

**Pró-Reitoria Acadêmica  
Curso de Engenharia de Software  
Trabalho de Algoritmos e Programação Estruturada**

***O Contexto Histórico e Científico  
na Criação do Computador***

**Autores: Yuri Guerra  
Gabriel da Silva Pedrosa  
Wemerson Conceição da Silva  
Pedro Braga de Lima  
Iori Abreu Castro  
Maikon Douglas Pereira da Silva**

**Orientador: Jefferson Salomão Rodrigues**

**YURI GUERRA  
IORI ABREU CASTRO  
PEDRO BRAGA DE LIMA  
WEMERSON CONCEIÇÃO SILVA  
GABRIEL DA SILVA PEDROSA  
MAIKON DOUGLAS PEREIRA DA SILVA**

## **O CONTEXTO HISTÓRICO E CIENTÍFICO NA CRIAÇÃO DO COMPUTADOR**

Documento apresentado ao Curso de graduação de Bacharelado em Engenharia de Software da Universidade Católica de Brasília, como requisito parcial para obtenção da aprovação na disciplina de Algoritmos e Programação Estruturada.

Orientador: Jefferson Salomão Rodrigues

**Brasília - DF**

**2024**



Artigo de autoria de Guerra, Yuri; Abreu Castro, Iori; Douglas Pereira da Silva, Maikon; Da Silva Pedrosa, Gabriel; Braga de Lima, Pedro; Conceição Silva, Wemerson, intitulado "O CONTEXTO HISTÓRICO E CIENTÍFICO NA CRIAÇÃO DO COMPUTADOR", apresentado como requisito parcial para obtenção da aprovação nas disciplinas de Algoritmos e Programação Estruturada do Curso de Bacharelado em Engenharia de Software, da Universidade Católica de Brasília, defendido e aprovado pela banca examinadora abaixo assinada:

---

Prof. Jefferson Salomão Rodrigues  
Algoritmos e Programação Estruturada

**Brasília - DF  
2024**

## **RESUMO**

Referências: Guerra, Yuri; Abreu Castro, Iori; Douglas Pereira da Silva, Maikon; Da Silva Pedrosa, Gabriel; Braga de Lima, Pedro; Conceição Silva, Wemerson, intitulado “O CONTEXTO HISTÓRICO E CIENTÍFICO NA CRIAÇÃO DO COMPUTADOR”, 2024. Curso de Bacharelado em Engenharia de Software, da Universidade Católica de Brasília, Taguatinga-DF, 2024.

Ao analisar a evolução dos computadores modernos, com o objetivo de compreender o contexto histórico e científico da criação de tais máquinas, as associando com os desafios propostos em sala e destrinchando os acontecimentos somados aos grandes nomes da história que influenciaram nesta evolução. O método utilizado é baseado em pesquisas bibliográficas e análises de documentos, os quais estarão listados ao final do artigo. O objetivo da metodologia é focar nas limitações do software e hardware atual, bem como nas técnicas de criptografia e decodificação. Os resultados apontam na compreensão da contextualização histórica da evolução gradual dos computadores em paralelo com os desafios supracitados. Conclui-se que necessidades práticas, como conflitos bélicos, elevaram o impulsionamento do avanço da computação. Outrossim, as tendências futuras da computação e os avanços tecnológicos continuam moldando a sociedade.

Palavras-chave: Evolução dos computadores; limitações de software; limitações de hardware; avanço da computação.

## ***ABSTRACT***

The emergence of the computer as we know it today is linked to the context of the Second World War, when there was a need to perform precise ballistic calculations. However, the path to the creation of this technology began much earlier, around 2500 BC, with the invention of the abacus, a tool designed to perform simple arithmetic calculations. Over the centuries, other devices, such as the astrolabe, slide rules and mechanical clocks, contributed to the evolution of calculation tools. However, it was only with the creation of the Pascalina, one of the first mechanical calculators invented by Blaise Pascal in the 17th century, that the foundations for future computing machines emerged.

In the 19th century, the British mathematician Charles Babbage conceived what would be the first concept of a programmable computer, with the contribution of Ada Lovelace, considered the first programmer. However, it was in the 20th century that electronic computing really took shape, with the fundamental contribution of Alan Turing, whose ideas made it possible to create universal programmable machines. In 1946, with the development of ENIAC, the first programmable electronic computer, the era of modern computing was officially initiated, forever transforming the world of science and technology.

Keywords: Evolution of computers; software limitations; hardware limitations; advancement of computing.

## ***LISTA DE FIGURAS***

	Pág.
Figura 1 - Cronograma do Desenvolvimento dos Códigos.....	26
Figura 2 - Cronograma de Desenvolvimento do Artigo. ....	26

## SUMÁRIO

RESUMO	4
ABSTRACT	5
LISTA DE FIGURAS	6
1 INTRODUÇÃO	7
2 MOTIVAÇÃO	9
DISCUSSÃO	9
3 DIAGNÓSTICO DA ANÁLISE	10
4 OBJETIVOS	12
4.1 OBJETIVOS GERAIS	12
4.2 OBJETIVOS ESPECÍFICOS	12
5 DESENVOLVIMENTO	13
6 METODOLOGIA	15
6.1 TIPO DE PESQUISA	15
6.3 PROCEDIMENTO	15
7 RESULTADO	17
8 CONCLUSÃO	22
9 REFERÊNCIAS	23
10 GLOSSÁRIO	24
11 ANEXO 01 – CRONOGRAMAS	25
12 ANEXO 02 – LINK DO REPOSITÓRIO GITHUB	26

### **1 INTRODUÇÃO**

O surgimento do computador, como conhecemos hoje, está ligado ao contexto da Segunda Guerra Mundial, quando havia a necessidade de realizar cálculos balísticos precisos. No entanto, o caminho para a criação dessa tecnologia começou muito antes, por volta de 2500 a.C., com a invenção do ábaco, uma ferramenta feita para realizar cálculos aritméticos simples. Ao longo dos séculos, outros dispositivos, como o astrolábio, réguas de cálculo e relógios mecânicos, contribuíram para a evolução das ferramentas de cálculo. No entanto, foi somente com a criação da Pascalina, uma das primeiras calculadoras mecânicas inventada por Blaise Pascal no século XVII, que surgiram as bases para as futuras máquinas computacionais.

No século XIX, o matemático britânico Charles Babbage concebeu o que seria o primeiro conceito de um computador programável, com a contribuição de Ada Lovelace, considerada a primeira programadora. Contudo, foi no século XX que a computação eletrônica realmente tomou forma, com a contribuição fundamental de Alan Turing, cujas ideias possibilitaram a criação de máquinas programáveis universais. Em 1946, com o desenvolvimento do ENIAC, o primeiro computador eletrônico programável, a era da computação moderna foi oficialmente iniciada, transformando para sempre o mundo da ciência e da tecnologia.



## **2 MOTIVAÇÃO**

### Discussão

A análise dos dados coletados oferece uma compreensão mais profunda sobre como a computação evoluiu ao longo do tempo. O desenvolvimento do computador foi impulsionado por necessidades práticas, como as operações militares da Segunda Guerra Mundial, e consolidado por avanços científicos.

### **3 DIAGNÓSTICO DA ANÁLISE**

#### **3.1 Análise dos Resultados e Comparação com a Segunda Guerra Mundial**

As análises mostram como as limitações tecnológicas da época exigiam soluções criativas para decodificar mensagens com rapidez e precisão. Isso pode ser comparado ao famoso caso da quebra da máquina Enigma pelos britânicos, que combinaram técnicas matemáticas com inovação tecnológica para superar os desafios de criptografia.

Da mesma forma, as dificuldades enfrentadas nos exercícios propostos revelam os desafios encontrados pelos primeiros computadores para processar grandes quantidades de dados. As técnicas de decodificação desenvolvidas durante a guerra, como o uso de padrões numéricos e a repetição de mensagens, podem ser vistas nas abordagens utilizadas para otimizar os processos computacionais da época.

#### **3.2 Conclusão da Análise**

Os dados mostram que o desenvolvimento do computador moderno foi diretamente influenciado pelo contexto da Segunda Guerra Mundial, mas suas raízes remontam a séculos anteriores de evolução tecnológica. A análise documental e bibliográfica destacou que a computação se desenvolveu de forma incremental, com avanços que revolucionaram a ciência, a indústria e a sociedade.

As técnicas desenvolvidas na época, particularmente em relação à automação de cálculos e à criação de algoritmos programáveis, formam a base da computação moderna.

#### **3.3 Implicações Futuras**

Os resultados da análise histórica sugerem que a computação está em constante evolução. Desde cálculos simples até algoritmos complexos e inteligência

artificial, o futuro da computação aponta para inovações como os computadores quânticos e a inteligência artificial avançada. Essas tecnologias têm o potencial de revolucionar a forma como interagimos com a tecnologia, moldando o futuro da ciência e da sociedade

## **4 OBJETIVOS**

### **4.1 Objetivos gerais**

Compreender o contexto histórico e científico que possibilitou a criação do computador moderno.

### **4.2 Objetivos específicos**

Entender como os cálculos computacionais contribuíram na codificação e decodificação de mensagens durante a guerra.

Compreender as limitações do computador no sentido de sistema de numeração posicional e limites de variáveis.

Demonstrar de que forma os exercícios propostos nos auxiliam a compreender como eram feitas as decodificações na época.

## **5 DESENVOLVIMENTO**

### **SEÇÃO 1: O CONTEXTO HISTÓRICO E CIENTÍFICO NA CRIAÇÃO DO COMPUTADOR**

#### **Contexto Histórico**

A análise bibliográfica revela que as bases para o desenvolvimento do computador moderno remontam ao século XVII, com a criação da Pascalina, a primeira calculadora mecânica funcional. A contribuição de Charles Babbage e Ada Lovelace no século XIX também foi crucial para o desenvolvimento dos conceitos de computação programável.

Contudo, foi no século XX, com o contexto da Segunda Guerra Mundial, que ocorreram avanços decisivos. A necessidade de realizar cálculos rápidos e precisos, particularmente no campo militar, impulsionou o desenvolvimento de máquinas como o ENIAC (Electronic Numerical Integrator and Computer), em 1946, considerado o marco inicial da era da computação moderna. O ENIAC foi essencial para cálculos balísticos e previsões meteorológicas, demonstrando a relevância prática da computação no contexto militar e científico.

O estudo também destacou a influência de Alan Turing, que, com sua Máquina de Turing Universal, idealizou o conceito de uma máquina programável capaz de executar qualquer algoritmo. Isso abriu caminho para a programação moderna e consolidou as bases teóricas da ciência da computação.

#### **Evolução Científica e Tecnológica**

Os dados históricos demonstram que a evolução da computação ocorreu em etapas, cada uma marcada por inovações tecnológicas que aumentaram a capacidade de processamento das máquinas. Durante a Segunda Guerra Mundial, máquinas como o Colossus foram desenvolvidas para decifrar códigos secretos, marcando um ponto chave na história da criptografia e da computação.

A introdução dos transistores na década de 1950 e o desenvolvimento dos circuitos integrados na década de 1960 revolucionaram ainda mais a computação, tornando os computadores menores, mais rápidos e acessíveis. A análise documental de algoritmos e técnicas usados durante esse período mostrou como conceitos de automação e aritmética digital foram integrados em operações militares e científicas, exemplificados pelos cálculos balísticos realizados pelo ENIAC.

*-- como os primeiros computadores enfrentaram limitações de hardware e software, particularmente em relação à manipulação de dados numéricos e ao sistema de codificação.*

## **6 METODOLOGIA**

### **6.1 TIPO DE PESQUISA**

Esse artigo visa entender e compreender o contexto histórico e científico que o computador foi criado usando algoritmos modernos para entender como as técnicas históricas se aplicam na tecnologia moderna. A pesquisa é descritiva e exploratória visando compreender a evolução das tecnologias e técnicas computacionais durante a Segunda Guerra Mundial e suas implicações.

### **6.2 COLETA DE DADOS**

Com o objetivo de aprofundar os conhecimentos sobre o contexto histórico e científico que possibilitou a criação do computador moderno, esse artigo terá como base pesquisas bibliográficas sobre o assunto. E como forma de entender e compreender de que forma o computador e os cálculos matemáticos eram utilizados na época de sua criação, será feita uma análise documental com base nos exercícios propostos e descritos nesse artigo.

### **6.3 PROCEDIMENTO**

Inicialmente, será elaborado um plano de pesquisa detalhado, definindo de que forma os dados serão captados e como será feita a análise documental com base nos exercícios propostos.

Após a definição de fontes de procura será feita uma análise detalhada de pesquisas relevantes sobre o tema sendo levado em conta a veracidade e importância para o estudo. Assim que análise for feita e a pesquisa realizada, o próximo passo será estruturar o artigo com base nos dados coletados.

Para proporcionar familiaridade com o tema de forma a descrever os fatos e fenômenos dessa realidade serão realizados três exercícios, cada um visando o entendimento do cenário estruturado. O primeiro exercício será feito e analisado para explorar e entender como os primeiros computadores enfrentaram limitações de hardware e software, particularmente em relação à manipulação de dados numéricos e ao sistema de codificação.

O segundo exercício de invalidar dados será realizado e analisado para melhor compreensão sobre contexto histórico referente a decodificação de mensagens durante a Segunda Guerra Mundial que envolvia algoritmos que manipulavam grandes quantidades de dados, incluindo caracteres alfanuméricos e de tal forma que era comum remover ou invalidar certos caracteres ou informações desnecessárias para otimizar o processo de criptografia/decodificação.

E o terceiro exercício para análise dos resultados e dados apresentados visa traçar um paralelo direto com os processos de criptografia e decodificação de mensagens que foram fundamentais durante a Segunda Guerra Mundial.



## 7 RESULTADO

### 7.1 Apresentação dos Dados Coletados e Análises Realizadas

Nesta seção, serão apresentados os dados coletados a partir da pesquisa bibliográfica e da análise documental, seguidos de uma interpretação dos resultados em relação ao contexto histórico e científico que possibilitou o surgimento do computador moderno e a Análise dos limites suportados pelos tipos de arquivos. A metodologia adotada permitiu uma análise crítica sobre como as técnicas computacionais desenvolvidas no passado influenciaram diretamente as tecnologias modernas.

### 7.2 Análise dos limites suportados pelos tipos de arquivos

Os limites numéricos que cada tipo inteiro está definido na biblioteca da linguagem LIMITS.H, na forma de constantes, de acordo com a capacidade de cada variável foram utilizadas as variáveis Char com a capacidade de 1 byte, valor mínimo de -128 e valor máximo de 127. Unsigned char 1 byte, valor mínimo de 0 e valor máximo de 255. Short int 2 bytes, valor mínimo -32768 e 32767. Unsigned short int 2 bytes, valor mínimo 0 e valor máximo 65535. Int 4 bytes, valor mínimo de -2147483648 e valor máximo de 2147483647. Unsigned int 4 bytes, valor mínimo de 0 e valor máximo de 4294967295. Long int 4 ou 8 bytes valor mínimo de -2147483648 e máximo de 2147483647. Unsigned long int 4 ou 8 bytes valor mínimo de 0 e máximo de 4294967295. Long long int 8 bytes valor mínimo de -9223372036854775808 e valor máximo de 9223372036854775807. Unsigned long long int 8 bytes, valor mínimo de 0 e valor máximo de 18446744073709551615.

Obs.: long int = 4 bytes (32 bits) ou 8 bytes (64 bits); Unsigned long int = 4 bytes (32 bits) ou 8 bytes (64 bits).

o que acontece se você utilizar um valor fora dos limites dos tipos de variáveis.

Overflow e Underflow:

Overflow ocorre quando um valor excede o limite máximo que uma variável pode

armazenar.

Underflow ocorre quando o valor é menor que o limite mínimo que a variável pode armazenar.

O que pode acontecer:

Linguagens com comportamento de rotação (como C e C++):

Em linguagens como C e C++, o overflow e underflow podem resultar em “comportamento de rotação” (wrap-around). Isso significa que, quando o valor excede o limite máximo, ele “volta” ao limite mínimo e vice-versa. Isso pode causar erros difíceis de detectar.

Perda de precisão:

Para variáveis de ponto flutuante (como float e double), valores fora dos limites podem resultar em infinito positivo ou negativo (Infinity, -Infinity), ou em NaN (Not a Number). Além disso, operações com valores muito pequenos podem resultar em perda de precisão devido ao underflow.

Erro de compilação:

Em algumas linguagens mais seguras ou em configurações mais restritivas, tentar atribuir um valor fora do limite de uma variável pode resultar em um erro de compilação.

Exemplo: C/C++: Se você declarar um unsigned int (sem sinal) e atribuir um valor negativo, isso resultará em um overflow, onde o número será convertido para um grande número positivo.

Esses cenários ilustram a importância de conhecer os limites de cada tipo de dado em uma linguagem.

### Função que invalida alguns caracteres

Uma função em C que invalida caracteres geralmente percorre uma string e altera ou remove certos caracteres indesejados. Isso pode ser feito de diferentes maneiras, dependendo de como você define "invalidação" que no caso ficou definida por “0”. Aqui estão algumas abordagens comuns:

Substituir caracteres inválidos por outro caractere

Um exemplo clássico é substituir caracteres inválidos (como caracteres não alfabéticos) por um caractere específico, como “0” (nulo).

Remover caracteres inválidos (compactação da string)

Outra forma de invalidar caracteres é removê-los completamente da string, "compactando" a string para que os caracteres válidos permaneçam.

### **7. 3 Explicação Geral de Como Funciona:**

1. Percorrer a string: A função percorre a string usando um loop que verifica cada caractere até encontrar o caractere nulo “0”, que marca o fim da string.

2. Verificar caracteres: Para verificar se um caractere é válido ou inválido, usamos funções auxiliares da biblioteca <ctype.h>, como isalpha (para letras) ou isdigit (para números).

3. Substituir ou remover: Dependendo do que você deseja fazer com os caracteres inválidos, você pode:

- Substituí-los por outro caractere (como '0').
- Compactar a string, movendo os caracteres válidos para as primeiras posições e eliminando os inválidos.

4. Modificar a string original: Como as strings em C são apenas arrays de caracteres, a modificação é feita diretamente na string original.

Essas abordagens são úteis em situações como validação de entrada de usuário, limpeza de dados, ou formatação de strings antes de processamento posterior.

Função principal

Neste algoritmo, recebemos uma cadeia de caracteres em formato hexadecimal, e o primeiro passo é separar essa cadeia em pares de caracteres (duplas). Em seguida, aplicamos uma função de invalidação de caracteres para filtrar ou remover valores inválidos, determinando quais devem ser traduzidos.

Após a filtragem, convertamos as duplas de caracteres hexadecimais para seus valores correspondentes em decimal. Esses valores decimais são então mapeados para seus respectivos caracteres da tabela ASCII \*CP437\*.

Finalmente, a sequência de caracteres resultante é utilizada para imprimir a imagem na tela.

#### 7.4 Limitações de Hardware

Os primeiros computadores, como o ENIAC e o Colossus, apresentavam severas limitações de hardware:

Capacidade de armazenamento: A memória disponível era extremamente limitada, forçando os engenheiros a simplificar algoritmos e dividir o processamento em pequenos blocos de dados.

Velocidade de processamento: Embora inovadores para a época, a velocidade de cálculo desses computadores era muito baixa em comparação com as máquinas modernas, o que impactava diretamente a eficiência em processos como a decodificação de mensagens criptografadas.

Precisão e erros: O uso de números muito grandes ou pequenos podia gerar erros, pois esses computadores tinham uma capacidade limitada para representar frações ou decimais.

Sistemas físicos delicados: Componentes como válvulas e relés frequentemente apresentavam falhas, afetando a confiabilidade e a continuidade do processamento.

Além das restrições de hardware, os primeiros computadores enfrentavam desafios relacionados ao software:

Programação manual: As instruções eram inseridas manualmente por meio de cartões perfurados, um processo demorado e propenso a erros.

Codificação rígida: Os algoritmos eram limitados a operações matemáticas básicas, complicando a execução de processos mais complexos, como a criptografia.

Falta de abstração: Não havia linguagens de alto nível ou compiladores. Todas as operações tinham que ser traduzidas diretamente em instruções binárias, tornando

a programação mais desafiadora.

*-- melhor compreensão sobre contexto histórico referente a decodificação de mensagens durante a Segunda Guerra Mundial que envolvia algoritmos que manipulavam grandes quantidades de dados, incluindo caracteres alfanuméricos e de tal forma que era comum remover ou invalidar certos caracteres ou informações desnecessárias para otimizar o processo de criptografia/decodificação.*

### 7.5 Manipulação de Dados Numéricos e Limitações Computacionais

Nos primeiros computadores, os algoritmos de criptografia e decodificação envolviam operações matemáticas complexas, como somas e multiplicações de grandes números, utilizando sistemas binários e decimais. As limitações de hardware e software obrigavam os operadores a lidar com restrições severas de memória e processamento.

A máquina Enigma, utilizada pelos alemães durante a guerra, utilizava padrões numéricos e alfanuméricos para criptografar mensagens. A decodificação dependia da capacidade de manipular grandes quantidades de dados de forma eficiente. As limitações tecnológicas tornavam esse processo mais lento, exigindo otimizações e avanços nos algoritmos de decodificação.

-- paralelo direto com os processos de criptografia e decodificação de mensagens que foram fundamentais durante a Segunda Guerra Mundial.

## 8 CONCLUSÃO

Em suma, o magnífico caminho das ferramentas simples até as complexas, percebe-se uma atitude humana movida por suas próprias necessidades incansáveis. Somado a isso, é adicionado ao computador moderno, um desenvolvimento que se deve muito mais do que meras necessidades, contudo uma profunda pesquisa científica em prol de alinhar necessidade e ciência indo além de gerações ou quaisquer fronteiras.

No princípio, as invenções mais brilhantes passavam a estabelecer os alicerces daquilo que conhecemos hoje como as atuais capacidades computacionais, que ainda sim vem sendo incrementadas em prol de evoluírem. Durante a Segunda Guerra, havia um senso de urgência que os levou a criação do precursor de tudo o que temos no âmbito computacional, o ENIAC, que fez com que fosse possível termos hoje diversas aplicações com nossos computadores, que vão de estratégia militar até previsões meteorológicas.

No entanto, os avanços vistos não são apenas reconhecidos do avanço dessas máquinas, sendo também a nossa própria evolução humana por trás da programação de cada código, ou seja, de cada inovação. As contribuições de Alan Turing, por exemplo, redefiniram o que as máquinas poderiam fazer, antecipando a era da computação universal que hoje facilita desde a comunicação global até a solução de problemas complexos em quase todos os campos imagináveis.

Logo, a história dos computadores se faz fascinante, sendo uma prova de superação e adaptação humana. Ademais, quando olhamos para o futuro com base no que temos visto, como a própria IA, somos constantemente lembrados que cada uma dessas inovações é, em síntese, um reflexo da sociedade. A jornada da computação, é uma jornada contínua, pois sempre aponta para inovações, evolução em tudo o que tange, tanto como as linhas de código quanto a própria sociedade.

## 9 REFERÊNCIAS

### ▪ Vídeo

**Eu TI Ensino.** Como surgiu o computador: Conheça a história da computação!!!  
Vídeo. YouTube. Disponível em:  
<https://www.youtube.com/watch?v=ilwewVTh1R4&t=450s>. Acesso em: 14 set. 2024.

### ▪ Livro:

ENIAC: The Triumph of the Computers. GOLDstine, Herman H. New York: Addison-Wesley, 1997.

### ▪ Artigo:

CERUZZI, Paul. Beyond the machine: visions of computing in the postwar era. **Technology and Culture**, Chicago, v. 36, n. 4, p. 806-847, out. 1995.

## 10 GLOSSÁRIO

- **Ábaco:** Primeira ferramenta conhecida para realizar cálculos aritméticos simples.
- **Pascalina:** Uma das primeiras calculadoras mecânicas, inventada por Blaise Pascal.
- **Máquina de Turing:** Modelo matemático de um computador universal, concebido por Alan Turing.
- **ENIAC:** Primeiro computador eletrônico programável, desenvolvido durante a Segunda Guerra Mundial.
- **Criptografia:** Ciência de transformar informações em um formato ilegível para pessoas não autorizadas, garantindo a segurança da comunicação.
- **Decodificação:** Processo inverso da criptografia, que consiste em restaurar a informação original a partir de um texto cifrado.
- **Algoritmo:** Conjunto de instruções bem definidas que descrevem um processo de cálculo para resolver um problema.
- **Hardware:** Componentes físicos de um computador, como a placa-mãe, processador, memória e dispositivos de entrada e saída.
- **Software:** Programas e aplicativos que executam as tarefas em um computador.
- **Bit:** Menor unidade de informação em um computador, podendo assumir os valores 0 ou 1.
- **Byte:** Unidade de medida de informação composta por 8 bits.
- **Sistema binário:** Sistema numérico que utiliza apenas dois dígitos (0 e 1) para representar todos os números.
- **Cálculo balístico:** Cálculos matemáticos utilizados para determinar a trajetória de um projétil.
- **Transistor:** Componente eletrônico que amplifica ou comuta sinais elétricos e elétricos.
- **Circuito integrado:** Pequeno chip de silício que contém milhares ou milhões de transistores.



## 11 ANEXO 01 – CRONOGRAMAS

**FIGURA 1: CRONOGRAMA DE DESENVOLVIMENTO DOS CÓDIGOS:]**

Parte código	Pessoas	Observações	Prazo
Atividade 3.2.1 - Escrever um programa que use a biblioteca LIMITS.H para imprimir os limites mínimo e máximo de variáveis de vários tipos, e apresente os resultados em tabela.	Wemerson		15/set
Atividade 3.2.1 - Escreva um programa que atribua a uma variável um valor maior que o permitido para seu tipo, demonstrando o erro que ocorre. Falar sobre overflow e lowflow.	Yuri	(Fazer após o Wemerson finalizar a parte dele)	19/set
Atividade 3.2.2 - Implementar a função func_val(int x, int b) conforme definida, usando round da biblioteca math.h para arredondar o resultado.	Maikon	(Realizar após a conclusão da atividade 3.2.1)	22/set
Atividade 3.2.3 - Implementar um código que leia o x e o b e a cadeia de texto e separe os caracteres da cadeia de texto em duplas.	Pedrosa		25/set
Atividade 3.2.3 - Testar o valor utilizando a função criada na atividade 3.2.2 e, se validado, converter cada dois caracteres em valores decimais e depois traduzir para tabela ASCII.	Iori	(Realizar após a conclusão da atividade 3.2.2)	25/set
Atividade Extra	Pedro		22/set

**FIGURA 2: CRONOGRAMA DE DESENVOLVIMENTO DO ARTIGO:**

Parte artigo	Pessoas	Observação	Prazo
Resumo	Pedrosa	Realizar após todo trabalho feito	25/set
Introdução	Maikon		15/set
Metodologia	Iori		20/set
Resultados/discussão	Pedro	Realizar após a introdução	20/set
Conclusão	Yuri	Realizar após os resultados e discussão	22/set
Formatar artigo, juntar as partes, colocar artigo e o apêndice	Wemerson		29/set

## **12 ANEXO 02 – LINK DO REPOSITÓRIO GITHUB**

**Acesso aos algoritmos deste trabalho em:**

<https://github.com/WemersonTech/TrabalhoN1>