

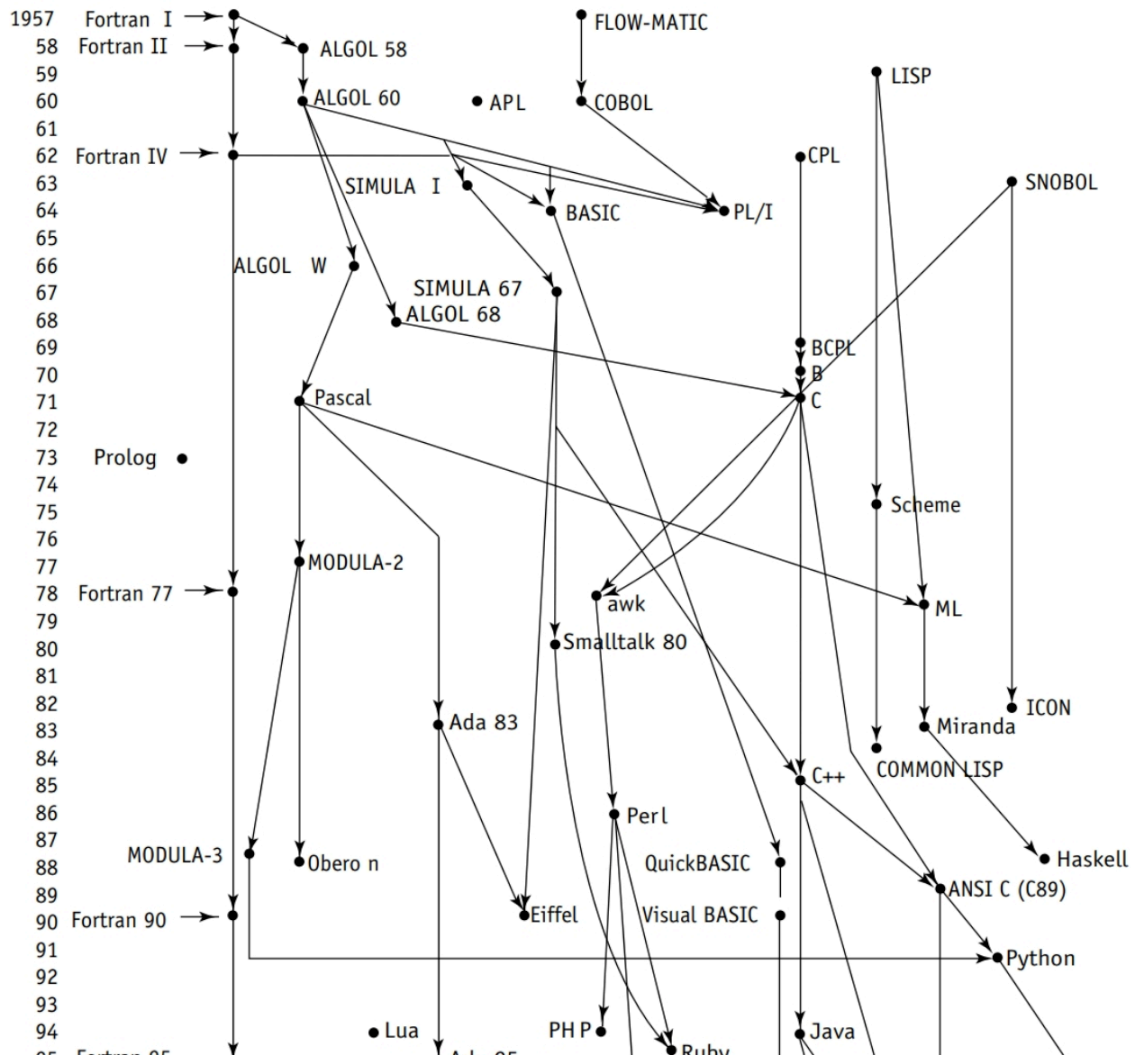


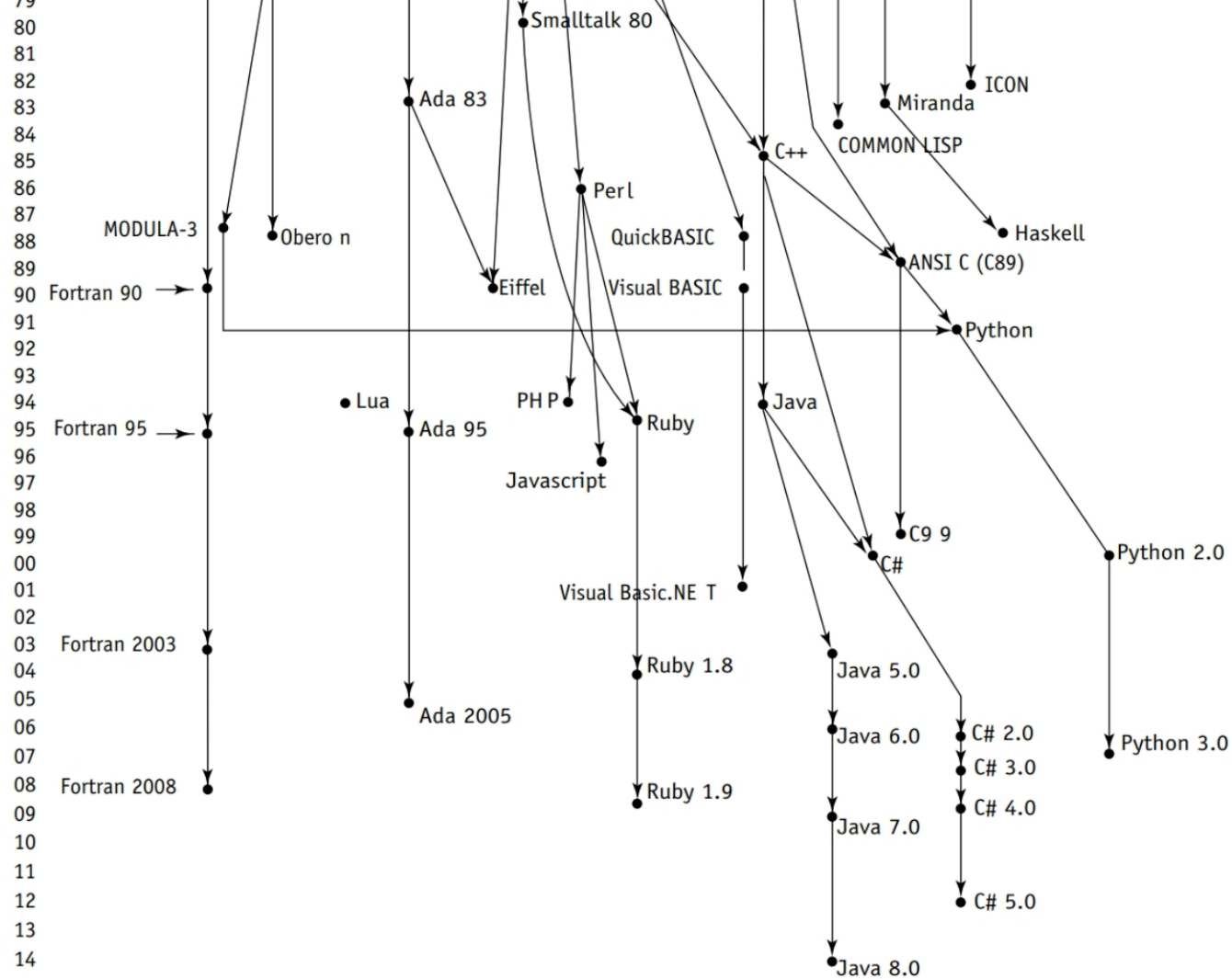
Histórico das LPs

A decorative graphic on the left side of the slide, featuring a cluster of hexagons in various shades of blue and cyan. Some hexagons are solid, while others are outlines. A small, dark blue icon resembling a network or molecular structure is also visible.

1

Histórico







1945: Plankakül de Zuse

- ◇ Nunca foi implementada
- ◇ Publicada em 1972
- ◇ Cientista alemão Konrad Zuse
 - Computadores com relés eletromecânicos
 - 1945: II GM destruiu todos, exceto o Z4
 - Linguagem para expressar computação no Z4



1945: Plankakül de Zuse

- ◇ Tipo bit, inteiro e float, vetores e registros
 - Registros aninhados
- ◇ For, mas não goto
- ◇ if, mas não else
- ◇ expressões matemática

		A + 1 => A
V		4 5
S		1.n 1.n

Equivalente a

$A[5] = A[4] + 1$





1945: Plankakül de Zuse

- ◇ No manuscrito publicado em 1972:
 - algoritmo para ordenar números
 - testar conectividade em grafos
 - análise sintática em fórmulas lógicas
 - 49 páginas de algoritmo para jogar xadrez





1940 - 50: Programação de Hardware

- ◇ Pseudocódigos, como eram chamadas
 - Hoje não podem mais ser consideradas pseudocódigos
- ◇ Eram utilizados apenas códigos de máquina
 - Códigos numéricos: 14 para a instrução ADD, por exemplo
- ◇ Endereçamento absoluto
 - Imagine fazer ADD e inserir no meio de um vetor?





1940 - 50: Programação de Hardware

- ◇ "Compilação" da UNIVAC (1951-53)
 - A-0, A-1 e A-2
 - Expandia pseudocódigos em código de máquina
- ◇ Universidade de Cambridge (1950)
 - Blocos de endereços para resolver o problema de endereçamento absoluto





1954: IBM 704 e o FORTRAN

- ◇ As capacidades do 704 permitiram o desenvolvimento do FORTRAN
- ◇ Indexação e instruções em ponto flutuante direto no hardware
- ◇ Até 1950 todas operações com ponto flutuante eram simuladas em software: muito lento!





1954: IBM 704 e o FORTRAN

- ◇ FORTRAN 0 (1954)
 - FORMula TRANslating System
- ◇ FORTRAN 1 (1957)
 - Formatação de entrada e saída
 - Nomes de variáveis de até 6 caracteres
 - Subprogramas, if e do
 - Variáveis iniciadas com I, J, K, L, M e N eram do tipo inteiro, as outras, de ponto flutuante
 - 18 trabalhadores-ano para preparar a linguagem, desde 1955





1954: IBM 704 e o FORTRAN

- ◇ FORTRAN 2 (1958)
 - Permitiu compilação separada de subprogramas
 - O IBM 704 tinha muitas falhas de hardware
 - programas de 300 a 400 linhas podiam ser compilados antes de uma falha, provavelmente
- ◇ FORTRANs 4, 77, 90, 95 e 2003
 - 2003: suporte a OO, ponteiros para procedimentos e interoperabilidade com C





1950: LISP

- ◇ Primeira linguagem de programação funcional
 - Aplicações de Inteligência Artificial, Processamento da Linguagem Natural e Prova de Teoremas
- ◇ Como processar dados simbólicos?
 - Processamento de listas!





1950: LISP

- ◇ LISP puro tem apenas duas estruturas de dados:
 - átomos
 - Listas

(A B C D)

(A (B C) D (E)) estrutura recursiva





1950: LISP

◇ Linguagem Funcional

- Todas as computações são realizadas por meio de aplicação de funções a argumentos
- Não há necessidade de atribuições e variáveis
- Ao invés de repetições, tem-se recursões





1950: LISP

- Exemplo de programa

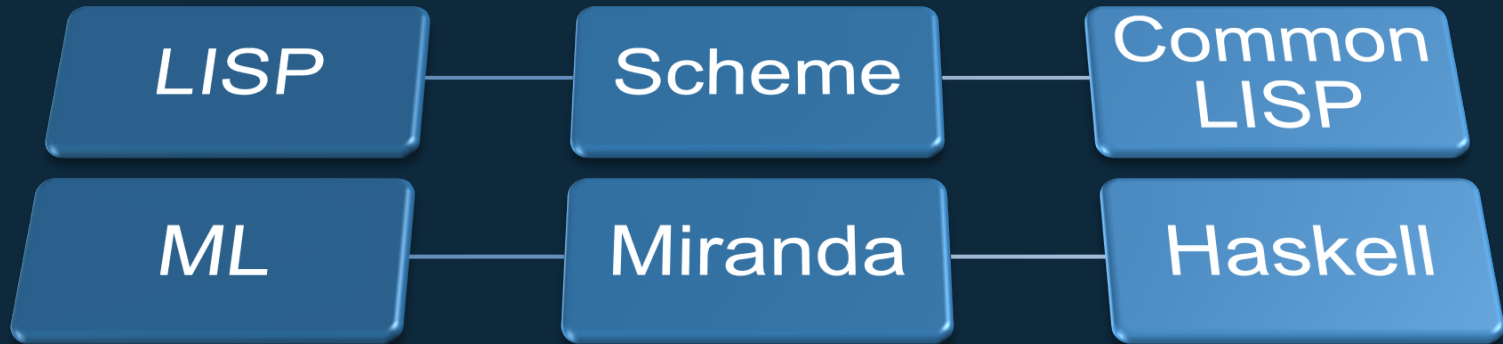
```
(DEFUN op_lists (lis1 lis2)
  (COND
    ((ATOM lis1) (EQ lis1 lis2))
    ((ATOM lis2) NIL)
    ((op_lists (CAR lis1) (car lis2))
     (op_lists (CDR lis1) (CDR lis2)))
    (T NIL)
  )
)
```

O que faz esse programa?





LINGUAGENS FUNCIONAIS





1959: ALGOL 60

- ◇ Influenciou LPs subsequentes
- ◇ Linguagem universal para aplicações científicas
 - Em 1957 muitas linguagens haviam sido desenvolvidas (FORTRAN, UNIVAC, MATHEMATIC e UNICODE).
- ◇ Diversos usuários pediram à ACM (*Association for Computation Machinery*) para formar um comitê para criar uma linguagem universal
 - FORTRAN era de propriedade da IBM





1959: ALGOL 60

- ◇ Projeto inicial
 - Sintaxe o mais próxima da notação matemática padrão
 - Possibilidade de utilizar a linguagem para descrição de algoritmos em publicações
 - Devem ser mecanicamente traduzíveis para código de máquina
- ◇ ALGO r ithmic Language
- ◇ Backus apresentou o BNF (*Backus-Naur Form*)





1959: ALGOL 60

◇ Novidades

- Conceito de estrutura de blocos
- Passagem de parâmetro por valor e referência
- Procedimentos podiam ser recursivos
- Vetores dinâmicos de pilha
- Dentre outros





1960: COBOL

- ◇ Uma das linguagens de programação mais usadas na história
 - Final dos anos 90, em um estudo do problema do ano 2000, constatou-se mais de 800 mi de linhas de código COBOL para 35 km² de Manhattan
- ◇ Voltada para aplicações comerciais
- ◇ Baseada no FLOW-MATIC





1960: COBOL

- ◇ Conceitos inovadores
 - DEFINE
 - Registros
 - Nomes de variáveis longos: até 30 caracteres e hífens
- ◇ Foi obrigatório seu uso no Departamento de Defesa dos EUA
- ◇ Levou à mecanização da contabilidade





1964: BASIC

- ◇ Como o COBOL, foi ignorada pelos cientistas da computação
- ◇ Fácil de aprender e pode ser implementado em computadores com pouca memória
- ◇ Em 1991 revitalizou-se seu uso pelo surgimento do Visual Basic (Microsoft)





1965: PL/I

- ◇ Primeira tentativa de uma linguagem de uso em qualquer área de aplicação
 - Antes focava-se em aplicações científicas, IA, e negócios
- ◇ Desenvolvida como produto da IBM
- ◇ Hoje é uma linguagem quase sem uso





1965: PL/I

- ◇ Primeira linguagem de programação a incluir
 - Concorrência de subprogramas
 - 23 tipos de exceções e erros em tempo de execução
 - Recursão em subprogramas
 - Ponteiro como tipo de dados
 - Referência a posições de matrizes





1960: APL e SNOBOL

- ◇ Tinham sintaxes bem diferentes
- ◇ Tipagem dinâmica e alocação dinâmica
- ◇ Ainda em uso





1967: SIMULA 67

- SIMULA I (1962) para realização de simulações no computador
- SIMULA 67, baseado no ALGOL 60
- Introduziu corrotinas
- Criou o conceito de "classe"
 - Avançando a abstração de dados





1969: ALGOL 68

- ◇ Bem diferente do ALGOL anterior
- ◇ Uso da **ortogonalidade**
 - Tipos de dados definidos pelo usuário
- ◇ Vetores dinâmicos, alocação em tempo de execução





1971: Pascal

- ◇ Todas as linguagens imperativas e OO devem algo ao projeto do ALGOL 60 e 68
- ◇ Muito utilizado no aprendizado de programação
 - Foi projetada para o ensino
 - Perdeu essa popularidade em 1990
- ◇ Não atendia muitas aplicações reais
 - Impossibilidade de subprograma que recebe vetor dinâmico como parâmetro
- ◇ Simplicidade e expressividade





1972: C

- ◇ Linguagem fortemente tipada, baseada na B
 - 1989: ANSI: C89
 - 1999: ISO: C99
- ◇ Pode ser utilizado em muitas áreas de aplicação
 - Estruturas de dados
- ◇ Popularidade devida a um compilador muito usado e livre, no UNIX, disponível em muitos tipos de computadores





1972: Prolog

- ◇ Programação não procedural, baseada em lógica formal para indicar ao computador o que fazer
- ◇ Definido em duas iniciativas
 - Universidade de Aix-Marseille
 - Universidade de Edimburgo
- ◇ Linguagem interpretada





1972: Prolog

- ◇ Em 1980 acreditou-se (um grupo de cientistas da computação) que a programação lógica era a melhor forma de fugir da complexidade das LPs imperativas.
- ◇ Hoje, os programas imperativos são mais eficientes que os lógicos equivalentes
- ◇ Programação lógica é efetiva para algumas aplicações: SGBDs e problemas de IA





19: ADA

- ◇ Mais extenso e caro projeto de uma linguagem de programação da história.
- ◇ Desenvolvida pelo Departamento de Defesa dos EUA
- ◇ 1974 metade das aplicações eram sistemas embarcados
 - Custos com software crescentes
 - Mais de 450 LPs em uso
 - Quase nenhuma reutilização





1983: ADA

- ◇ Vários grupos de trabalhos envolvendo Exército, Marinha e Aeronáutica
 - Desde 1975
 - Em 1979 um projeto foi escolhido
 - Ada: primeiro programadora do mundo
- ◇ Ada 83 e Ada 95





1983: ADA

◇ Contribuições

- Uso de pacotes para encapsular objetos de dados, procedimentos, etc.
- Tratamento de exceções
- Procedimentos genéricos
- Concorrência na execução de unidades de programas
 - Tarefas





1980: SmallTalk

- ◇ Conceitos da tese de Alan Kay, de 1969.
 - Vislumbrava que os computadores de mesa se popularizariam e seriam necessários softwares acessíveis a não programadores
 - Na Xerox, desenvolveu o SmallTalk-80





1980: SmallTalk

- ◇ Tudo é objeto
- ◇ Técnica uniforme:
 - Mensagens enviadas para objetos, para invocar seus métodos
- ◇ Contribuições
 - Interfaces gráficas com usuário (Sistema de Janelas)
 - Orientação a Objetos





1985: C++

- ◇ Tem diversos recursos originados no SIMULA 67 sobre a linguagem C para oferecer suporte a Orientação a Objetos
- ◇ Construído a partir do C para melhorar seus recursos e suportar OO





1985: C++

- ◇ Herança múltipla
- ◇ Sobrecarga de operadores e métodos
- ◇ Vinculação dinâmica de métodos
- ◇ Tratamento de exceções
- ◇ Popularidade
 - Compiladores bons e baratos
 - Compatibilidade total com C
 - Apareceu no momento certo: interesse por OO





2000: Delphi

- ◇ Linguagem híbrida, similar ao C++
 - Adição de OO ao Pascal
- ◇ Linguagem menos complexa que C++
- ◇ Fornece GUI (*Graphical User Interface*)





1990: Java

- ◇ Projetistas iniciaram do C++
- ◇ Remoção de algumas construções
- ◇ Adição de outras
- ◇ Linguagem menor e mais segura que C++





1990: Java

- ◇ Sun Microsystems: LP para dispositivos eletrônicos embarcados
 - Confiabilidade era essencial
 - Não tem ponteiros como o C++, mas tipos de referência (qual a diferença)?
 - Imagine o custo de um *recall* de televisores...
- ◇ Em 1993, na popularização da WWW, Java foi uma boa opção: *applets* Java





1990: Java

- ◇ Não suporte herança múltipla
- ◇ Controle de concorrência simples (*Synchronized*)
- ◇ Liberação implícita de armazenamento (*Garbage Collection*)
- ◇ Coerção de tipos em atribuições apenas por alargamento (*int* para *float* e não de *float* para *int*)
- ◇ Portabilidade através da JVM (*Java Virtual Machine*)
- ◇ Aumento da performance pela compilação JIT (*Just-in-Time*)





1987: Perl

- ◇ Combinação de *sh* e *awk* (linguagens de script muito usadas em sistemas UNIX)
- ◇ Linguagem imperativa de uso geral
 - Facilidade na manipulação de strings
 - Biologia computacional
 - Linguística computacional





1990: Lua

- ◇ Participação de um brasileiro (Luis Henrique de Figueiredo – PUC-Rio) em seu projeto
- ◇ Linguagem pequena e simples (21 palavras reservadas)
- ◇ Filosofia: fornecer apenas o necessário e maneiras simples de estender a linguagem para aplicações específicas





1991: Python

- ◇ Prioriza a legibilidade do código sobre a velocidade ou expressividade
 - Sintaxe concisa
 - Disponibilidade de muitas bibliotecas pela comunidade
- ◇ Multi-paradigmas
 - Imperativo, OO
 - Suporte a programação funcional





1993: JavaScript

- ◇ Necessidade de computação associadas à visualização das informações (HTML)
- ◇ Alternativas:
 - Era possível o uso de CGI (*Common Gateway Interface*)
 - Java applets
- ◇ Linguagens de script tornaram-se preferenciais





1993: JavaScript

- ◇ Desenvolvido no Netscape
 - Chamava-se Mocha
 - Depois LiveScript
 - Depois, em conjunto com Sun Microsystems, chamou-se JavaScript
- ◇ Geralmente, é embarcado em documentos HTML e interpretado pelo navegador
 - Modificação dinâmica de conteúdos HTML





1994: PHP

- ◇ Desenvolvida com o objetivo de rastrear visitantes em um site pessoal
 - Chamava-se inicialmente *Personal Home Page*
 - A comunidade mudou para *Hypertext Preprocessor*
- ◇ Muitas similaridades com o JavaScript, mas é processado no lado do servidor





1996: Ruby

- ◇ LP orientada a objetos pura
 - Python e Perl não são
- ◇ Projetada no Japão e usada intensamente nos EUA





2002: C#

- ◇ Lançada pela Microsoft junto com a plataforma .NET
- ◇ Baseada em
 - C++ e Java
 - Delphi e Visual Basic
- ◇ Aprimoramento do C++ e Java
- ◇ Código compilado para linguagem intermediária (*CIL* – *Common Intermediate Language*), interpretado pela CLR (*Common Language Runtime*)





1999: JSP

- ◇ *Java Server Pages*
 - Coleção de tecnologias projetadas para oferecer suporte a documentos Web dinâmicos
- ◇ Linguagem Híbrida de marcação/programação
- ◇ Necessita de um *servlet* (instância de uma classe Java que é executada em um servidor Web)





Frameworks para desenvolvimento web

◇ Back-end

- Express (Node.js)
- Django (Python)
- Rails (Ruby)
- Laravel (PHP)
- Spring (Java)



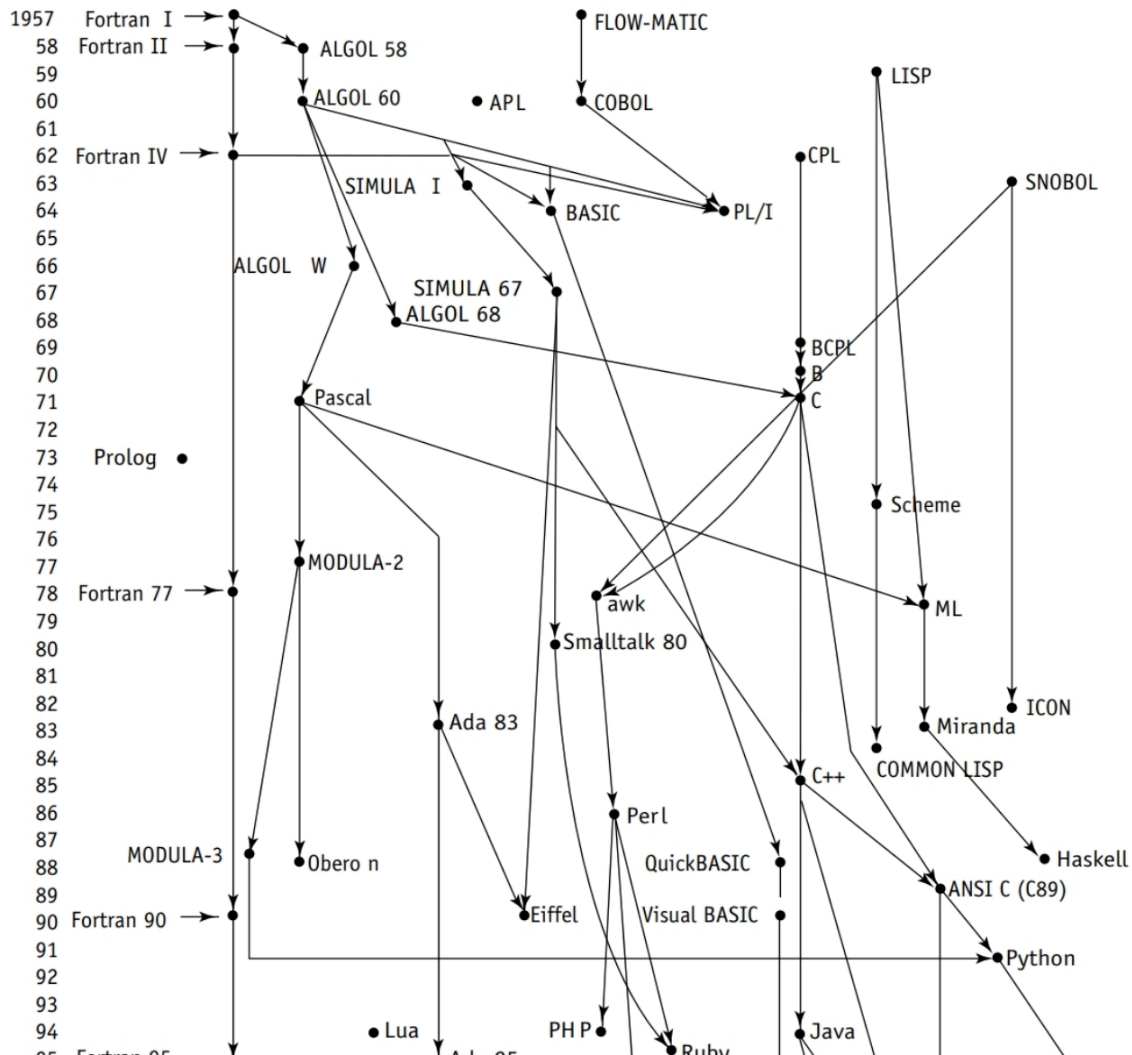


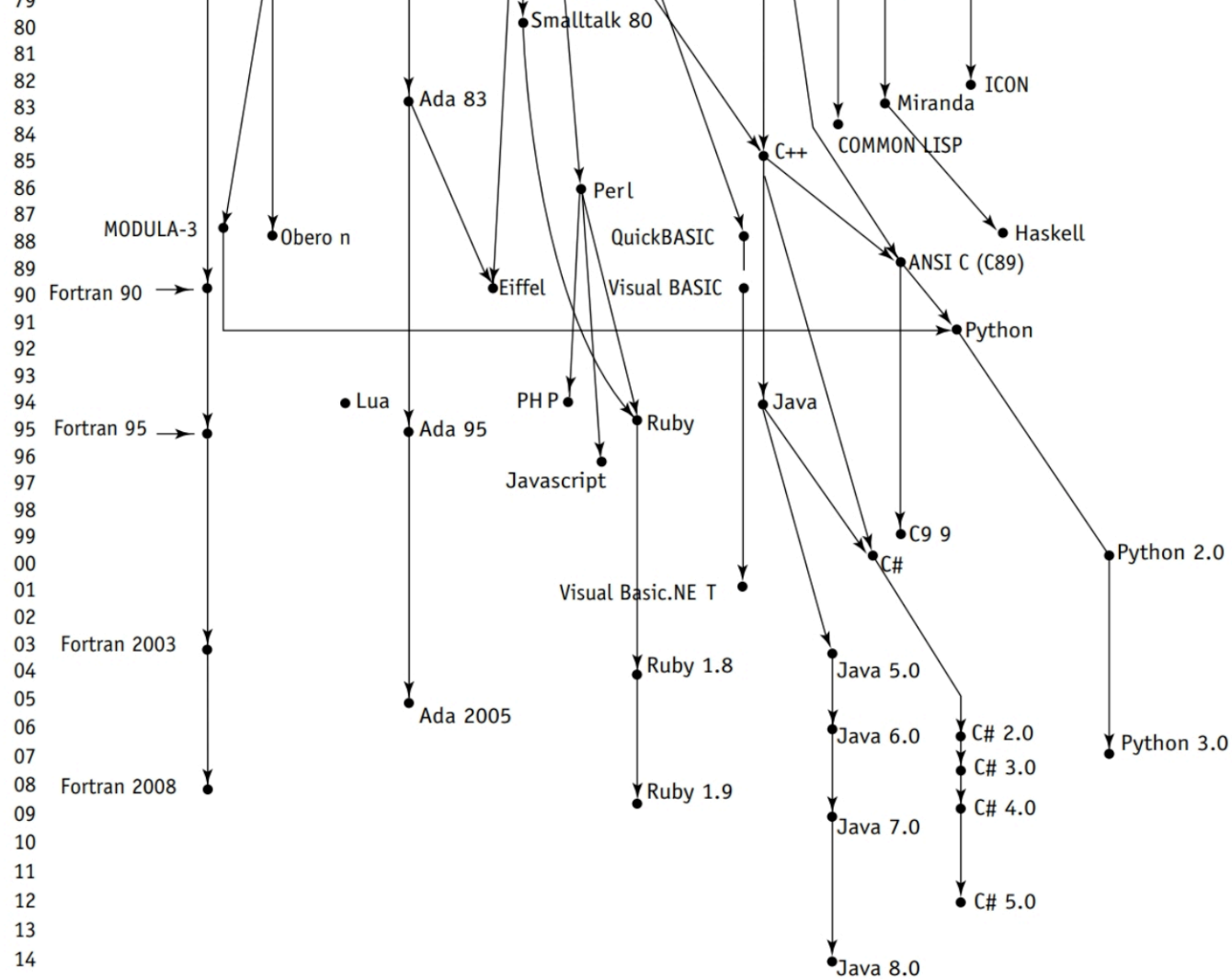
Frameworks para desenvolvimento web

◇ Front-end

- Angular (JavaScript, Typescript)
- React
- Vue.js
- Ember
- Backbone









Referência Bibliográfica

Sebesta, R. W. (2011). *Conceitos de Linguagens de Programação*. 9 ed. Bookman.

Capítulo 2

