro Mote – 2014

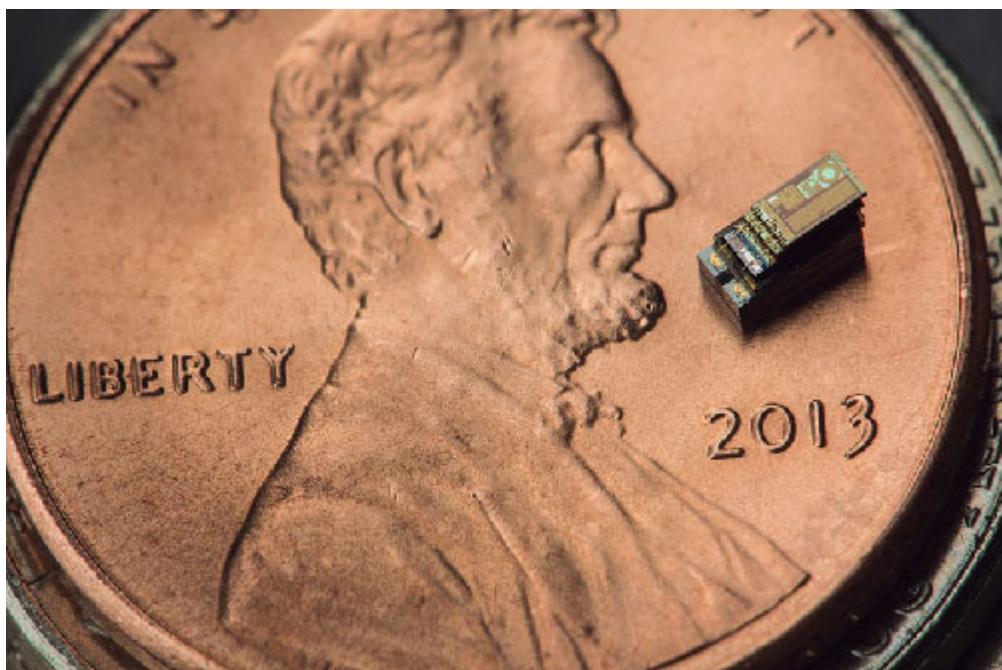
Em 2014, a Universidade de Michigan anunciou a construção do menor computador de todos os tempos: o Michigan Micro Mote ou M<sup>3</sup>. O computador completo é tão pequeno que 150 deles cabem dentro de um dedal. O projeto foi realizado pelos professores David Blaauw, Dennis Sylvester, David Wentzloff e Prabal Dutta, juntamente com vários estudantes.

Trata-se de um computador completo, inclusive com bateria e um sistema de recarregamento baseado em célula fotoelétrica. Ele usa tão pouca energia que pode ser carregado até com a luz interna de uma residência. Sua célula fotoelétrica tem apenas um milímetro quadrado e produz até 20 nano Watts (20 bilionésimos de Watt). O consumo do computador em modo de espera é de 2 nano Watts, ou um milionésimo daquele de um smartphone em estado de espera.

Ele foi projetado para ser usado como sensor. Um pesquisador pode, por exemplo, jogar um monte deles de um avião sobre uma floresta e passar a receber dados de cada um sobre temperatura,

umidade, pressão do ar *etc*. Ele pode também ser usado na Internet das Coisas, para funcionar dentro dos mais diversos dispositivos ou locais.

O M<sup>3</sup> mede menos de 2 milímetros ([Figura 12.4](#)) e é considerado um computador completo porque tem entrada de dados, habilidade de processá-los e de tomar decisões, e possibilidade de produzir saída de dados. Seus sensores são usados para receber as entradas e um transmissor de rádio para emitir a saída.



**Figura 12.4:** Michigan Micro Mote colocado sobre uma moeda de um centavo de dólar.<sup>22</sup>

Esse tipo de computador tão diminuto é apelidado de “smart dust”, ou poeira inteligente, e pode revolucionar totalmente a maneira como a computação será vista no futuro. Será possível, por exemplo, inseri-lo dentro de um corpo humano em um procedimento de ambulatório e, com ele, acompanhar as informações clínicas de um paciente, como temperatura, pressão e batimentos cardíacos,

emitindo alarmes ou até injetando medicamentos quando necessário.

## 12.8 Apple Watch – 2015

O Apple Watch foi um dos projetos mais ambiciosos da empresa. Numa época em que já se está falando em “computação vestível”, nada melhor do que um relógio computador para abrir o mercado. Vários competidores da Apple tentaram produzir e chegaram a lançar esse tipo de tecnologia, mas ninguém usava. Assim, eles se comprometeram em produzir o melhor produto de todos os tempos.

O projeto iniciou por volta de 2002, quando engenheiros da Apple solicitaram que a Nike lhes enviasse quantidades de relógios esportivos de alta tecnologia. Eles estudaram estes equipamentos e se familiarizaram com eles. A partir de 2011, a Apple começou a solicitar patentes de dispositivos vestíveis. A primeira foi de um equipamento no formato de um punho de camisa, capaz de carregar sua bateria a partir da energia cinética produzida pelo braço humano. O dispositivo poderia incluir uma tela sensível ao toque flexível e vários sensores para monitorar sinais vitais enquanto uma pessoa se exercita, por exemplo.

Mais ou menos na mesma época, a empresa submete a patente do iTime, uma espécie de relógio inteligente com a capacidade de tocar músicas e vídeos.

O Apple Watch foi concebido no laboratório de Jony Ive (Reino Unido, 1967), na Apple, logo após a morte de Steve Jobs em 2011. A notícia de que a Apple estaria desenvolvendo um dispositivo com uma tela de 1,5 polegadas e bluetooth de baixa energia, com comando de voz e vestível como um relógio, vazou em sites de notícias tecnológicas chineses em 2012.

Porém, nessa época, Ive e sua equipe ainda estavam estudando as modificações que a interface do relógio deveria ter em relação ao iPhone para que pudesse ser operada sem dificuldade pelos usuários.

Em 2013 correram boatos de que o Apple Watch já havia passado da fase de teste e estava em franca produção. As fontes não identificadas do vazamento afirmavam que o dispositivo teria a capacidade de fazer e receber chamadas de vídeo e navegar por mapas, além de possuir um pedômetro para contar o número de passos que a pessoa estava dando e sensores para outras informações relacionadas à saúde do usuário. Nesse ano, o fornecedor de vidro flexível da Apple, a empresa Corning (fabricante do Willow Glass), afirmou que seu produto ainda não estava pronto para uso como relógio de pulso. Porém, ainda em 2013 a Apple solicitou 79 patentes com a palavra “pulso” nelas e começou a registrar a marca “iWatch” em vários países.

Porém, a conferência mundial de desenvolvedores da Apple em 2014 aconteceu sem nenhuma menção ao projeto. Nesse meio-tempo, a Apple continuava contratando especialistas da área de relojoaria, medicina, esportes e moda.

Finalmente, em setembro de 2014 a espera chegou ao fim quando, ao final do lançamento do iPhone 6 e 6 plus, o apresentador Tim Cook, chefe executivo da Apple diz “*one more thing...*”<sup>23</sup> Ele então apresenta o Apple Watch, como foi batizado, e não iWatch, como todos esperavam visto que a marca iWatch, a Apple descobriu, já havia sido registrada nos Estados Unidos e União Europeia.

Ele foi lançado em três versões: Watch, Sport e Edition. Atualmente uma quarta versão chamada Hermès também está disponível. Cada modelo está disponível em dois tamanhos de 3,8 e 4,2 cm. O relógio é controlado por um clássico botão giratório lateral, como nos relógios antigos, porque os designers consideraram que a tecnologia de interface no estilo iPhone não iria funcionar bem com um dispositivo tão pequeno. Com o botão, o usuário pode dar zoom e selecionar aplicativos, além de acionar a Siri em caso de necessidade.

O Apple Watch passou a ser distribuído em abril de 2015 e funciona conectado a um iPhone por rede sem fio. A [Figura 12.5](#) mostra um

Apple Watch iniciando uma mensagem que é confirmada no iPhone. O objetivo, segundo os criadores do aparelho, é dar liberdade aos usuários do iPhone que poderão então visualizar mensagens ou imagens em seus pulsos sem ter que pegá-los no bolso ou seja lá onde estiverem.

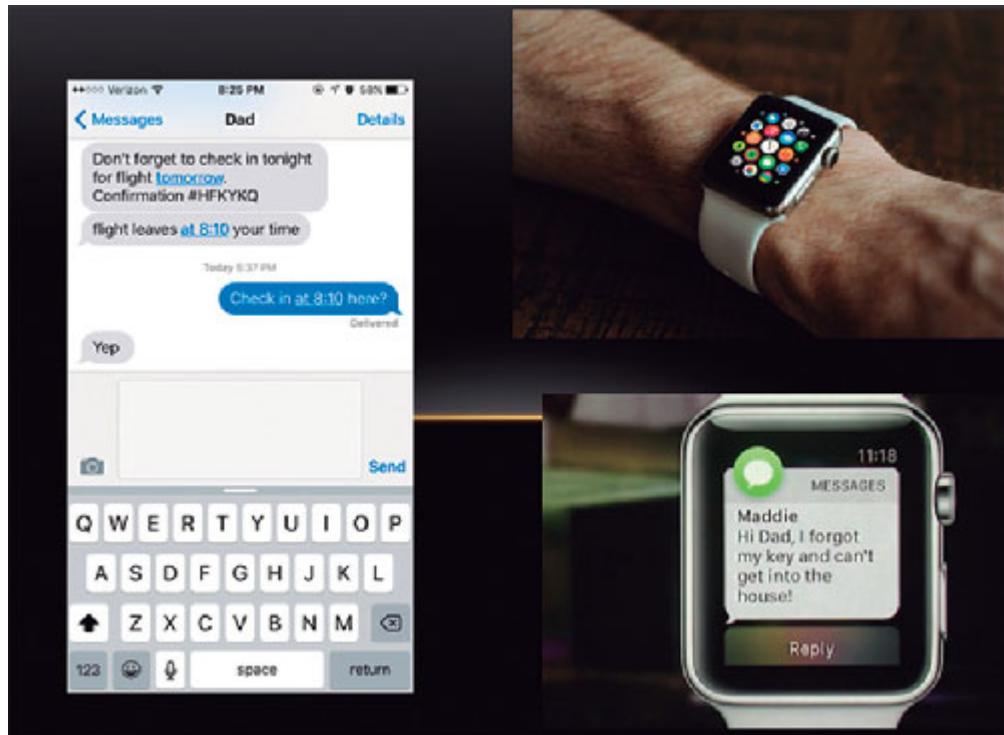


Figura 12.5: Um Apple Watch ao lado de um iPhone.<sup>24</sup>

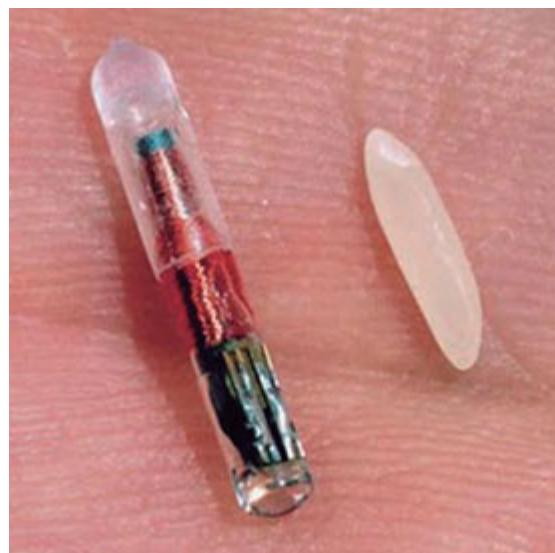
Um analista comentou que gostaria muito mais de um Apple Watch do que um Google Glass pois as pessoas usam óculos normalmente porque precisam, mas ninguém usa relógio porque precisa, e sim porque gosta. A Apple anunciou também que no futuro o Watch poderá substituir as chaves do carro.

## 12.9 Previsões até 2020

A história continua. Até aqui, fomos capazes de relatar uma pequena parte de tudo o que aconteceu e nos trouxe ao ponto em que estamos. Hoje, em 2016, mal podemos vislumbrar os

desenvolvimentos futuros desta história. Assim, apenas como curiosidade, apresentaremos nesta seção e nas próximas as previsões relacionadas à computação do site <http://www.futuretimeline.net/>, uma especulação, sim, sobre nossa história futura, mas baseada em pesquisa detalhada, tendências atuais, mudanças ambientais de longo prazo e leis da tecnologia, como a Lei de Moore, que prevê que a quantidade de transistores que se consegue integrar em um único chip deve dobrar a cada 18 meses e que vem se concretizando ao longo das últimas décadas. Esperamos, porém, que o leitor não leve esta sessão muito a sério e dentro de 5 ou 10 anos venha nos cobrar essas previsões. A chance de errar aqui é de quase 100%. Mesmo assim, trata-se de um exercício interessante de imaginação, mas uma imaginação com base naquilo que se observa hoje e no que é possível, considerando a tecnologia atual e não apenas a fantasia.

Para 2016, o site prevê que todos os cães da Inglaterra terão microchips. Isso será exigido por lei a partir de abril. Dessa forma, será muito mais fácil encontrar cães perdidos. A [Figura 12.6](#) mostra um desses microchips comparado a um grão de arroz.



**Figura 12.6: Microchip e grão de arroz.<sup>25</sup>**

Também em 2016 a China terá os dois supercomputadores mais velozes do mundo, o Tianhe-2a, uma expansão do Tianhe-2 ([Figura 12.7](#)) que já era o mais rápido desde 2013 e que terá sua capacidade dobrada para chegar a 100 petaflops, ou 100 quadrilhões de operações de ponto flutuante por segundo, e o Shenwei-x, já criado com essa capacidade. Porém, a liderança deverá ser curta porque os Estados Unidos já estão a ponto de lançar computadores de 150, 180 e 300 petaflops. Um deles será construído pela IBM. Até o final da década, espera-se construir computadores com capacidade de 1.000 petaflops.

O ano de 2016 também prevê substancial aumento na automatização de operações de mineração e agricultura. Câmeras de vídeo de alta resolução estarão por toda a parte vigiando as ruas e instalações, pessoas se lembrarão do livro *1984* e de seu “Grande Irmão”, mas se resignarão em serem vigiadas por questões de segurança. Bem, talvez nem todas. Esse ano também deve ver o aumento do uso dos monitores OLED, ou LED orgânico, que consome muito menos energia do que os LCD usuais.

Em 2017, a China deverá lançar a primeira sonda não tripulada à Lua, que vai ir e voltar com amostras de solo. Toda a operação robótica da sonda será automatizada.



**Figura 12.7: Tianhe-2, atualmente o computador mais rápido do mundo.<sup>26</sup>**

Nesse ano, deve entrar em produção o chip de 10 nanômetros (atualmente o limite é 14 nm). Dessa forma, mais de 10 bilhões de transistores poderão ser empacotados em um único chip. Isso deve aumentar a performance e reduzir o consumo de energia em computadores, smartphones e quaisquer outros dispositivos baseados em chips. Porém, a Lei de Moore deve chegar ao final por volta de 2020, quando a redução dos componentes de um chip alcançar os limites onde a mecânica quântica passe a ser um problema. Talvez por essa época, novos materiais como o grafeno ajudem a construir chips mais poderosos do que os de silício.

Ainda em 2017, espera-se que a quantidade de dispositivos com capacidade de gravar ou transmitir vídeo conectados em rede excederá a população humana, ultrapassando a marca dos 8 bilhões.

Por essa época também se espera que as pesquisas que já duram 10 anos com papel eletrônico produzam dispositivos usáveis, introduzindo, por exemplo, smartphones ultrafinos, realmente da espessura de uma folha de papel. Tablets talvez possam ser dobrados e guardados no bolso.

Espera-se ainda para 2017 o início da implantação de chips sob a pele de pessoas para monitorar condições de saúde através da interligação com smartphones.

Para 2018, acredita-se que o avanço na robótica permitirá que muitas cirurgias de alta complexidade passem a ser realizadas por robôs, mas sempre com controle ou supervisão de um médico. Com recursos de telemedicina, por exemplo, um médico poderá operar pacientes em várias cidades diferentes no mesmo dia.

Neste ano ainda os projetos militares de insetos robotizados para operações especiais deverão estar operacionais.

Além disso, em 2018, os HDs baseados em discos giratórios já deverão estar praticamente obsoletos, e os dispositivos de armazenamento sólido terão capacidade de 128 TB para empresas e 16TB para usuários domésticos. Em termos de velocidade de transferência, dispositivos com 100 GB/s poderão transferir todo o conteúdo de um filme em Blu-ray em 3 segundos.

Em 2018, com o avanço da tecnologia, scanners 3D poderão ser capazes de efetuar leituras precisas a mais de 1 km do objeto.

A partir de 2019 espera-se que a impressão 3D se torne tão trivial quanto a 2D é hoje. Nesse ano, os computadores devem atingir um petaflop, ou um quintilhão de operações por segundo. Com a evolução de projetos como o Watson, provavelmente estaremos nos comunicando com os computadores em linguagem natural quase da mesma forma como hoje usamos nossos browsers de internet.

Hoje já temos aparelhos auditivos que funcionam praticamente como ouvidos biônicos, restaurando a audição de pessoas completamente surdas. Em 2019, isso deverá ser possível também

para os olhos, com a disponibilização dos primeiros olhos biônicos comerciais.

Também em 2019 talvez ainda não haja carros robôs nas ruas, embora a tecnologia já exista mas os automóveis poderão estar interconectados para evitar acidentes. Por exemplo quando um carro freia bruscamente, ele avisa todos os carros nos arredores e, assim, mesmo um carro que venha três ou quatro veículos atrás poderá reduzir a velocidade antes que seu dono seja capaz de visualizar o automóvel que freou. Quanto aos carros robôs, espera-se que em 2019 a tecnologia de caminhões robôs, que já funciona em alguns aeroportos, seja expandida para as estradas. Afinal, dessa forma os caminhões podem trabalhar 24 horas por dia e motoristas robôs não dormem e dificilmente causam acidentes. A maior restrição a esse tipo de transporte hoje ocorre pelo medo da robótica, um fenômeno que o autor de ficção científica Isaac Asimov já previa em meados do século XX. Você realmente se sente tranquilo pensando em caminhões sem motorista humano correndo pelas estradas?

Para 2020, prevê-se que o número de usuários da Internet atinja 5 bilhões de pessoas. Além disso, a tecnologia de leitura de ondas cerebrais poderá estar tão avançada que as pessoas serão capazes de escrever textos ou comandar seus computadores apenas com o uso do pensamento.

Em termos de televisores, neste ano possivelmente serão comuns as telas de Ultra HD, com resolução de mais de 4 mil linhas (duas vezes mais que 4K e quatro vezes mais que HD). Além disso, a evolução dos displays holográficos poderá estar levando esses dispositivos para as residências nessa época.

Em 2020 espera-se que a tecnologia de drones que já temos hoje amadureça, especialmente nos aspectos legais, para que milhares dessas máquinas estejam dia e noite patrulhando nossos céus e realizando serviços de entregas.

## 12.10 Previsões até 2030

Previsões tecnológicas para mais de cinco anos são difíceis, mas se não houver nenhuma grande ruptura, ainda podemos ter os pés no presente e considerando pesquisas em andamento imaginar um futuro possível (sem compromisso).

Em 2021, poderemos ter scanners mentais sendo usados em locais como aeroportos ou estádios assim como hoje usamos aparelhos de raios X. Essas máquinas, espera-se, poderão detectar pessoas mal-intencionadas. Atentados terroristas recentes certamente estão fazendo com que a indústria de segurança explore todas as possibilidades, e essa é uma delas. Na verdade, a máquina não vai precisar ler as ondas cerebrais de uma pessoa para saber se ela está mal-intencionada. Para isso, basta analisar a expressão do rosto, linguagem corporal, temperatura e batimentos cardíacos, enfim, vários indicadores que ainda precisarão ser estudados e que podem produzir essa informação (ou não – afinal detectores de mentira já foram descartados como provas em julgamentos há muito tempo).

Também nesse ano espera-se que passe a ser comum aparelhos carregarem suas baterias por eletricidade transmitida sem fios, como Nikola Tesla sonhava.

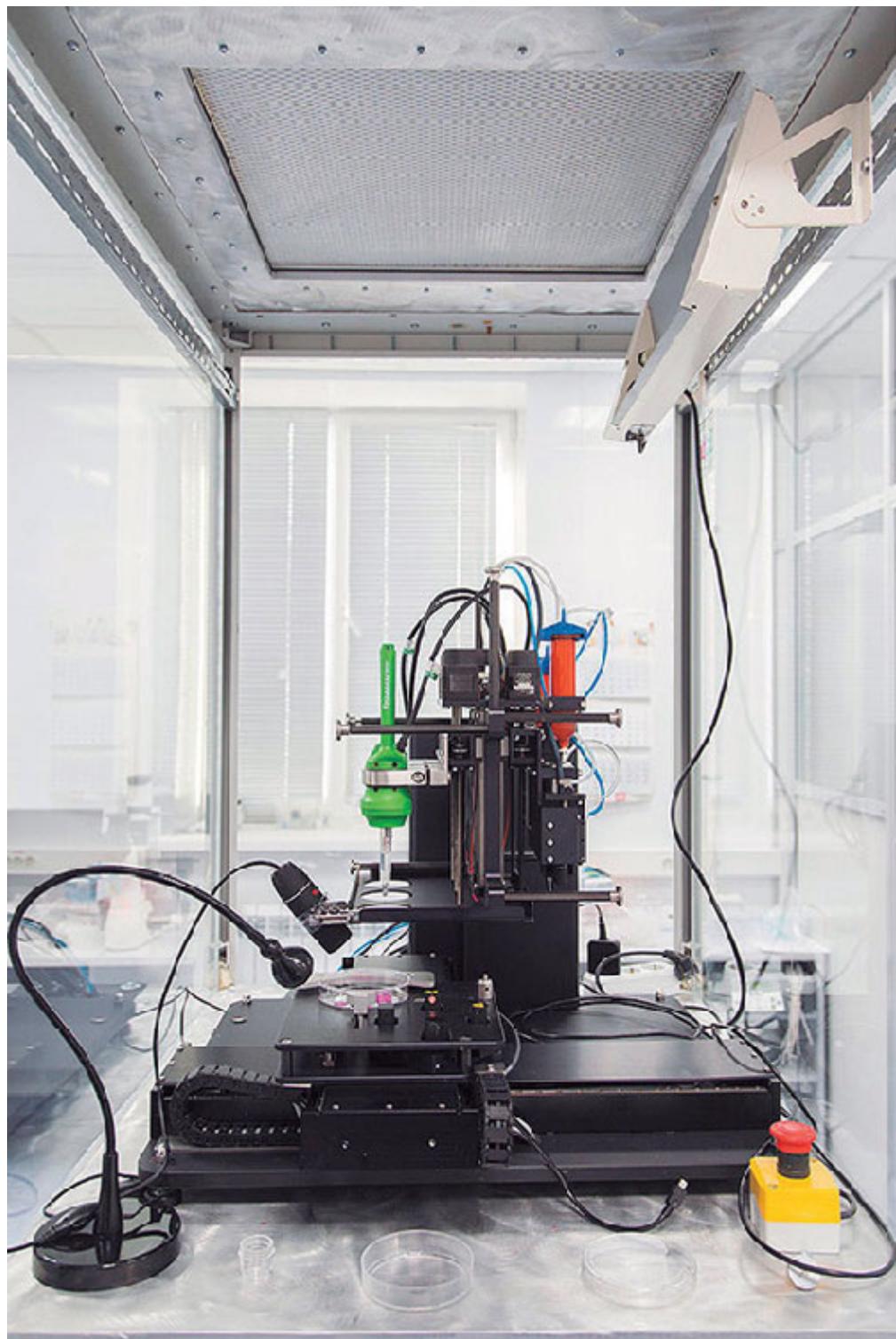
Por volta de 2023 espera-se que as pesquisas atuais com implantes neurais eletrônicos permitam usar microchips para que pessoas com doenças como Alzheimer possam se manter funcionais com o apoio de memórias eletrônicas implantadas diretamente em seus cérebros e interfaceando com eles. Atualmente, a pesquisa é feita com ratos e símios.

Em 2024 espera-se que a tecnologia de impressão 3D já seja tão sofisticada e barata que roupas impressas em 3D custarão praticamente nada. Estaremos bem próximos da tecnologia dos replicadores da série *Jornada nas Estrelas*.

Por outro lado, nessa época, a energia eólica deverá ter se expandido muito e o processo de inspeção das turbinas, que hoje é

manual, deverá tornar-se em uma indústria bilionária para fabricantes de drones e software de inspeção.

Em 2025, o aumento exponencial da capacidade dos computadores, que chegará a  $10^{19}$  flops, e do conhecimento neurológico deverá permitir a simulação de cérebros humanos. Ao mesmo tempo, robôs superminiaturizados, conhecidos como nanobots estarão sendo usados em procedimentos médicos. Espera-se também por volta deste período que órgãos humanos funcionais possam ser impressos a baixo custo pela tecnologia 3D, acabando com as filas de transplante em alguns casos. Hoje já é possível substituir alguns ossos com material impresso em 3D.



**Figura 12.8: Bioimpressora 3D.<sup>27</sup>**

Para 2027, espera-se que mãos robotizadas possam ter a mesma capacidade da mão humana.

Em 2028 o papel eletrônico será tão barato que estará por toda a parte. Haverá monitores de vídeo em embalagens de produtos, e colados nas paredes das casas e locais públicos como se fossem papel de parede.

Para 2029, espera-se que já tenha sido criada a primeira inteligência artificial semelhante à humana, ou seja, uma máquina capaz de passar no Teste de Turing. Neste mesmo ano, a maior parte dos supermercados e lojas nos países mais desenvolvidos não terá mais operadores de caixa, tornando-se completamente automatizados. Como em um comercial veiculado pela IBM na década passada, o comprador simplesmente entrará na loja e encherá seus bolsos e sacolas com os produtos que deseja. Ao sair, ele será automaticamente cobrado em seu cartão eletrônico, o qual é lido juntamente com as informações sobre todos os produtos através de radiofrequência.

Também em 2029, a tecnologia de propaganda inteligente deverá ter atingido os níveis mostrados no filme *Minority Report – A Nova Lei* (*Minority Report*, 2002). Esse tipo de inteligência de marketing hoje é restrito à Internet e seus aplicativos, mas no futuro poderá estar por toda a parte.

Em 2030 a inteligência artificial poderá estar tão disseminada quanto a Internet hoje. Vários empregos em áreas administrativas, como call centers, poderão ter sido substituídos por computadores. Papéis comuns, não eletrônicos, serão raros. Por volta desse período também se imagina que metade dos shoppings centers na América do Norte terão fechado as portas devido ao fato de as vendas terem migrado predominantemente para a Internet.

Novas profissões que poderão estar na moda em 2030: construtor de partes de corpos, cirurgião de aumento de memória, nanomédico, advogado virtual e professor virtual.

## 12.11 Previsões até 2040

A partir de 2031 a inteligência artificial terá dominado a Internet e feito surgir a Web 4.0, uma internet capaz de realizar tarefas que hoje nem sonhamos, rodando em supercomputadores com tanta capacidade de processamento que serão capazes de realizar atividades que exigem as mais sutis capacidades humanas. A Internet estará coletando informação de toda a parte: automóveis, casas, espaço, drones, nanobots etc. Apesar das questões de privacidade, que estarão em debate permanente, as novas tecnologias ainda não impedirão crimes de acontecer, mas farão com que a sua solução e punição dos responsáveis aconteça de forma muito mais rápida, pois uma enormidade de dados estará disponível sobre quase tudo o que acontece no planeta.<sup>28</sup> Por volta desse ano, praticamente todos os documentos, livros e obras de arte produzidos pela humanidade ao longo da história estarão digitalizados e disponíveis.

Em 2032 a velocidade da Internet, que hoje está em torno de 100 megabits, deverá atingir um terabit, ou seja, será 10 mil vezes mais rápida; isso se continuar na mesma velocidade de crescimento observada desde 1979 até hoje.

A partir de 2035, espera-se que enxames de robôs, como os vistos na animação *Operação Big Hero* (*Big Hero 6*, 2014), sejam comuns. Com essa tecnologia pode-se combinar milhões de minúsculos robôs muito simples que, isolados, não são capazes de fazer muita coisa, mas que em conjunto conseguem combinar suas habilidades como células de um corpo. Esses dispositivos poderão ter, por exemplo, a forma de insetos e inclusive voar; eles poderão ser programados para polinizar campos, na falta de outros insetos, ou realizar missões de resgate, digamos, procurando pessoas perdidas em florestas ou no mar. Os enxames poderão até o final da década de 2030 ser compostos por robôs tão miniaturizados que parecerão nuvens de poeira ao olho nu – mas serão matéria programável e inteligente.

Ainda em 2035 deverá ser possível realizar recriações holográficas de pessoas, inclusive falecidas, e a IA poderá lhes dar um comportamento semelhante aos originais.

Por outro lado, a maioria das guerras por essa época estará sendo lutada por robôs. Hoje já temos algo parecido em relação aos ataques aéreos, mas no futuro poderemos ter isso acontecendo no combate corpo a corpo.

Em 2036, espera-se que os olhos biônicos já sejam mais eficientes do que o olho humano em termos de resolução, dando capacidade de visão noturna, gravação e transmissão de imagens pela Internet. O preço dos implantes iniciará muito alto, mas deverá cair até o final da década.

Para 2037, espera-se que a maioria das universidades e órgãos públicos tenham pelo menos um computador quântico capaz de processamentos trilhões de vezes mais rápido que os de hoje, realizando criptografias praticamente inquebráveis.

Em 2038, computadores muito antigos, ainda dependentes do Unix correrão risco de passar a funcional mal devido ao erro do ano 2038, já que sua capacidade de representar datas em 32 bits terá se esgotado no início deste ano.

Em 2039, espera-se que, com os avanços da neurologia, nanorobótica e enxames robóticos, seja possível obter realidade virtual totalmente imersiva pela implantação de nanobots diretamente no cérebro humano. Pelo menos em nível experimental, prevê-se atingir um nível de realidade virtual compatível com o apresentado no filme *Matrix* (1999). Os nanobots serão auto-organizados e se prenderão diretamente aos neurônios do usuário, substituindo os estímulos reais do corpo pelos produzidos artificialmente. Quando o usuário quiser mover seu corpo, os nanobots interceptarão o sinal, impedindo o movimento real, mas produzindo no usuário a sensação de que realmente se moveu. Embora muitos ainda possam ter medo dos nanobots, cerca de 20 anos de uso com sucesso em procedimentos médicos terá criado

uma base científica para que esse tipo de experiência, em maior ou menor escala, realmente seja possível. Os nanobots poderiam ser ativados e desativados à vontade do usuário e, com o passar do tempo, as pessoas poderiam se acostumar a tê-los dentro de seus corpos, como as bactérias que digerem os alimentos. Bem, pelo menos seria uma forma econômica de tirar férias: deite-se na cama e viaje pelo mundo, ou pela galáxia.

À medida que a comunicação cérebro/computador evoluir, o ano de 2040 poderá ter como forma dominante de comunicação a transmissão telepática de pensamentos. Porém, não envolverá nada de paranormalidade: apenas nano sensores dentro de seu cérebro, lendo as informações diretamente de seus neurônios e transmitindo-as pela Internet até o destinatário, cujos próprios nanobots decodificarão as palavras ou imagens que você transmitiu. Aposto que você teve um frio na espinha nessa hora, não? Entretanto, considerando a evolução da IA, da neurologia e da nanorobótica, talvez seja possível um dia. Em vez de teclar em nossos smartphones, talvez tenhamos apenas que pensar nas pessoas. O problema vai ser controlar as caixas de correio.

## 12.12 Previsões até 2050

Se a velocidade de crescimento da capacidade de processamento dos computadores prosseguir, eles devem atingir um yottaflop por volta de 2040. Isso é um trilhão de trilhões de operações de ponto flutuante por segundo, 1.000 vezes mais rápido que o zettaflop de 2030 e 1 milhão de vezes mais veloz que o exaflop de 2019. A freada que o crescimento da velocidade dos computadores deu na década de 2010 pode ser apenas um evento temporário que será compensado por novas ideias e tecnologias de processamento do futuro, por exemplo, substituindo silício por nano tubos de carbono. Nessa época, um supercomputador poderá simular milhares de cérebros humanos em nível de neurônio, todos ao mesmo tempo.

Você consegue imaginar a vida hoje sem um smartphone, sem a internet? Pois bem, é possível que os humanos de 2045 não consigam imaginar a vida sem estarem integrados à tecnologia, seja através de próteses, olhos biônicos, expansões de memória, ou nanobots capazes de curar doenças e levar você em viagens astrais. Nesse ano, o computador pessoal terá a capacidade de 1 milhão de cérebros humanos e possivelmente vida própria, trocando milhões de mensagens por dia através da internet. A única maneira de um humano conseguir interagir com tanta informação será através de uma imersão na rede, literalmente, deixando sua consciência entrar em contato diretamente com a rede. Os nanobots, que antes eram apenas para aplicações médicas ou de realidade virtual, agora são um instrumento de comunicação diário. Não há mais necessidade de monitores ou teclados, nem mesmo de microfones ou alto-falantes. Mesmo pessoas nascidas cegas e surdas, como Helen Keller (Estados Unidos, 1880-1968) poderiam se comunicar com o resto do mundo. Implantes nanorrobóticos passarão a ser permanentes e essenciais ao invés de temporários e opcionais.

O sonho de ter robôs empregados em casa, que víamos no desenho animado *Os Jetsons* dos anos 1960, precisará ser postergado até por volta de 2049 quando a IA e a mecânica fina permitirão que máquinas possam efetivamente realizar nossas tarefas domésticas com eficiência. Hoje temos robôs que aspiram o pó do chão e máquinas de lavar louça e roupa, mas eles são meros brinquedos se comparados ao que poderá vir nos próximos anos. Os androides, como são chamados os robôs com aspecto humano, serão especialmente populares entre idosos e deficientes. É possível que toda uma indústria de robôs para atividades sexuais também esteja em plena atividade nessa época. O filme *I.A.: Inteligência Artificial* (*A.I. – Artificial Intelligence*, 2001) já mencionava essa possibilidade. Além de androides, poderemos ter toda sorte de animais domésticos na forma de robôs: eles se alimentarão da eletricidade distribuída pelo ar e não farão sujeira.

Aliás, isso deve vir primeiro, pois acredita-se ser bem mais fácil simular a inteligência de um animal como um cão, gato ou peixe, do que a inteligência de um ser humano.

Nos hospitais, possivelmente a maior parte do trabalho estará sendo feita por robôs e nanobots. E, no espaço, eles já terão explorado a maior parte das luas e planetas do sistema solar.

Em 2050, a grande maioria dos carros fabricados no mundo deverá ser dirigida por computador, enquanto as redes de tráfego serão gerenciadas por grandes sistemas de inteligência artificial. Em função do controle automatizado, as velocidades nas estradas serão maiores do que hoje, os congestionamentos quase não existirão e os acidentes serão extremamente raros.

## 12.13 Previsões até 2100

À medida que passamos da metade do século XXI, as previsões vão se tornando cada vez mais imprecisas, pois muitos eventos podem interromper o progresso ou acelerá-lo em direções diferentes daquelas que imaginamos hoje. Mas vamos lá...

Por volta de 2055 não haverá mais a grande mídia como a conhecemos hoje: entretenimento e informações estarão fragmentados e diversificados.

Em 2058, o computador mais rápido será capaz de simular 1 bilhão de cérebros humanos simultaneamente.

Por volta de 2060, a educação como hoje conhecemos com salas de aula e professor terá mudado muito. Com a nanotecnologia e a inteligência artificial, o ensino será individual e acessível diretamente pela rede. A informação poderá efetivamente ser “baixada” no cérebro do estudante como hoje ele baixa textos na internet. Esse sistema, possivelmente, será tão barato que estará disponível a todos, inclusive nos países menos desenvolvidos. Além disso, os tradutores universais terão removido todas as barreiras de linguagem. Hoje já podemos traduzir automaticamente, embora de forma ainda imperfeita, nossas páginas Web a partir de centenas de

línguas. No futuro, isso deverá ser ainda mais sofisticado. Mesmo a comunicação telepática, a que nos referimos antes, poderá ser feita com pessoas que falam qualquer língua conhecida. A realidade virtual totalmente imersiva fará com que estudantes possam aprender sobre história, digamos, indo diretamente aos locais e épocas dos fatos. O mesmo poderá ser realizado com as outras ciências: fenômenos, físicos, químicos ou biológicos poderão ser observados em macro ou microescala, em alta resolução e com didática especificamente personalizada para cada estudante individualmente.

Nessa época, como a maioria das profissões repetitivas e de esforço físico estará sendo realizada pelas máquinas, os seres humanos poderão se dedicar às profissões que exigem criatividade como ciência, arte e design.

Por volta de 2062, nano fábricas, uma versão mais sofisticada de impressoras 3D estarão disponíveis. Elas poderão criar praticamente qualquer coisa, até comida, com o uso de nanobots e matéria-prima que pode ser obtida de diversas formas. A nanotecnologia trará, inclusive, a possibilidade de prédios que se autoconstroem átomo por átomo a partir de 2065. Em 2070, arranha-céus poderão brotar do chão aparentemente sozinhos, mas na verdade erguidos por um exército de trilhões de nanobots comandados por uma inteligência artificial.

Em 2070, as casas poderão estar mais inteligentes, gerando sua própria energia e água e tratando seus próprios resíduos. Além disso, elas estarão cobertas de superfícies inteligentes e interativas e, provavelmente, serão menores porque a população estará crescendo.

Em 2072, a humanidade poderá estar se aproximando da picotecnologia, ou seja, o controle de estruturas ainda menores do que os nanobots: mil vezes menores. Com ela, será possível trabalhar diretamente na estrutura dos átomos, criando formas de matéria exóticas com propriedades nunca antes vistas.

Em 6 de junho de 2079, os poucos computadores que ainda estiverem rodando bancos de dados baseados em SQL, então completando 100 anos de existência, sofrerão do problema “smalldatetime”, pois os registros de tempo nesses sistemas retornarão à 1º de janeiro de 1900, um problema similar ao do ano 2038.

Em 2083, computadores do tamanho de notebooks terão o poder de processamento dos cérebros de toda a humanidade ao longo dos últimos 10 mil anos em menos de 10 microssegundos. A tecnologia e o conhecimento avançam tão rápido que implantes cerebrais para expansão de memória e raciocínio se tornarão muito procurados. As pessoas são, cada vez mais, formadas por partes criadas artificialmente.

Em 2084 é possível que androides hiperinteligentes acabem substituindo as forças policiais. Eles serão muito mais fortes do que uma pessoa, mas terão que ser adequadamente controlados para evitar o uso excessivo da força. As leis de robótica de Isaac Asimov possivelmente serão recitadas regularmente: 1. Um robô não pode ferir um ser humano ou, por inação, permitir que um ser humano sofra algum mal. 2. Um robô deve obedecer às ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens entrem em conflito com a Primeira Lei. 3. Um robô deve proteger sua própria existência desde que tal proteção não entre em conflito com a Primeira ou Segunda Leis.

## **12.14 Previsões até 2200**

O século XXII... Eu não estarei aqui, por certo, mas talvez você, leitor, ainda viva por estes tempos. Afinal, no momento em que escrevo este livro, faltam apenas 85 anos e não sei quando você estará lendo. Previsões aqui são fantasiosas em grande parte, mas arriscaremos, ainda seguindo o site <http://www.futuretimeline.net/> e suas previsões feitas em 2016.

Em 2100, a inteligência humana terá sido amplamente aumentada pela IA. Sistemas de IA poderão estar ocupando cargos-chave em governos e empresas. Cada vez mais haverá uma mescla entre a mente humana e a IA. Pessoas poderão aprender novas habilidades, como por exemplo, andar de bicicleta, simplesmente baixando o respectivo programa da Internet em seu cérebro. Assim como os Amish, hoje, se recusam a compartilhar da tecnologia moderna, haverá comunidades de pessoas “puras”, que não aceitarão esse nível de envolvimento com a tecnologia. Isso poderá criar uma grande divisão na sociedade.

Em 2110, a femtotelecomunicação, operando com elementos 1.000 vezes menores que a picotelecomunicação e 1 milhão de vezes menores que a nanotelecomunicação, permitirá manipular quarks e cordas, bem como a evolução de estudos relacionados à teleportação e antigravidade.

Em 2120, quem sabe, com tanto progresso será possível fazer upload de uma mente humana em um computador, permitindo assim às pessoas viverem eternamente dentro das máquinas ou ainda construindo novos corpos artificiais através dos quais continuarão sua vida consciente. Isso, claro, depende fundamentalmente de que a consciência humana seja realmente resultado das interações elétricas e químicas dentro do cérebro. Acontece que há filósofos e mesmo físicos modernos que duvidam disso e acreditam que a consciência é algo que vai muito além dessas interações e, portanto, impossível de se simular em uma máquina. Certamente haverá grandes conflitos com grupos religiosos.

Por volta de 2150, a evolução da realidade virtual permitirá a criação de ambientes idênticos aos holodecks da série *Jornada nas Estrelas*. Planetas inteiros poderão ser criados em realidade virtual e sua fauna e flora poderão seguir padrões de evolução diferentes daqueles que aconteceram na Terra. Bilhões de anos de evolução poderão ser simulados em poucas horas.

As cidades serão muito mais limpas. Placas e sinais, por exemplo, serão supérfluos pois bastará que a pessoa pense em um prédio

para receber diretamente em seu córtex as informações sobre ele.

## 12.15 Previsões Finais

O século XXIII provavelmente verá o fim do emprego como hoje conhecemos porque as máquinas realmente farão todo o trabalho. O planeta provavelmente terá uma população menor e suas áreas selvagens estarão sendo repovoadas.

Por 2220, quem estiver por aqui poderá viajar para outros planetas, sem o perigo e desconforto da viagem espacial: simplesmente bastaria fazer o upload de sua mente em androides que já estarão lá. As pessoas também poderão colocar as próprias mentes em diversos corpos, como o de aves, peixes, criaturas mitológicas ou quaisquer outros que estiverem disponíveis. Algo parecido com o que acontece no filme *Avatar* (2009).

Um ser humano de 2300 terá tantas partes de seu corpo e mente otimizadas pela cibernetica que provavelmente seria considerado um super-homem nos dias de hoje.

Avançando um pouco mais até o ano 4000, é possível que a simbiose entre humanos e máquinas tenha se tornado tão perfeita que a ciência da computação se torne uma ciência obsoleta como campo de estudo, pois todas as possibilidades técnicas já terão sido aperfeiçoadas ao máximo. Nem hardware nem software poderão evoluir mais a partir desse ponto e todo o progresso seguinte estará nas mãos de outras ciências.

Imaginar 8 mil anos no futuro a partir de hoje já é um grande salto. Porém, podemos claramente imaginar a humanidade 8 mil anos no passado, em suas tentativas pioneiras de agricultura e suas primeiras guerras, e ainda nos reconhecer lá. Então talvez o ano 10.000 nem esteja tão longe. É possível que alguns dos super-humanos do século XXIII ainda estejam vivos no século C, em função das transferências de consciência. Eles seriam, portanto, os primeiros imortais relativos, ou seja, pessoas que não morreriam nunca de causas naturais.

Bem... Como dissemos, caro leitor, não leve estas previsões muito a sério. São apenas possibilidades e existem inúmeras outras. Em 1910, um artista francês desenhou várias imagens sobre como ele imaginava que seria o mundo no ano 2000. Numa delas ([Figura 12.9](#)), aparecem grandes dirigíveis cruzando o Oceano Atlântico. Ele acertou a intenção (viajar), mas errou a tecnologia (dirigíveis), pois em 1910 ninguém ainda imaginava que um dia existiriam turbinas a jato.

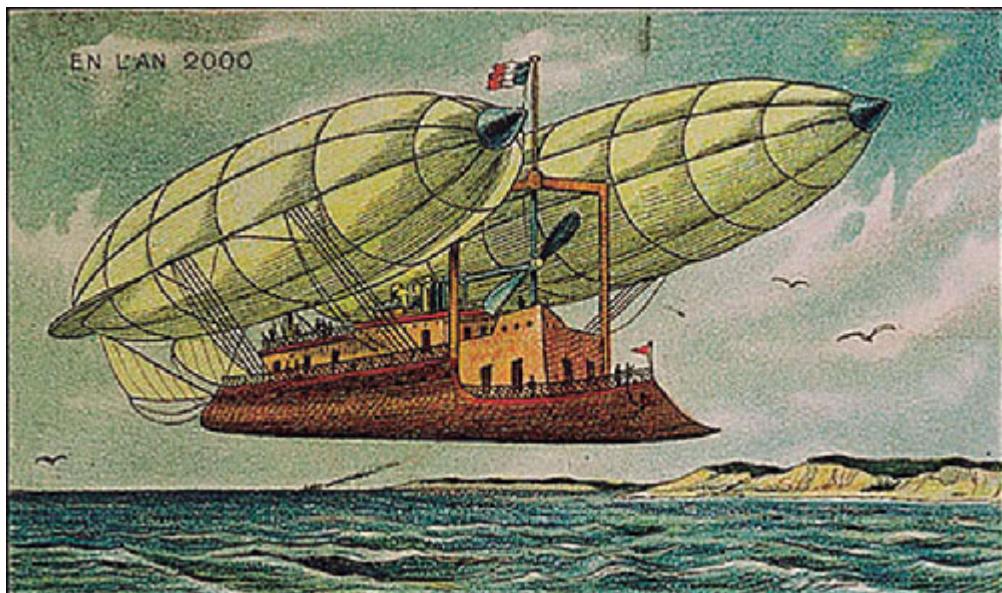


Figura 12.9: "No ano 2000" – previsão de 1910.<sup>29</sup>

O certo é que se nenhuma grande ruptura, como uma nova Idade Média causada por extremistas religiosos ou *anti-vaxxers* (movimento antivacinação) acontecer, a tecnologia deverá continuar evoluindo. Houve quem pensasse que ela não iria mais se expandir depois de 1980, mas aqui estamos em uma sociedade já bastante diferente daquela em termos de evolução tecnológica. Enfim, o futuro virá e nós veremos.

---

<sup>1</sup> A história original pode ser vista em:  
[http://www.google.com/googlebooks/chrome/small\\_00.html](http://www.google.com/googlebooks/chrome/small_00.html).

<sup>2</sup> By LG전자 –Disponível em: <http://www.flickr.com/photos/lge/11426847334/>This file comes from LG Electronics's official Flickr. CC BY 2.0,  
<https://commons.wikimedia.org/w/index.php?curid=30217668>

<sup>3</sup> Tradução: Compartilhamento rápido de fotos bonitas.

<sup>4</sup> Tradução: “O YouTube derruba a barreira entre a audiência e o criador. Eles sentem uma conexão com aquele a que estão assistindo.”

<sup>5</sup> Tradução: Manos.

<sup>6</sup> Tradução: Exército dos manos.

<sup>7</sup> Tradução: “Pensando nisso agora, foi totalmente absurdo.”

<sup>8</sup> By Clockready - Own work, CC BY-SA 3.0. Disponível em:  
<https://commons.wikimedia.org/w/index.php?curid=15891787>

<sup>9</sup> Tradução: “Eu me senti obsoleto. Eu me senti como um trabalhador de fábrica de Detroit nos anos 1980 vendo um robô fazer seu trabalho em uma linha de montagem. Eu senti como se ‘competidor de show de perguntas e respostas’ fosse agora o primeiro trabalho que tinha se tornado obsoleto sob o novo regime das máquinas que pensam.”

<sup>10</sup> Tradução: Pensador Profundo.

<sup>11</sup> Tradução: “Seu maior aeroporto foi nomeado em homenagem a um herói da Segunda Guerra Mundial; e o segundo maior em homenagem a uma batalha da Segunda Guerra Mundial.”

<sup>12</sup> Tradução: “Eu, por exemplo, dou boas-vindas a nossos novos dominadores computadores.”

<sup>13</sup> Tradução: “Eu, por exemplo, dou boas-vindas a nossos novos dominadores insetos.”

<sup>14</sup> Tradução: “Para mim foi uma experiência muito legal e eu a faria novamente em um piscar de olhos. Não se trata dos resultados, mas de fazer parte do futuro.”

<sup>15</sup> Tradução: Interface de Reconhecimento e Interpretação de Fala.

<sup>16</sup> Tradução: Assistente Cognitivo que Aprende e Organiza.

<sup>17</sup> Tradução: HAL está de volta – mas desta vez ele é bom.

<sup>18</sup> Tradução: “Hum, tente ser legal com as pessoas, evite comer gordura, leia um bom livro de vez em quando, procure caminhar e tente conviver em paz e harmonia com pessoas de todos os credos e nações.”

<sup>19</sup> By Mikepanhu - Own work, CC BY-SA 3.0. Disponível em:  
<https://commons.wikimedia.org/w/index.php?curid=32348020>

<sup>20</sup> Um trocadilho com o termo “glass” e o xingamento em inglês “asshole”, que pode ser livremente traduzido como “babaca”.

<sup>21</sup> Tradução: “O que pode e o que não pode fazer.”

<sup>22</sup> © University of Michigan. Cortesia. Disponível em:  
<http://hexus.net/ce/news/general/82216-michigan-micro-mote-worlds-smallest-autonomous-computer/>

<sup>23</sup> Tradução: “E mais uma coisa...”

<sup>24</sup> By Tscott3714 - Own work, CC BY-SA 4.0. Disponível em:  
<https://commons.wikimedia.org/w/index.php?curid=45118974>

<sup>25</sup> By No machine-readable author provided. Light Warrior assumed (based on copyright claims). - No machine-readable source provided. Own work assumed (based on copyright claims)., Public Domain. Disponível em:  
<https://commons.wikimedia.org/w/index.php?curid=7866702>

<sup>26</sup> By O01326 - Own work, CC BY-SA 4.0. Disponível em:  
<https://commons.wikimedia.org/w/index.php?curid=45399546>

<sup>27</sup> By Андрей Ильин - Бизнес-журнал, № 11/2014, Disponível em: <http://b-mag.ru/2014/tehnodrom/fabrika-zhizni/>, CC0.  
<https://commons.wikimedia.org/w/index.php?curid=37096132>

<sup>28</sup> No momento em que esse livro foi escrito, por exemplo, a Operação “Lava-Jato” da Polícia Federal só é possível na extensão que alcançou devido ao grande número de informações disponíveis eletronicamente, o que não existia nas décadas anteriores.

<sup>29</sup> By Jean Marc Cote (if 1900) or Villemard (if 1910) - Expositions BNF / Репродукция бумажной карточки, Public Domain. Disponível em:  
<https://commons.wikimedia.org/w/index.php?curid=15843412>

# Referências

- 3D Printing. What is 3d Printing? Disponível em:  
<http://3dprinting.com/what-is-3dprinting/>
- Aamoth, D. A Brief History of Skype. Time. 10 maio 2011. Disponível em: <http://techland.time.com/2011/05/10/a-brief-historyof-skype/>
- Aamoth, D. First Smartphone Turns 20: Fun facts about Simon. Time. 18 agosto 2014. Disponível em: <http://time.com/3137005/first-smartphone-ibm-simon/>
- AdaCore. The Ada Programming Language. Disponível em: <http://www.adacore.com/adaanswers/about/ada>
- AEITA. Zezinho, o primeiro computador a gente nunca esquece. 2013. Disponível em: <http://www.aeitaonline.com.br/wiki/index.php?title=Zezinho>
- Aiken, H. Proposed Automatic Calculating Machine. Harvard University Archives, 1937.
- All USB. USB History. Disponível em: <http://www.allusb.com/usb-history>
- Amadeo, R. Chrome OS can now Run Android Apps, no Porting Required. Ars Technica. 11 set 2014. Disponível em:

<http://arstechnica.com/gadgets/2014/09/chrome-os-can-now-run-android-apps-no-porting-required/>

Amadeo, R. The History of Android. Ars Technica. 15 jun 2014. Disponível em: <http://arstechnica.com/gadgets/2014/06/building-android-a-40000-word-history-of-googles-mobile-os/>

Antov, L. History of MS-DOS. Digital Research. 1996. Disponível em: <http://www.digitalresearch.biz/HISZMSD.HTM>

Armstrong, W.J. Mechanical and Electrical Equipment of the Toronto Union Station. Journal of the Engineering Institute of Canada. 1921; 4:87-97. Disponível em: [http://www.trainweb.org/oldtimetrains/TTR/mechanical\\_electrical\\_1921.htm](http://www.trainweb.org/oldtimetrains/TTR/mechanical_electrical_1921.htm)

Associação Nacional dos Inventores. Computador Patinho Feio. Disponível em <http://inventores.com.br/computador-patinho-feio/>

Association for Computer Machinery. ACM History. Disponível em: <http://www.acm.org/about/history>

Atanasoff, J. Computing Machine for the Solution of Large Systems of Linear Algebraic Equations. Iowa State University, 1940.

Atari Headquarters. The Tetris Saga. Disponível em: <http://www.atarihq.com/tsr/special/tetrislist.html>

Backus, J.W. Specifications for the IBM Mathematica FORmula TRANslating System – FORTRAN. 1954. Disponível em: <http://history-computer.com/Library/Fortran.pdf>

Badgley, S. The Twitter Fail Whale: Gone but Not Forgotten. About Tech. 31 jul 2014. Disponível em: <http://twitter.about.com/od/Twitter-Basics/fl/The-Twitter-Fail-Whale-Gone-but-Not-Forgotten.htm>

Baer, R. Genesis: How the videogames industry began. 1997. Disponível em: [http://www.ralphbaer.com/how\\_videogames.htm](http://www.ralphbaer.com/how_videogames.htm)

Bakken, S. (ed.) PHP Manual. PHP Documentation Group. 2000.

Barger, J. Weblog Resources FAQ. Wayback Machine. 1999. Disponível em:

<http://web.archive.org/web/20000817093828/http://www.robotwise.com/weblogs/>

Barnett, J.H. Origins of Boolean Algebra in the Logic of Classes: George Boole, John Venn and C. S. Peirce. Convergence. julho 2013. Disponível em: <http://www.maa.org/press/periodicals/convergence/origins-of-boolean-algebra-in-the-logic-of-classes-george-boole-john-venn-and-c-s-peirce>

Barret, L., Connell, M. Jevons and the Logic Piano. The Rutherford Journal. 2006. Disponível em: <http://rutherfordjournal.org/article010103.html>

Basulto, D. Why Google's new Quantum Computer Could Launch an Artificial Intelligence Arms Race. The Washington Post. 10 dez 2010. Disponível em: <https://www.washingtonpost.com/news/innovations/wp/2015/12/10/why-googles-new-quantum-computer-could-launch-an-artificial-intelligence-arms-race/>

Battle, J. Sol20.org. 2006. Disponível em: <http://www.sol20.org/>

Bauer, F.L., Wössner, H. The “Plankalkül” of Konrad Zuse: A Forerunner of Today’s Programming Languages. ACM. 1972. Disponível em: <http://www.catb.org/retro/plankalkuel/>

BBC News. How the Spectrum began a Revolution. 23 abril 2007. Disponível em: <http://news.bbc.co.uk/2/hi/technology/6572711.stm>

Bellis, M. The History of the Telephone – Antonio Meucci. About.com. 2015. Disponível em: [http://inventors.about.com/library/inventors/bl\\_Antonio\\_Meucci.htm](http://inventors.about.com/library/inventors/bl_Antonio_Meucci.htm)

Bellis, M. The Invention of the Intel 1103 - The World's First Available DRAM Chip. About.com. Disponível em: <http://inventors.about.com/library/weekly/aa100898.htm>

Bellis, M. Who Invented Wifi? About Money. 5 out 2015. Disponível em: <http://inventors.about.com/od/wstartinventions/a/WiFi.htm>

Bellón, F. M. El Primer Robot de Ajedrez era Español y Cumple Cien Años. Blogs ABC. 2012. Disponible em: <http://abcblogs.abc.es/poker-ajedrez/public/post/el-primer-robot-de-ajedrez-era-espanoly-cumple-cien-anos-14155.asp/>

Bennet, K. No One Remembers who Invented the Telephone. Motherboard. 17 jul 2012. Disponible em: <http://motherboard.vice.com/blog/alexander-graham-bell-did-not-invent-the-telephone>

Bergin, T.J. The Origins of Word Processing Software for Personal Computers: 1976-1985. IEEE Annals of the History of Computing. 2006; 28(4):32–47.

Berkeley, E.C., Jensen, R.A. World's Smallest Electric Brain. Radio-Electronics. 1950; Outubro. Disponible em: <http://history-computer.com/Library/Simon.pdf>

Berners-Lee, T. Information Management: A proposal. CERN. 1989. Disponible em: <https://www.w3.org/History/1989/proposal.html>

Berners-Lee, T. The WorldWideWebrowser. Disponible em: <https://www.w3.org/People/Berners-Lee/WorldWideWeb.html>

Berry, C. Manual for the ABC. 1941. Disponible em: <http://jva.cs.iastate.edu/img/ABC%20%20manual.pdf>

Best, J. IBM Watson: The inside story of how the Jeopardy-winning supercomputer was born, and what it wants to do next. TechRepublic. Disponible em: <http://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/>

Betters, E. Google Glass: A brief history. Pocket-lint. 15 jan 2015. Disponible em: <http://www.pocket-lint.com/news/132399-google-glass-a-brief-history>

- Bjornsson, M. The History of eBay.. 2001. Disponível em:  
<http://www.cs.brandeis.edu/~magnus/ief248a/eBay/history.html>
- Bolt, Beranek & Newman Inc. A History of the ARPANET: The First Decade. Darpa, 1981.
- Bonner, A. What was Llull up to? Disponível em:  
[http://www.ramonllull.net/sw\\_studies/studies\\_original/compon.html](http://www.ramonllull.net/sw_studies/studies_original/compon.html)
- Boole, G. An Investigations on the Laws of Thought, on which are founded the mathematical theories of logic and probabilities. Projeto Gutenberg. 1854. Disponível em:  
[http://www.gutenberg.org/files/15114/15114-pdf.pdf?session\\_id=b8846d0d3fad2dc923b3d23eae98942cd8ef9b1e](http://www.gutenberg.org/files/15114/15114-pdf.pdf?session_id=b8846d0d3fad2dc923b3d23eae98942cd8ef9b1e)
- Bosker, B. Siri Rising: The inside story of siri's origins - and why she could overshadow the iPhone. Huffpost Tech. 24 jan 2013. Disponível em: [http://www.huffingtonpost.com/2013/01/22/siri-do-engine-apple-iphone\\_n\\_2499165.html](http://www.huffingtonpost.com/2013/01/22/siri-do-engine-apple-iphone_n_2499165.html)
- Bright, P. Ten years of Windows XP: how longevity became a curse. Ars Technica. 25 out 2011. Disponível em:  
<http://arstechnica.com/information-technology/2011/10/ten-years-of-windows-xp-how-longevity-became-a-curse/>
- Brin, S., Page, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Seventh International WorldWide Web Conference; 1998; Brisbane, Austrália. Disponível em:  
<http://infolab.stanford.edu/~backrub/google.html>
- Brodie, L. Starting Forth. Nova Jersey: Prentice Hall, 1987. Disponível em: <http://www.forth.com/startng-forth/index.html>
- Brookhaven National Laboratory. The First Videogame? Disponível em:  
<https://www.bnl.gov/about/history/firstvideo.php>
- Brooks, F. The Mythical Men-Month: Essays on Software Engineering. Boston: Addison-Wesley, 1975.

- Brucker, J. A Brief history of MatLab. Lincoln: University of Nebraska-Lincoln. Disponível em: [http://cse.unl.edu/~lksoh/Classes/CSCE155N\\_Spring13/2004-01-2-History-Matlab-Jim.ppt](http://cse.unl.edu/~lksoh/Classes/CSCE155N_Spring13/2004-01-2-History-Matlab-Jim.ppt)
- Bush, V. As We May Think. The Atlantic Monthly. 1945; 176(1):101–8. Julho. Disponível em: <http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/>
- Bush, V. Instrumental Analysis. Bulletin of the American Mathematical Society. 1936; 42(10): 649-669. Disponível em: [http://projecteuclid.org/download/pdf\\_1/euclid.bams/1183499313](http://projecteuclid.org/download/pdf_1/euclid.bams/1183499313)
- Callaham, J. A (very) brief history of Windows XP. Neowin. 8 abril 2014. Disponível em: <http://www.neowin.net/news/a-very-brief-history-of-windows-xp>
- Campbell-Kelly, M., Aspray, W. Computer: A history of the information machine (The Sloan Technology Series). Westview Press. 3rd edition. 2013.
- Campos, A. Micro Sistemas. Museu da Computação e Informática. 2002. Disponível em: <http://www.mci.org.br/historia/ms/ms.html>
- Cardi, M. L., Barreto, J. M. Primórdios da Computação no Brasil. Simpósio de História da Informática na América Latina e Caribe. Medellín, 2012.
- Carlson, N. The Real History of Tweeter. Business Insider. 13 abril 2011. Disponível em: <http://www.businessinsider.com/how-twitter-was-founded-2011-4?op=1>
- Carpenter, B.E., Doran, R.W. The Other Turing Machine. The Computer Journal. 1977; 20(3): 269-279. Disponível em: <http://comjnl.oxfordjournals.org/content/20/3/269.full.pdf+html>
- Catambay, B. The Pascal Programming Language. Pascal Central. 2001. Disponível em: <http://pascal-central.com/ppl/index.html>

Centre for Computing History. Introduction of Intel 80286 at 6 MHz, with 134,000 transistors. Disponível em:  
<http://www.computinghistory.org.uk/det/6187/Introduction-of-Intel-80286-at-6-MHz-with-134-000-transistors/>

Centre for Computing History. Sinclair ZX Spectrum 48k. Disponível em: <http://www.computinghistory.org.uk/det/424/sinclair-zx-spectrum-48k/>

Cernosek, G. A Brief History of Eclipse. IBM. 2005. Disponível em:  
<http://www.ibm.com/developerworks/rational/library/nov05/cernosek/>

Ceruzzi, P. E. Computing: A concise history (The MIT Press Essential Knowledge Series). The MIT Press. 2012.

Chaarani, M.S. L'orgue hydraulique des Banu Mûsa. Muslim Heritage. Disponível em:  
<http://www.muslimheritage.com/article/I%E2%80%99orgue-hydraulique-des-banu-m%C3%BBsa-hydraulic-organ-banu-musa>

Chamberlin, D.D., Boyce, R.F. SEQUEL: A Structured English Query Language. Proceedings of the 1974 ACM SIGFIDET Workshop on Data Description, Access and Control. 1974; 249–64. Disponível em:  
<http://www.almaden.ibm.com/cs/people/chamberlin/sequel-1974.pdf>

Chapman, C. A Brief History of Blogging. WDD. 14 mar 2011. Disponível em: <http://www.webdesignerdepot.com/2011/03/a-brief-historyof-blogging/>

Chip Download. History of Microsoft Windows 95. 2015. Disponível em: [http://download.chip.eu/en/Historyof-Microsoft\\_166789879.html](http://download.chip.eu/en/Historyof-Microsoft_166789879.html)

Chip Download. History of Microsoft Windows 98. Disponível em: [http://download.chip.eu/en/Historyof-Microsoft-Windows-98\\_168419847.html](http://download.chip.eu/en/Historyof-Microsoft-Windows-98_168419847.html)

- Chirgwin, R. Nuke plants to rely on PDP-11 code UNTIL 2050! The Register. 2013. Disponível em: [http://www.theregister.co.uk/2013/06/19/nuke\\_plants\\_to\\_keep\\_pdp11\\_until\\_2050/](http://www.theregister.co.uk/2013/06/19/nuke_plants_to_keep_pdp11_until_2050/)
- Christopher, W. Commodore 64 at 30: Computing for the Masses. The Daily Telegraph. 4 ago 2012.
- Clark, K. M., Montelle, C. Logarithms: The Early History of a Familiar Function. Mathematical Association of America. 2011. Disponível em: <http://www.maa.org/publications/periodicals/convergence/logarithms-the-early-history-of-a-familiar-function>
- COBRA Tecnologia S.A. BB Tecnologia e Serviços. 2015. Disponível em: [http://www.cobra.com.br/index.php?option=com\\_content&view=article&id=219%3Aempresa-40anos&catid=49%3Amateria&Itemid=53](http://www.cobra.com.br/index.php?option=com_content&view=article&id=219%3Aempresa-40anos&catid=49%3Amateria&Itemid=53)
- Codd, E. F. A Relational Model of Data for Large Shared Data Banks. Communications of the ACM. 1970; 13(6):377-387. Disponível em: <http://history-computer.com/Library/codd.pdf>
- Coding Unit. The History of the C Language. Disponível em: <http://www.codingunit.com/the-history-of-the-c-language>
- Colmerauer, A., Roussel, P. The Birth of PROLOG. 1992. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.7438&rep=rep1&type=pdf>
- Colstad, K., Lipkin, E. Community Memory: a public information network. ACM SIGCAS Computers and Society. 1975; 6(4):6-7.
- Computer History Museum. Max Mathews Makes MUSIC. Disponível em: <http://www.computerhistory.org/revolution/computer-graphics-music-and-art/15/222>
- Computer History Museum. The Computer History Museum, SRI International, and BBN Celebrate the 40th Anniversary of First ARPANET Transmission. 2009. Disponível em:

<http://www.computerhistory.org/press/museum-celebrates-arpanet-anniversary.html>

Computer History Museum. The Cray-1 Supercomputer. 2015. Disponível em:  
<http://www.computerhistory.org/revolution/supercomputers/10/7>

Computer History Wiki. Windows 95. Disponível em:  
[http://gunkies.org/wiki/Windows\\_95](http://gunkies.org/wiki/Windows_95)

Computer Hope. Microsoft Windows 98. Disponível em:  
<http://www.computerhope.com/win98.htm>

Conscious Entities. Medieval Chat Bot. 2004. Disponível em:  
<http://www.consciousentities.com/lull.htm>

Cooper, S. Whatever Happened to Netscape? Engadget. 5 out 2014. Disponível em: <http://www.engadget.com/2014/05/10/history-of-netscape/>

Copeland, J. Colossus, the First Large-Scale Electronic Computer. Colossus. Disponível em: <http://www.colossus-computer.com/colossus1.html>

Coventry, Joshua. Interview with Dan Bricklin, Inventor of the Electronic Spreadsheet. Low End Mac. 2006. Disponível em: <http://lowendmac.com/2006/dan-bricklin-inventor-of-the-electronic-spreadsheet/>

Cray. Cray History. Disponível em:  
<http://www.cray.com/company/history>

Cringely, R. X. Accidental Empires: How the Boys of Silicon Valley Make Their Millions, Battle Foreign Competition, and Still Can't Get a Date. Nova York: Harper Business; 1996.

Cruz, F. Herman Hollerith. Columbia University. 2011. Disponível em:  
<http://www.columbia.edu/cu/computinghistory/hollerith.html>

Cruz, F. Hollerith 1890 Census Tabulator. Columbia University. 2011. Disponível em:  
<http://www.columbia.edu/cu/computinghistory/census-tabulator.html>

- Cruz, F. The Jacquard Loom. Columbia University. 2008. Disponível em: <http://www.columbia.edu/cu/computinghistory/jacquard.html>
- cSounds.com. Remembering Max – Atribute. Disponível em: <http://www.csounds.com/mathews/>
- Cuartero, F. Ramon Llull, el Ars Magna y la Informática. El País. 2012. Disponível em: <http://blogs.elpais.com/turing/2012/10/ramonllull-el-ars-magna-y-la-informatica.html>
- Cunningham, A. A Brief History of USB, what it Replaced, and what has Failed to Replace it. Ars Technica. 17 ago 2014. Disponível em: <http://arstechnica.com/gadgets/2014/08/a-brief-history-of-usb-what-it-replaced-and-what-has-failed-to-replace-it/>
- Da Cruz, F. The IBM Selective Sequence Electronic Calculator. Columbia University Computing History. 2015. Disponível em: <http://www.columbia.edu/cu/computinghistory/ssec.html>
- Dade, L. How the Enigma Machine Work. Enigma. 2015. Disponível em: <http://enigma.louisedade.co.uk/howitworks.html>
- Daftari, L. Iran bans WhatsApp because of link to ‘American Zionist’ Mark Zuckerberg. Fox News. 4 maio 2014. Disponível em: <http://www.foxnews.com/world/2014/05/04/iran-bans-whatsapp-because-link-to-american-zionist-mark-zuckerberg.html>
- Dahmke, M. The Compaq Portable. Byte. Janeiro 1983. Disponível em: [https://archive.org/stream/bytemagazine198301rescan/1983\\_01\\_BYTE\\_08-01\\_Looking\\_Ahead#page/n31/mode/2up](https://archive.org/stream/bytemagazine198301rescan/1983_01_BYTE_08-01_Looking_Ahead#page/n31/mode/2up)
- Dalakov, G. History of Computers, Hardware, Software, Internet... Disponível em: <http://history-computer.com/>
- Dartmouth College. BASIC. 1964. Disponível em: [http://bitsavers.informatik.uni-stuttgart.de/pdf/dartmouth/BASIC\\_Oct64.pdf](http://bitsavers.informatik.uni-stuttgart.de/pdf/dartmouth/BASIC_Oct64.pdf)
- Dasgupta, S. It Began with Babbage: The genesis of computer science. Oxford: Oxford University Press, 2014.

- Daskeyboard. Typing Through Time: Keyboard History. Disponível em: <http://www.daskeyboard.com/blog/typing-through-time-the-history-of-the-keyboard/>
- De Rosnay, J. History of Cybernetics and Systems Science. Principia Cybernetica Web. 2000. Disponível em: <http://pespmc1.vub.ac.be/cybhist.html>
- Dernbach, C. Apple LISA. Mac History. 2007. Disponível em: <http://www.machistory.net/apple-history-2/apple-lisa/2007-10-12/apple-lisa>
- Dernbach, C. The History of the Apple Macintosh. Mac History. 2011. Disponível em: <http://www.machistory.net/top/2011-01-24/the-history-of-the-apple-macintosh>
- Devotta, N. A Short History of Computer Viruses. Comodo Antivirus. 2014. Disponível em: <https://antivirus.comodo.com/blog/computer-safety/short-history-computer-viruses/>
- Dewey, C. Who is PewDiePie, the first person to ever hit 10 billion YouTube views? The Washington Post. 9 set 2015. Disponível em: <https://www.washingtonpost.com/news/the-intersect/wp/2015/09/09/who-is-pewdiepie-the-first-person-to-ever-hit-10-billion-youtube-views/>
- Dickey, M.R. The 22 Key Turning Points in the History of YouTube. Business Insider. 15 fev 2013.
- Digital 60. The Ferranti Mark I. Disponível em: <http://curation.cs.manchester.ac.uk/digital60/www.digital60.org/birth/manchestercomputers/mark1/ferranti.html>
- Dijkstra, E. The Humble Programmer. ACM Turing Lecture. Communications of the ACM. 1972; 15(10):859-866. Disponível em: <http://www.cs.utexas.edu/users/EWD/ewd03xx/EWD340.PDF>
- Disponível em: <http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA115440>

- Djuraskovic, O. Robot Wisdom and How Jorn Barger Invented Blogging. FirstSiteGuide. 20 mar 2015. Disponível em: <http://firstsiteguide.com/robotwisdom-and-jorn-barger/>
- Doornbusch, P. Computer Sound Synthesis in 1951: The music of CSIRAC. Computer Music Journal. 2004. Disponível em: <http://www.doornbusch.net/CSIRAC/CMJ28.1-Doornbusch-CSIRAC.pdf>
- Dornian, A. ReserVec: Trans-Canada Airlines' Computerized Reservation System. IEEE Annals of the History of Computing. 1994; 16(2):31-42.
- DuBois, R.L. The First Computer Musician. The New York Times. 2011. Disponível em: <http://opinionator.blogs.nytimes.com/2011/06/08/the-first-computer-musician/>
- Dudley, B. Scientist's Invention was let go for a song. The Seattle Times. 29 nov 2004. Disponível em: [http://old.seattletimes.com/html/businesstechnology/2002103322\\_cdman29.html](http://old.seattletimes.com/html/businesstechnology/2002103322_cdman29.html)
- Duisterhout, J.S., Franken, B. MUMPS as a host language for AIDA. Computer Methods and Programs in Biomedicine. 1987; 25(3):349-363, Novembro-Dezembro.
- Eadicicoo, L. The rise of Android: How a flailing startup became the world's biggest computing platform. Business Insider. 27 mar 2015. Disponível em: <http://www.businessinsider.com/how-android-was-created-2015-3>
- Early Computers Project. Computer Trainer Model 650. 2015. Disponível em: <http://www.earlycomputers.com/cgi-bin/item-report-main.cgi?20060219>
- Early Computers Project. Sphere 1. 2015. Disponível em: <http://www.earlycomputers.com/cgi-bin/item-report-main.cgi?20110729>
- Eastwood, B.S., Skrien, D.J. Maxwell, B.A. IAS Computer. Computer Organization. 2012. Disponível em:

<http://cs.colby.edu/courses/S12/cs232/ias.php>

Eccles, W.H., Jordan, F.W. Improvements in Ionic Relays. Patent Specification. GB148582(A), 1918. Disponível em: <http://worldwide.espacenet.com/publicationDetails/originalDocument?CC=GB&NR=148582&KC=&FT=E>

Eclipse. Eclipse Forms Independent Organization. 2004. Disponível em: <http://www.eclipse.org/org/press-release/feb2004foundationpr.php>

Edwards, B. Birth of a Standard: The Intel 8086 microprocessor. PCWorld. 2008. Disponível em: <http://www.pcworld.com/article/146957/article.html>

Edwards, D. Designing and Building Atlas. Resurrection: The Bulletin of the Computer Conservation Society. 2013; 62:9–18. Disponível em: <http://www.computerconservationsociety.org/resurrection/res62.htm#c>

Eiffel Software. Frequently Asked Questions on the Eiffel Language. Disponível em: <https://www.eiffel.com/resources/faqs/eiffel-language/>

Ellsbury, G. The enigma and the Bombe. 2003. Disponível em: <http://www.ellsbury.com/enigmabombe.htm>

Encyclopaedia Britannica. Disponível em: <http://global.britannica.com/>

Engineering and Technology History Wiki. Milestones: Fleming Valve, 1904. Disponível em: [http://ethw.org/Milestones:Fleming\\_Valve,\\_1904](http://ethw.org/Milestones:Fleming_Valve,_1904)

Engineering and Technology History Wiki. Oberlin Smith. Disponível em: [http://ethw.org/Oberlin\\_Smith](http://ethw.org/Oberlin_Smith)

Epp, S. Discrete Mathematics with Applications. Stamford: Cengage Learning, 2010.

Ewing, J. How MP3 was Born. Bloomberg Business. 5 mar 2007. Disponível em: <http://www.bloomberg.com/bw/stories/2007-03->

05/how-mp3-was-bornbusinessweek-business-news-stock-market-and-financial-advice

Falk, J. Things that Count: The rise and fall of calculators. Early Evolution of the Modern Calculator. 2014. Disponível em: <http://metastudies.net/pmwiki/pmwiki.php?n=Site.TheModernEpochAndTheEmergenceOfTheModernCalculator#Moreland>

Famous Scientists. Muhammad ibn Musa al-Khwarizmi. Disponível em: <http://www.famousscientists.org/muhammad-ibn-musa-al-khwarizmi/>

fCoder Group International. History of RS-232-C. 1998. Disponível em: [http://www.lookrs232.com/rs232/history\\_rs232.htm](http://www.lookrs232.com/rs232/history_rs232.htm)

Flanagan, C. The Bresenham Line-Drawing Algorithm. Tietojenkäsittelytieteen laitos. Disponível em: <http://www.cs.helsinki.fi/group/goa/mallinnus/lines/bresenh.html>

Fowler, M. Writing the Agile Manifesto. 2006. Disponível em: <http://martinfowler.com/articles/agileStory.html>

Ppt.com. History of PowerPoint: The Amazing Facts You Did Not Know. Disponível em: <http://www.free-power-point-templates.com/articles/historyof-powerpoint-the-amazing-facts-you-did-not-know/>

Fuller, M. Foreword. Throughout: Art and culture emerging with ubiquitous computing (Eckman, U. ed.). Cambridge: MIT Press, 2013.

Funding Universe. Sun Microsystems Inc. Disponível: <http://www.fundinguniverse.com/company-histories/sun-microsystems-inc-history/>

Gaskins, R. PowerPoint at 20: back to basics. ACM. 2007. Disponível em: <http://www.gbwizards.com/files/gaskins-ppt-at-20-cacm-vol50-no12-dec-2007-p15-p17.pdf>

Gentilviso, C. The 50 Worst Inventions - Clippy. Time. 27 maio 2010. Disponível em:

[http://content.time.com/time/specials/packages/article/0,28804,1991915\\_1991909\\_1991755,00.html](http://content.time.com/time/specials/packages/article/0,28804,1991915_1991909_1991755,00.html)

Giganews. 1979: Origins of Usenet; “A news”. Disponível em: <http://br.giganews.com/usenet-history/origins.html>

Glass Almanac. The History of Google Glass. 2015. Disponível em: <http://glassalmanac.com/history-google-glass/>

Goldberg, D. History of 3D Printing: It’s Older Than You Are (That Is, If You’re Under 30). Line/Shape/Space. 5 set 2014. Disponível em: <http://lineshapespace.com/historyof-3dprinting/>

Google. Nossa história a fundo. Disponível em: <https://www.google.com.br/about/company/history/>

Gray, M. Credits and Background. Cambridge: Massachusetts Institute of Technology. 1996. Disponível em: <http://www.mit.edu/~mkgray/net/background.html>

Grier, D.A. Human Computers: the first pioneers of the information age. Endeavor. 2001; 25(1): 28–32. Março.

Grier, D.A. When Computers were Humans. Princeton: Princeton University Press, 2007.

Grimes, W. Max Mathews, Pioneer in Making Computer Music, Dies at 84. The New York Times. 23 abril 2011. Disponível em: [http://www.nytimes.com/2011/04/24/arts/music/max-mathews-father-of-computer-music-dies-at-84.html?\\_r=0](http://www.nytimes.com/2011/04/24/arts/music/max-mathews-father-of-computer-music-dies-at-84.html?_r=0)

Guynn, J. Facebook’s WhatsApp hits 900 million users, aims for 1 billion. USA Today. 4 set 2015. Disponível em: <http://www.usatoday.com/story/tech/2015/09/04/whatsapp-facebook-900-million-mark-zuckerberg-jan-koum-messenger/71704760/>

Hamilton, M.H. Computer Got Loaded. *Datamation*. Cahners Publishing Company. 1º mar 1971.

Hassani, S.T. Al-Jazari: The Mechanical Genius. Muslim Heritage. Disponível em: <http://muslimheritage.com/article/al-jazari-mechanical-genius>

- Hauben, M., Hauben, R. Netizens: On the History and Impact of Usenet and the Internet. Columbia University. 1995. Disponível em: <http://www.columbia.edu/~rh120/ch106.x10>
- Hénin, S. Luigi Torchis Keyboard and Direct-Multiplication Calculator. Reflections on the History of Computing – Preserving memories and sharing stories. IFIP AICT 387. (Ed. Arthur Tatnall). Nova York: Springer, 2012.
- Herrick, J. The History of Ruby. Sitepoint. 26 jul 2014. Disponível em: <http://www.sitepoint.com/history-ruby/>
- Highsmith, J. History: The agile manifesto. 2001. Disponível em: <http://agilemanifesto.org/history.html>
- Hillis, W.D. Richard Feynman and the Connection Machine. Physics Today. 1989; 42(2). Disponível em: <http://longnow.org/essays/richard-feynman-connection-machine/>
- Hockey, T. Biographical Encyclopaedia of Astronomers. Nova York: Springer Science+Business Media, LLC; 2007.
- Hoffmann, T.V. IBM PCjr. Creative Computing. 1984; 10:3:74. Março. Disponível em: [http://www.atarimagazines.com/creative/v10n3/74\\_IBM\\_PCjr.php](http://www.atarimagazines.com/creative/v10n3/74_IBM_PCjr.php)
- Hormby, T. Acorn and the BBC Micro: From Education to Obscurity. Low End Mac. 2007. Disponível em: <http://lowendmac.com/2007/acorn-and-the-bbc-micro-from-education-to-obscurity/>
- Hormby, T. Think Different: The Ad Campaign that Restored Apple's Reputation. Low end Mac – Mac History. 2013. Disponível em: <http://lowendmac.com/2013/think-different-ad-campaign-restored-apples-reputation/>
- HP. History of the 9100A desktop calculator, 1968. 2012. Disponível em: <http://www.hp.com/hpinfo/abouthp/histnfacts/museum/personalsystems/0021/0021history.html>

- Huff, D. How the Eclipse Foundation Evolves to Stay Relevant. Opensource.com. 22 out 2013. Disponível em: <https://opensource.com/business/13/10/interview-mike-milinkovich-eclipse>
- Hurley, P.J. A Concise Introduction to Logic. Boston: Cengage Learning, 2011
- I Programmer. PC-1512 and the Fall of Amstrad. 2011. Disponível em: <http://www.i-programmer.info/history/machines/1364-alan-sugar-and-the-fall-of-amstrad.html>
- IBM Archives. 704 Data Processing System. Disponível em: [http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe\\_PP704.html](http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_PP704.html)
- IBM Archives. IBM 5100 Portable Computer. Disponível em: [http://www-03.ibm.com/ibm/history/exhibits/pc/pc\\_2.html](http://www-03.ibm.com/ibm/history/exhibits/pc/pc_2.html)
- IBM Archives. IBM 701. Disponível em: [http://www-03.ibm.com/ibm/history/exhibits/701/701\\_1415bx01.html](http://www-03.ibm.com/ibm/history/exhibits/701/701_1415bx01.html)
- IBM Archives. IBM/360 from Computers to Computer Systems. Disponível em: <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/system360/>
- IBM Archives. SAGE, The First National Air Defense Network. Disponível em: <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/sage/>
- IBM Archives. System/360 Announcement. Disponível em: [https://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe\\_PR360.html](https://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_PR360.html)
- IBM Archives. The Birth of the IBM PC. Disponível em: [https://www-03.ibm.com/ibm/history/exhibits/pc25/pc25\\_birth.html](https://www-03.ibm.com/ibm/history/exhibits/pc25/pc25_birth.html)
- IBM. Chronological History of IBM. Disponível em: [www-03.ibm.com/ibm/history/history/history\\_intro.html](http://www-03.ibm.com/ibm/history/history/history_intro.html)

- IBM. Deep Blue. Disponível em: <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/>
- IBM. IBM Floppy Disk. 2011. Disponível em: <http://ibmtechnologys.blogspot.com.br/2011/08/ibm-floppy-disk.html>
- IBM. Think. Disponível em: <http://www.ibm.com/ibm/think/>
- IEEE Standards Association. History of Ethernet. 2015. Disponível em: <http://standards.ieee.org/events/ethernet/history.html>
- Ifrah, G. The Universal History of Computing: From the Abacus to the Quantum Computer. Nova York: John Wiley & Sons, Inc., 2001.
- iMac History - The Complete History of the iMac. iMac Adverts. Disponível em: <http://www.imachistory.com/imac-adverts/>
- Indiana University. ARCHIVED: What is the History of Microsoft Windows? Disponível em: <https://kb.iu.edu/d/abwa>
- Instituto Newton C. Braga. As Válvulas: O que você precisa sobre esses componentes antigos! (V001). Disponível em: <http://www.newtoncbraga.com.br/index.php/mundo-das-valvulas/457-as-valvulas-o-que-voce-precisa-sobre-esses-componentes-antigos-v001>
- Intel. The Story of the Intel 4004. Disponível em: <http://www.intel.com.br/content/www/br/pt/history/museum-story-of-intel-4004.html>
- Internet Encyclopaedia of Philosophy. Gottlob Frege (1848-1925). Disponível em: <http://www.iep.utm.edu/frege/>
- Irvine, M.M. Early Digital Computers at Bell Telephone Laboratories. IEEE Anals of the History of Computing; 2001; julho-setembro, 22-42.
- Java T Point. History of Java. Disponível em: <http://www.javatpoint.com/historyof-java>
- Jevons, W.S. On the mechanical performance of logical inference. Philosophical Transactions of the Royal Society. 1869; 160: 497-

518. Disponível em:  
<https://archive.org/stream/philtrans01714956/01714956#page/n0/mode/2up>

Johnston S. Making the arithmometer count. Bulletin of the Scientific Instrument Society, 1997; 52:12-21. Disponível em <http://www.mhs.ox.ac.uk/staff/saj/arithmometer/>

Kay, A.C. A Personal Computer for Children of All Ages. Proc. Of the ACM National Conference; Agosto 1972; Boston, Estados Unidos.

Kay, A.C. The Early History of Smalltalk. Gagne. 1993. Disponível em:

<http://gagne.homedns.org/~tgagne/contrib/EarlyHistoryST.html>

Kay, A.C. The Early History of Smalltalk. In: Proceedings of HOPL II. "ACM SIGPLAN notices", Vol. 28, No. 3, Março. 1993. Conferência republicada em revista.

Kenbak Corp. Kenbak-1 The Training Computer. *Computerworld*. 17 nov 1971, p. 43.

Kimizuka, M. Historical Development of Magnetic Recording and Tape Recorder. National Museum of Nature and Science. Survey Reports on the Systemization of Technologies. 2012, Vol. 17.

Knuth, D. The Art of Computer Programming (TAOCP). Stanford University. Disponível em: <http://www-cs-faculty.stanford.edu/~knuth/taocp.html>

Konrad Zuse Internet Archives. Disponível em <http://zuse.zib.de/>

Krasner, M.A. Digital Encoding of Speech and Audio Signals Based on the Perceptual Requirements of the Auditory System [Ph. D. Thesis]. Massachusetts: MIT; 1979. Disponível em: <http://dspace.mit.edu/handle/1721.1/16011>

Krebs, R.E., Krebs, C.A. Groundbreaking Scientific Experiments, Inventions, and Discoveries of the Ancient World. Santa Barbara: Greenwood Press; 2003.

- Kurtz, T.E. BASIC. In: Wexelblat, R.L. (ed.) History of Programming Languages. Cambridge: Academic Press; 1981.
- Lamont, T. Napster: the day the music was set free. The Guardian. 24 fev 2013. Disponível em: <http://www.theguardian.com/music/2013/feb/24/napster-music-free-file-sharing>
- Landley, R. A Short Story of CP/M. Disponível em: <http://landley.net/history/mirror/cpm/history.html>
- Lapowsky, I. Happy Birthday, Facebook! A Look Back at 10 Years. Inc. 4 fev 2014. Disponível em: <http://www.inc.com/issie-lapowsky/happy-birthday-facebook.html>
- Lasar, M. 25 years of HyperCard—the missing link to the Web. Ars Technica. 30 mai 2012. Disponível em: <http://arstechnica.com/apple/2012/05/25-years-of-hypercard-the-missing-link-to-the-web/1/>
- Lathrop, D.H. To IV or not to IV: That is the question. Sovernet. 1996. Disponível em: <http://homepages.sover.net/~donnl/iiiivsiv.html>
- Leibniz, G.W. Explication de l'Arithmétique Binaire. Que se sert seul caractères 0 & 1; avec des Remarques sur son utilité, & sur ce qu'elle donne le sens des anciennes figures Chinoises de Fohy. IN: Histoire de L'Academie Royale des Sciences. Année M.DCC.III.... Seconde Edition, revûe, corrigée & augmentée. Paris: Chez Charles-Etienne Hochereau, 1720. Tradução em inglês disponível em: <http://www.leibniz-translations.com/binary.htm>
- Lemelson MIT Program. William Seward Burroughs. Disponível em: <http://lemelson.mit.edu/resources/william-seward-burroughs>
- Leonardo Torres y Quevedo. IEEE Computer Society. Disponível em: <http://history.computer.org/pioneers/torres.html>
- Lewis, P. H. Personal Computers; Of Chips and Byggy Whips: Reassessing Future Needs. The New York Times. 23 maio 1989. Disponível em: <http://www.nytimes.com/1989/05/23/science/personal->

[computers-of-chips-and-buggy-whips-reassessing-future-needs.html](#)

Lindgren, M. Glory and Failure: The Difference Engines of Johann Müller, Charles Babbage, and Georg and Edvard Sheutz (History of Computing). 2nd Edition. Massachusetts: MIT Press; 1990.

Linux Online Inc. Why a Penguin? 2008. Disponível em: <http://wayback.archive.org/web/20100815081131/http://www.linux.org/info/penguin.html>

Linzmayer, O.W. 30 Pivotal Moments in Apple's History. Macworld. 30 mar 2006. Disponível em: <http://www.macworld.com/article/1050112/30moments.html>

Logo Foundation. Logo History. 2015. Disponível em: [http://el.media.mit.edu/logo-foundation/what\\_is\\_logo/history.html](http://el.media.mit.edu/logo-foundation/what_is_logo/history.html)

Ludgate, P.E. On a Proposed Analytical Machine. Scientific Proceedings of the Royal Dublin Society. 1909; 12:77-91, Abril. Disponível em: <http://www.fano.co.uk/ludgate/paper.html>

MacDonald, K. The History of Atari. IGN. 20 mar 2014. Disponível em: <http://www.ign.com/articles/2014/03/20/ign-presents-the-history-of-atari>

MacKenzie, I. The Man Who Invented the Microprocessor. BBC News. 4 maio 2011. Disponível em: <http://www.bbc.com/news/technology-13260039>

Madrigal, A. IBM's First 100 Years: A Heavily Illustrated Timeline. The Atlantic. 16 jun 2011. Disponível em: <http://www.theatlantic.com/technology/archive/2011/06/ibms-first-100-years-a-heavily-illustrated-timeline/240502/>

Malinovsky, B.N. Pioneers of Soviet Computing. ed. Anne Fitzpatrick, 2006; 2nd ed., SIGCIS, 2010. Disponível em: [http://www.sigcис.org/files/SIGCISMС2010\\_001.pdf](http://www.sigcис.org/files/SIGCISMС2010_001.pdf)

Mangalindan, J.P. A History of the iPhone. Fortune. 26 set 2013. Disponível em: <http://fortune.com/2013/09/26/ahistory-of-the->

[iphone/](#)

- Marconi, A.M. History of Basic. q7basic.org. Disponível em:  
<http://www.q7basic.org/History%20of%20BASIC.pdf>
- Markoff, J. Computer Wins on ‘Jeopardy’: Trivial, It’s Not. The New York Times. 16 fev 2011. Disponível em:  
[http://www.nytimes.com/2011/02/17/science/17jeopardy-watson.html?pagewanted=all&\\_r=0](http://www.nytimes.com/2011/02/17/science/17jeopardy-watson.html?pagewanted=all&_r=0)
- Marquand, A. A New Logical Machine. 1885. Philos Mag. Outubro.
- Martin, J.A. Sexy Portable Storage. The Washington Post. 5 set 2007. Disponível em: <http://www.washingtonpost.com/wp-dyn/content/article/2007/09/05/AR2007090500370.html>
- Mason, B. Poqet PC® Web Site. Brian’s Place. 2011. Disponível em:  
<http://www.bmason.com/PoqetPC/>
- Massachusetts Institute of Technology. Multics. Disponível em:  
<http://web.mit.edu/multics-history/#history>
- Mastin, L. INDIAN MATHEMATICS – BRAHMAGUPTA. The Story of Mathematics. 2010. Disponível em:  
[http://www.storyofmathematics.com/indian\\_brahmagupta.html](http://www.storyofmathematics.com/indian_brahmagupta.html)
- Mathews, I. Commodore PET History. Commodore Computers. 2003. Disponível em: <http://www.commodore.ca/commodore-products/commodore-pet-the-worlds-first-personal-computer/>
- Matthews, I. MOS – The rise of the MOS technology & the 6502. Commodore Computers. 2003. Disponível em:  
<http://www.commodore.ca/commodore-history/the-rise-of-mos-technology-the-6502/>
- Matthews, I. The Commodore VIC-20. Commodore Computers. 2003. Disponível em:  
[http://www.commodore.ca/products/vic20/commodore\\_vic-20.htm](http://www.commodore.ca/products/vic20/commodore_vic-20.htm)
- Matzkin, J. Product Flashback: Poqet PC. PC Magazine. 14 nov 1989. Disponível em:  
<http://www.pcmag.com/article2/0,2817,2102143,00.asp>

Mauchly, J.W. The Use of High-Speed Vacuum Tube Devices for Calculating. Filadélfia: University of Pennsylvania 1942. Disponível em: [http://archive.computerhistory.org/resources/text/Knuth\\_Don\\_X4\\_100/PDF\\_index/k-8-pdf/k-8-u2600-Vacuum-Tubes-Calculating.pdf](http://archive.computerhistory.org/resources/text/Knuth_Don_X4_100/PDF_index/k-8-pdf/k-8-u2600-Vacuum-Tubes-Calculating.pdf)

Maximum PC. Palm: The rise and fall of a legend. TechnoBuffalo. 31 mar 2011. Disponível em: <http://www.technobuffalo.com/2011/03/31/palm-the-rise-and-fall-of-a-legend/>

May, K.T. How did supercomputer Watson beat Jeopardy champion Ken Jennings? Experts discuss. TEDBlog. 5 abril 2013. Disponível em: <http://blog.ted.com/how-did-supercomputer-watson-beat-jeopardy-champion-ken-jennings-experts-discuss/>

McCarthy, J. History of Lisp. Stanford: Stanford University. 1979. Disponível em: <http://www-formal.stanford.edu/jmc/history/lisp/lisp.html>

McCarthy, J. Reminiscences on the History of TimeSharing. Stanford: Stanford University. 1983. Disponível em: <http://www-formal.stanford.edu/jmc/history/timesharing/timesharing.html>

McCracken, H. “Osborne!”. *Technologizer*. 1º abril 2011. Disponível em: <http://www.technologizer.com/2011/04/01/osborne-computer/>

McCracken, H. Please Don’t Call It Trash-80: A 35th Anniversary Salute to Radio Shack’s TRS-80. Time. 3 ago 2012. Disponível em: <http://techland.time.com/2012/08/03/trs-80/>

McCullough, B. On the 20th Anniversary – The history of Yahoo’s founding. 1º mar 2015. Disponível em: <http://www.internethistorypodcast.com/2015/03/on-the-20th-anniversary-the-history-of-yahoos-founding/>

McGowan, L. How the PDP-11 was Born. Hamster. 1998. Disponível em: <http://hampage.hu/pdp-11/birth.html>

- McJones, P. History of ALGOL. Computer History Museum. 2015.  
<http://www.softwarepreservation.org/projects/ALGOL>
- McMillan, R. Her Code Got Humans on the Moon – and invented software itself. Wired. 13 out 2015. Disponível em:  
<http://www.wired.com/2015/10/margaret-hamilton-nasa-apollo/>
- Menabrea, L.F. Sketch of the Analytical Engine Invented by Charles Babbage. Bibliothèque Universelle de Genève, 1842; outubro, No. 82. Disponível em:  
<http://www.fourmilab.ch/babbage/sketch.html>
- Mesa, A.F. The early Mac OS. The Apple Museum. 1997. Disponível em: <http://apple museum.bott.org/sections/os.html>
- Metz, C. Innovators: Alan Emtage. PC Magazine. 4 abril 2007. Disponível em:  
<http://www.pc mag.com/article2/0,2817,2108909,00.asp>
- Meyer, R. Even Early Focus Groups Hated Clippy. The Atlantic. 23 jun 2015. Disponível em:  
<http://www.theatlantic.com/technology/archive/2015/06/clippy-the-microsoft-office-assistant-is-the-patriarchys-fault/396653/>
- Michael, G. The PDP-1. Stories of the Development of Large Scale Scientific Computing. Disponível em:  
<http://www.computerhistory.info/Page4.dir/pages/PDP.1.dir/>
- Michigan University. Michigan Micro Mote (M<sup>3</sup>) Makes History. 17 mar 2015. Disponível em:  
<http://www.eecs.umich.edu/eecs/about/articles/2015/Worlds-Smallest-Computer-Michigan-Micro-Mote.html>
- Microsoft. Introduction to the C# Language and the .NET Framework. Disponível em: <https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx>
- Microsoft. Uma História do Windows. 2015. Disponível em:  
<http://windows.microsoft.com/pt-br/windows/history#T1=era4>
- Miller, M. A Brief History of Microsoft Windows. Pearson. 2009. Disponível em:

[http://www.quepublishing.com/articles/article.aspx?  
p=1358665&seqNum=6](http://www.quepublishing.com/articles/article.aspx?p=1358665&seqNum=6)

Miller, M.J. Creating the 8080: The Processor That Started the PC Revolution. PC Magazine. 18 dez 2014. Disponível em: <http://forwardthinking.pcmag.com/none/330501-creating-the-8080-the-processor-that-started-the-pc-revolution>

Mindell, D.A. Cybernetics – Knowledge domains in engineering systems. Massachusetts: MIT; 2000. Disponível em: <http://web.mit.edu/esd.83/www/notebook/Cybernetics.PDF>

MIT, Lincoln Laboratory. The SAGE Air Defense System. 2015. Disponível em: <https://www.ll.mit.edu/about/History/SAGEairdefensesystem.htm>

Moler, C. The Origins of MATLAB. MathWorks. 2004. Disponível em: <http://www.mathworks.com/company/newsletters/articles/the-origins-of-matlab.html>

Moore, C.H. The Evolution of Forth, an Unusual Language. *Byte*. 1980.

Morrison, J.E. The Astrolabe. Cambridge: Janus; 2007.

Museu da Computação e Informática – MCI. IBM. Disponível em: <http://www.mci.org.br/fabricante/ibm.html>

Museum Victoria. CSIRAC, Australia's First Computer. Disponível em: <http://museumvictoria.com.au/csirac/index.aspx>

Naur, P. The European Side of the last Phase of the Development of ALGOL 60. In: Wexelblat, R.L. (ed.) History of Programming Languages. Cambridge: Academic Press; 2014.

Network Working Group. PARRY Encounter the DOCTOR. 1973. Disponível em: <https://tools.ietf.org/html/rfc439>

Newell, A., Simon, H.A. The Logic Theory Machine – A complex information processing system. Santa Mônica: The Rand Corporation; 1956. Disponível em: <https://www.u->

[picardie.fr/~furst/docs/Newell\\_Simon\\_Logic\\_Theory\\_Machine\\_1956.pdf](http://picardie.fr/~furst/docs/Newell_Simon_Logic_Theory_Machine_1956.pdf)

Nichols, S. IBM PCjr Stripped Bare: We tear down the machine Big Blue would rather you forgot. The Register. 22 mar 2014. Disponível em: [http://www.theregister.co.uk/2014/03/22/ibm\\_pcjr\\_stripped\\_bare\\_still\\_mediocre\\_after\\_all\\_these\\_years/](http://www.theregister.co.uk/2014/03/22/ibm_pcjr_stripped_bare_still_mediocre_after_all_these_years/)

Nieva R. Ashes to Ashes, Peer to Peer: An oral history of Napster. Fortune. 5 set 2013. Disponível em: <http://fortune.com/2013/09/05/ashes-to-ashes-peer-to-peer-an-oral-historyof-napster/>

Nobel Prize. History and Properties of Liquid Crystals. Disponível em: [http://www.nobelprize.org/educational/physics/liquid\\_crystals/history/](http://www.nobelprize.org/educational/physics/liquid_crystals/history/)

Norberg, A.L. Software Development at the Eckert-Mauchly Computer Company Between 1947 and 1955. Iterations. 2003. Disponível em: <http://www.cbi.umn.edu/iterations/norberg.html>

Nori, S. How many Engineers does it take to Change a Lightbulb? Boiled Beans. 2009. Disponível em: <http://www.boiledbeans.net/2009/04/20/how-many-engineers-does-it-take-to-change-a-lightbulb/>

Norman, J. The Contributions of Vannevar Bush to Analog Computing, Information Retrieval, and the Concept of Hypertext. History of Information. 2002. Disponível em: <http://www.historyofinformation.com/expanded.php?id=869>

NuSphere. PHP History. Disponível em: [http://www.nusphere.com/php/php\\_history.htm](http://www.nusphere.com/php/php_history.htm)

O'Connor, J.J., Robertson, E.F. Brahmagupta. MacTutor History of Mathematics. 2000. Disponível em: <http://www-history.mcs.st-and.ac.uk/Biographies/Brahmagupta.html>

O'Connor, J.J., Robertson, E.F. Wilhelm Schickard. MacTutor History of Mathematics. 2009. Disponível em: <http://www-history.mcs.st-and.ac.uk/Biographies/Schickard.html>

[andrews.ac.uk/Biographies/Schickard.html](http://andrews.ac.uk/Biographies/Schickard.html)

O'Reilly Media Inc. The Perl Camel Usage and Trademark Information. 2016. Disponível em:  
<http://www.onlamp.com/pub/a/oreilly/perl/usage/>

Oberlin Smith. Ferracute Goes to China – 1897. Disponível em:  
<http://www.oberlinsmith.org/>

Obsolete Computer Museum. Compac Portable. Disponível em:  
<http://www.obsoletecomputermuseum.org/compaq/>

Obsolete Technology Website. Compucolor 8001. Disponível em:  
<http://www.oldcomputers.net/compucolor-8001.html>

Oikarinen, J. IRC History. IRC.org. Disponível em:  
[http://www irc.org/history\\_docs/jarkko.html](http://www irc.org/history_docs/jarkko.html)

Old Computers. Vax 11/780, The First VAX System. Disponível em:  
<http://www.oldcomputers.com/history/detail.asp?n=20&t=3>

Oliveira, J.P. I SEMISH – 1974. Site do Prof. Palazzo. Disponível em: [http://palazzo.pro.br/Joomla/pt-BR/homepage/historia/194-i-semish-1974?  
highlight=WyJpliwic2VtaXNoliwiaSBzZW1pc2giXQ==](http://palazzo.pro.br/Joomla/pt-BR/homepage/historia/194-i-semish-1974?highlight=WyJpliwic2VtaXNoliwiaSBzZW1pc2giXQ==)

Olley A. The SSEC First Electronic Machine on the Silver Screen. IT History Society. 2015. Disponível em: <http://ithistory.org/blog/?p=2333>

Operating System. Windows 98. Disponível em:  
[http://www.operating-system.org/betriebssystem/\\_english/bs-win98.htm](http://www.operating-system.org/betriebssystem/_english/bs-win98.htm)

Oracle. The History of Java Technology. Disponível em:  
<http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>

Osborn, J. Deep Inside C#: An Interview with Microsoft Chief Architect Anders Hejlsberg. O'Reilly. 8 jan 2000. Disponível em:  
[http://www.windowsdevcenter.com/pub/a/oreilly/windows/news/hejlsberg\\_0800.html](http://www.windowsdevcenter.com/pub/a/oreilly/windows/news/hejlsberg_0800.html)

- Oxford, T. Getting connected: a history of modems. Techradar.pro. 2009. Disponível em: <http://www.techradar.com/news/internet/getting-connected-a-history-of-modems-657479/2>
- Page-Bucci, H. CD-ROM Technology and Development – Is there a future? HKA Designs. 2002. Disponível em: <http://www.hkadesigns.co.uk/websites/msc/foun/index.htm>
- Parliament.UK. History of the Parliamentary Estate. Disponível em: <http://www.parliament.uk/about/living-heritage/building/palace/estatehistory/from-the-parliamentary-collections/fire-of-westminster/tallysticks/>
- Paul, F. At 25, Tetris Still Eyeing Growth. Reuters. 2 jun 2009. Disponível em: <http://www.reuters.com/article/us-video-games-tetris-idUSTRE5510V020090602>
- Peddie, J. The History of Visual Magic in Computers. Nova York: Springer; 2013
- Peirce, C.S. Logical Machines. The American Journal of Psychology. 1887; 1:165-170. Disponível em: <http://history-computer.com/Library/Peirce.pdf>
- Perlis, A. J., Samelson, K. Preliminary Report – International Algebraic Language. ACM-GAMM Committee. 1958.
- Phillips, S. A Brief History of Facebook. The Guardian. 25 jul 2007. Disponível em: [http://www.theguardian.com/technology/2007/jul/25/media.new\\_media](http://www.theguardian.com/technology/2007/jul/25/media.new_media)
- PHP/FI Version 2.0. Brief History. Disponível em: <http://php.net/manual/phpfi2.php#history>
- Picard, A. The History of Twitter, 140 Characters at a Time. The Globe and Mail. 20 mar 2011. Disponível em: <http://www.theglobeandmail.com/technology/digital-culture/the-history-of-twitter-140-characters-at-a-time/article573416/>

Pichai, S. Introducing the Google Chrome OS. Official Google Blog. 7 jul 2009. Disponível em: <http://googleblog.blogspot.com/2009/07/introducing-google-chrome-os.html>

Pierce, D. The Mac Turns 30: A visual history. The Verge. 24 jan 2014. Disponível em: <http://www.theverge.com/2014/1/24/5340320/the-mac-turns-30-a-visual-history>

Pixar Animation Studios. The History of Pixar Animation Studios. Disponível em: <http://pixar-animation.weebly.com/journal/the-history-of-pixar-animation-studios>

Pixar. The Pixar Timeline – 1979 to Present. Disponível em: <http://www.pixar.com/about/Our-Story>

Pollack, A. Fifth Generation' Became Japan's Lost Generation. The New York Times. 5 jun 1992. Disponível em: <http://www.nytimes.com/1992/06/05/business/fifth-generation-became-japan-s-lost-generation.html>

Powel, D.B. From Basement to Boardroom. PC Magazine. 7 fev 1984. Disponível em: [https://books.google.com.br/books?id=knOwB0kBuYQC&pg=PA135&lpg=PA135&dq=dbase+ii+history&source=bl&ots=wqwRaaj8Wy&sig=fMpVDQk8tHze4kCmRRr9CnCjXC&hl=pt-BR&sa=X&ved=0ahUKEwiTyOOv0u\\_JAhVJHZAKHTuIAjwQ6AEIUTAG#v=onepage&q=dbase%20ii%20history&f=false](https://books.google.com.br/books?id=knOwB0kBuYQC&pg=PA135&lpg=PA135&dq=dbase+ii+history&source=bl&ots=wqwRaaj8Wy&sig=fMpVDQk8tHze4kCmRRr9CnCjXC&hl=pt-BR&sa=X&ved=0ahUKEwiTyOOv0u_JAhVJHZAKHTuIAjwQ6AEIUTAG#v=onepage&q=dbase%20ii%20history&f=false)

Powers, J. November 15, 1971: Intel 4004 Microprocessor. Day in tech history/Intel. 15 nov 2015. Disponível em: <http://dayintechhistory.com/dith/november-15-1971-intel-4004-microprocessor/>

Project Gutenberg. Disponível em: <https://www.gutenberg.org/>

Purple. The evolution of Wi-Fi. 27 maio 2014. Disponível em: <http://purple.ai/history-wifi/>

R2E. Micral N Users Manual. Courtaboeuf, França: 1974. Disponível em:

[http://bitsavers.org/pdf/r2e/MICRAL\\_N\\_Users\\_Manual\\_Jan74.pdf](http://bitsavers.org/pdf/r2e/MICRAL_N_Users_Manual_Jan74.pdf)

- Rakos, M. History of Instagram. DashBurst. 12 maio 2014. Disponível em: <https://blog.dashburst.com/historyof-instagram/>
- Randell, B. From Analytical Engine to electronic Digital Computer: The contributions of Ludgate, Torres, and Bush. Annals of the History of Computing; 1982; 4(4), Outubro.
- Recording History. Inventing the Wire Recorder. Disponível em: <http://www.recording-history.org/HTML/wire2.php>
- Redondo, F.A. Mirando Hacia Atrás. La Gaceta de la RSME. 2004; 7.3: 787-810.
- Reimer, J. A history of the Amiga, part 1: Genesis. Ars Technica. 1º ago 2007. Disponível em: <http://arstechnica.com/gadgets/2007/07/ahistory-of-the-amiga-part-1/>
- Reimer, J. A history of the Amiga, part 2: The birth of Amiga. Ars Technica. 13 ago 2007. Disponível em: <http://arstechnica.com/gadgets/2007/08/ahistory-of-the-amiga-part-2/>
- Reimer, J. A history of the Amiga, part 3: The first prototype. Ars Technica. 22 ago 2007. Disponível em: <http://arstechnica.com/gadgets/2007/08/ahistory-of-the-amiga-part-3/>
- Reimer, J. A history of the Amiga, part 4: Enter Commodore. Ars Technica. 22 out 2007. Disponível em: <http://arstechnica.com/gadgets/2007/10/amiga-history-4-commodore-years/>
- Reimer, J. A history of the Amiga, part 5: postlaunch blues. Ars Technica. 10 dez 2007. Disponível em: <http://arstechnica.com/gadgets/2007/12/amiga-history-part-5/>
- Reimer, J. Half an Operating System: The triumph and tragedy of the OS/2. Ars Technica. 24 nov 2013. Disponível em:

<http://arstechnica.com/business/2013/11/half-an-operating-system-the-triumph-and-tragedy-of-os2/>

Richardson, M. Larry Wall, the guru of Perl. Linux Journal. 1º mar 1999. Disponível em: <http://www.linuxjournal.com/article/3394>

Riordan, M. The Lost History of the Transistor. IEEE Spectrum. 30 abril 2004. Disponível em: <http://spectrum.ieee.org/biomedical/devices/the-lost-history-of-the-transistor>

Rios, J.M. Leonardo Torres Quevedo y sus Inventos o um Inventor Universal. Compujedrez. 2014.

Ritchie, E. History of iPhone: Apple reinvents the phone. iMore. 31 ago 2015. Disponível em: <http://www.imore.com/history-iphone-original>

Roberts, L.G. The Evolution of Packet Switching. Packet. 1978. Disponível em: <http://www.packet.cc/files/ev-packet-sw.html>

Robinson, P., Edwards, J.R. The Atari 1040ST. Byte. Março 1986. Disponível em: [https://archive.org/stream/byte-magazine-1986-03/1986\\_03\\_BYTE\\_11-03\\_Homebound\\_Computing#page/n91/mode/2up](https://archive.org/stream/byte-magazine-1986-03/1986_03_BYTE_11-03_Homebound_Computing#page/n91/mode/2up)



# **Automatização de Teste de Software com Ferramentas de Software Livre**

Maldonado, Jose 9788535288162

256 páginas

[Compre agora e leia](#)

As atividades de Verificação, Validação e Teste (VV&T) são fundamentais para o desenvolvimento de produtos de software de alta qualidade. Em particular, as atividades de Teste de Software representam um custo significativo para as empresas para o desenvolvimento e evolução de produtos de software ou de sistemas intensivos em software, como sistemas de sistemas. No livro Introdução ao Teste de Software (Editora Elsevier, 2016, 2a edição), tratamos dos conceitos, critérios, técnicas e estratégias de teste do ponto de vista teórico, com exemplos e ilustrações mais acadêmicos. Neste livro, os temas são abordados com ênfase na automatização da aplicação das

atividades de teste com ferramentas e ambientes open source. Assim, para cada um dos temas, são apresentados e discutidos alguns ambientes e ferramentas de automatização dos critérios e técnicas abordados, inclusive com instruções para a instalação e aplicação das ferramentas. São fornecidas também diretrizes para avaliação da adequação dessas ferramentas e ambientes, de acordo com o Modelo de Aceitação de Tecnologia ? TAM (Technology Acceptance Model), fundamental para atomada de decisão no aprimoramento de processos.

[Compre agora e leia](#)

# HISTÓRIA GERAL DO BRASIL

Maria Yedda Leite Linhares (*in memoriam*)

Francisco Carlos Teixeira da Silva

Ciro Flamarion Santana Cardoso (*in memoriam*)

Hamilton de Matos Monteiro (*in memoriam*)

João Luis Ribeiro Fragoso

Karl Schurter de Sousa Leão

Marcello Otávio Neri de Campos Basile

Rafael Pinheiro de Araújo

Ricardo Pinto dos Santos

Sonia Regina de Mendonça

10ª edição

Atualizada e ampliada

ELSEVIER

# História Geral do Brasil

Linhares, Maria 9788535285444

528 páginas

[Compre agora e leia](#)

História Geral do Brasil é um balanço moderno, atual e renovador do conjunto da História do Brasil, com a participação de historiadores consagrados por suas pesquisas e livros sobre os diversos períodos da nossa história e de grandes universidades brasileiras – UFRJ, UFF, UERJ, UPE e UnB. Em linguagem simples, direta e rigorosa o livro percorre a história do país desde a colonização até os nossos dias, com ênfase nas características gerais da colonização, na formação dos grupos sociais brasileiros, no Império e suas grandes particularidades, incluindo a escravidão e sua superação. Analisamos também a criação da República e a ampliação da participação política no Brasil, suas crises, transformações, e seus

sucessos, com a emergência da classe operária, da urbanização e da industrialização. Na 10a edição do livro apresentamos novas abordagens da República no Brasil, em especial do fim do Regime Civil-Militar de 1964-1985, o papel do Congresso Constituinte, as mudanças e continuidades entre as instituições republicanas antes e depois de 1985, momento da mudança do regime autoritário para uma democracia no Brasil. Nesse novo capítulo, abordamos a criação da "Nova República" e suas crises, o papel dos partidos políticos e dos grandes programas sociais e econômicos que fizeram do Brasil – de uma grande nação agrária e atrasada – uma das maiores economias do mundo. Analisamos, ainda, o papel do país, e sua postura nas relações internacionais, sua integração na Globalização e a busca de um papel de grande player no mundo de hoje. Por fim, tratamos, de forma original e inédita, o papel dos esportes, em especial do futebol, na sociedade brasileira e o impacto dos megaeventos na sociedade brasileira, inclusive a explosão das chamadas Jornadas de Junho (de 2013) e a emergência de uma grave crise social e política ao

fim da "Era Lula". Trata-se de um livro para os semestres iniciais de História, Geografia, Ciências Sociais, Comunicação Social, Serviço Social, Relações Internacionais e Defesa, além de apresentar uma visão geral para o Ensino Médio e o grande público. ?

[Compre agora e leia](#)



MARCELO DE PAIVA ABREU  
ORGANIZADOR

# A ORDEM DO PROGRESSO

DOIS SÉCULOS DE POLÍTICA  
ECONÔMICA NO BRASIL



NOVA EDIÇÃO REVISTA  
E ATUALIZADA



# A Ordem do Progresso

Abreu, Marcelo de Paiva 9788535279511

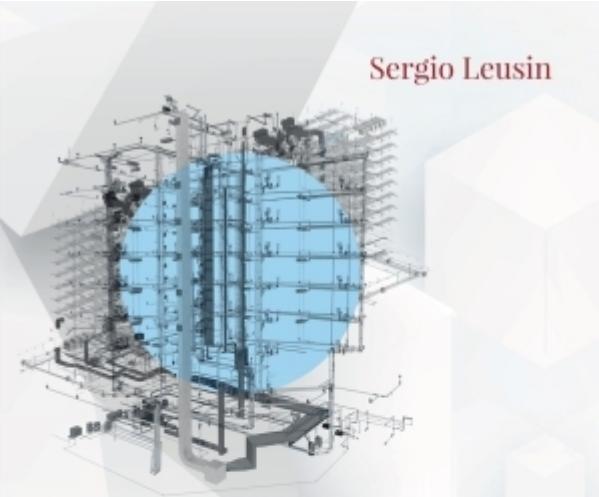
472 páginas

[Compre agora e leia](#)

A primeira edição de A Ordem do Progresso foi publicada há quase um quarto de século, em comemoração ao centenário da República. Muitas crises tiveram de ser enfrentadas. Com a vitória da oposição, a transição em 2002-2003 revelou-se menos problemática do que se temia, com o Partido dos Trabalhadores abandonando às pressas os seus excessos mais impetuosos como o repúdio das dívidas interna e externa. Parecia que se assistia ao fim de ideias equivocadas em matéria de política econômica. Na esteira do mensalão, em 2004-05, o compromisso petista com políticas macroeconômicas prudentes começou a arrefecer. De fato, a partir de 2010, acumularam-se indícios claros de reversão das políticas que haviam sido

estabelecidas na década de 1990 quanto à abertura comercial e ao papel do Estado na economia. Até mesmo o compromisso com a estabilização passou a ser relativizado. Esta nova edição pode ser vista como comemoração antecipada dos dois séculos do Brasil independente e contém artigos de Marcelo de Paiva Abreu, Dionísio Dias Carneiro, Gustavo Franco, Winston Fritsch, Luiz Aranha Correa do Lago, Eduardo Modiano, Luiz Orenstein, Demósthenes Madureira de Pinho Neto, André Lara Resende, Antonio Claudio Sochaczewski e Sérgio Besserman Vianna. ?

[Compre agora e leia](#)



Sergio Leusin

# Gerenciamento e coordenação de Projetos BIM

UM GUIA DE FERRAMENTAS E BOAS PRÁTICAS  
PARA O SUCESSO DE EMPREENDIMENTOS

ELSEVIER

Material  
na WEB  
[www.elsevier.com.br](http://www.elsevier.com.br)

# **Gerenciamento e Coordenação de Projetos BIM**

Roberto Leusin, Sergio 9788535290059

168 páginas

[Compre agora e leia](#)

Building Information Modeling (BIM): sua eficiência no gerenciamento e na coordenação. Sérgio Roberto Leusin apresenta indicadores e ferramentas usadas no gerenciamento dos projetos, além de abordar as características dos softwares mais comuns do mercado. Com destaque para a área imobiliária e grandes organizações proprietárias, Gerenciamento e Coordenação de Projetos BIM traz as boas práticas no gerenciamento e coordenação de projetos, inclusive com exemplos de procedimentos de contratação e controle de projeto. O autor ainda faz a análise de funcionalidades para efeito de seleção da solução.

[Compre agora e leia](#)

LUIZ PAULO FÁVERO | PATRÍCIA BELFIORE

MANUAL DE

# ANÁLISE DE DADOS

Estatística e Modelagem Multivariada com  
Excel®, SPSS® e Stata®



ELSEVIER

Material  
de Apoio

# Manual de Análise de Dados

Favero, Luiz Paulo 9788535285055

1216 páginas [Compre agora e leia](#)

Este livro, em que são abordadas as principais técnicas estatísticas e de modelagem multivariada, é resultado de vários anos de estudo e pesquisa e enfatiza a importância da análise de dados em ambientes acadêmicos e organizacionais, podendo ser considerado o principal fruto de inúmeras discussões e elucubrações sobre a importância da modelagem aplicada voltada à tomada de decisão. O princípio básico do livro consiste em explicitar, a todo instante, a hierarquia entre dados, informação e conhecimento . Os dados, quando tratados e analisados, transformam-se em informações. Já o conhecimento é gerado no momento em que tais informações são reconhecidas e aplicadas na tomada de decisão. Analogamente, a hierarquia reversa também pode ser aplicada, visto que o conhecimento, quando difundido ou explicitado,

torna-se determinada informação que, quando desmembrada, tem capacidade para gerar um conjunto de dados.

[Compre agora e leia](#)