



UNIVERSIDADE  
FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ

# Olá mundo

QXD0001 - Fundamentos de Programação

Prof. Bruno Góis Mateus ([brunomateus@ufc.br](mailto:brunomateus@ufc.br))

# Agenda

- Introdução
- Breve história da computação
- Como funciona um computador?
- O que é um algoritmo
- Introdução à linguagem Go
- Ambiente de desenvolvimento

# Introdução

**Por que aprender  
programação?**

# Introdução

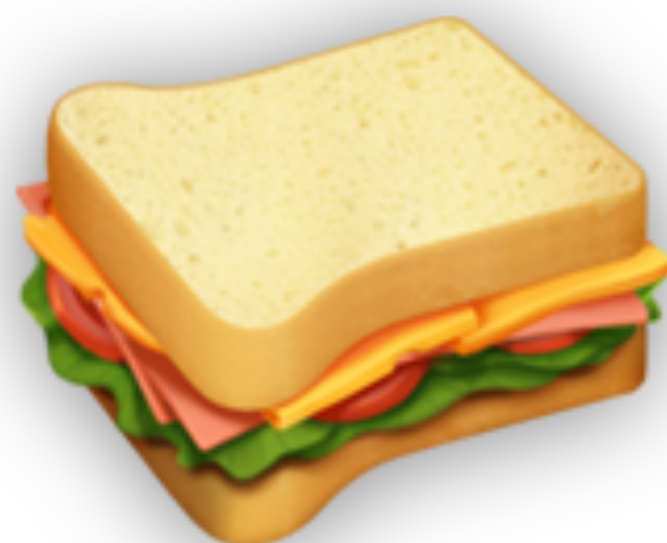
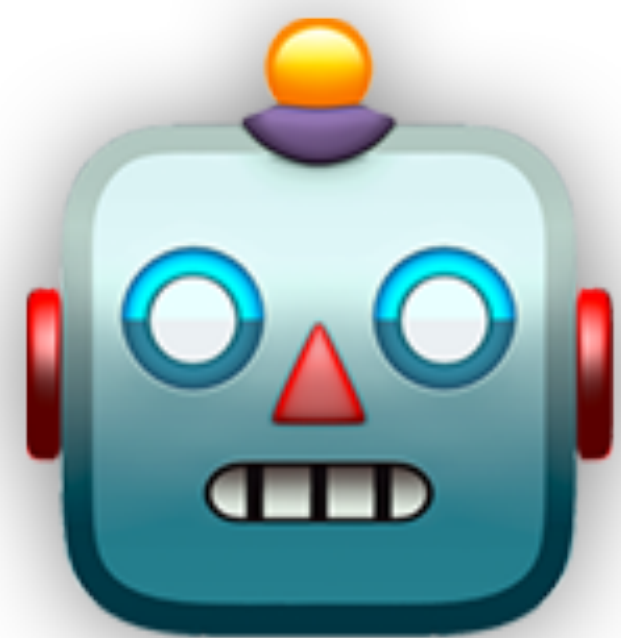
## Por que aprender programação?

- Automatizar tarefas
- Resolver problemas complexos
- Construir sistemas digitais
- Analisar dados
- Desenvolver novas tecnologias

# Introdução

## Programação está em todo lugar

- Exemplos de sistemas que dependem de software:
  - Aplicativos de transporte
  - Redes sociais
  - Sistemas hospitalares
  - Sistemas financeiros
  - Satélites
  - Carros modernos



Imagine que você  
precisa dar  
instruções a um robô  
para fazer um  
sanduíche.  
Quais seriam os  
passos?

# Introdução

## Problemas comuns nas instruções

- Quando descrevemos um processo:
  - Esquecemos passos
  - Usamos instruções ambíguas
  - Assumimos conhecimento implícito

# Introdução

## Conclusão

- Computadores são extremamente literais
  - Eles executam exatamente o que foi programado
  - Nem mais. Nem menos
- Por isso precisamos de instruções claras e precisas
- Essas instruções são chamadas de algoritmos

# Breve história da computação

# Breve história da computação

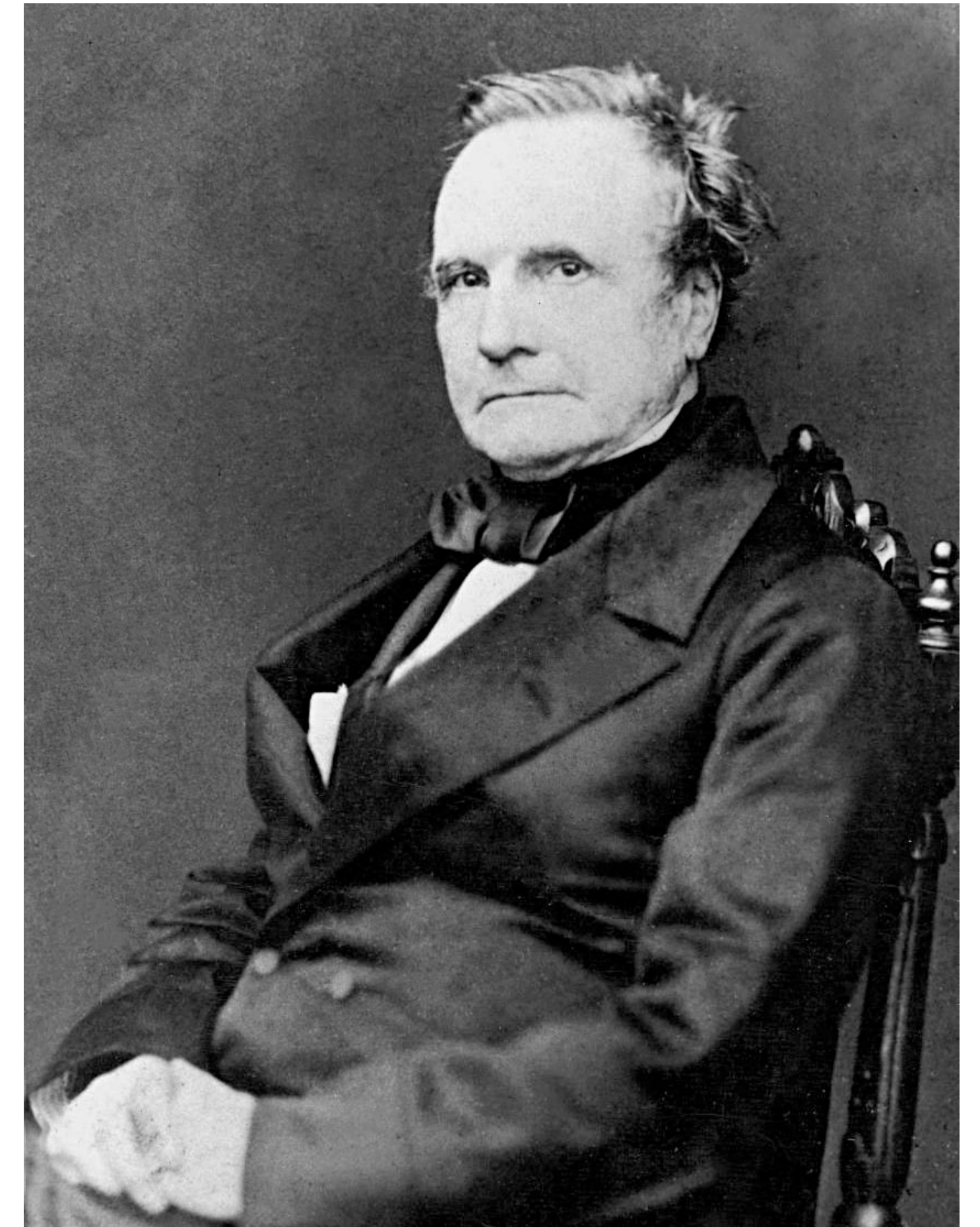
## Um pouco de história

- Antes dos computadores modernos, vários cientistas contribuíram para o desenvolvimento da computação, dentre eles destacamos:
  - Charles Babbage
  - Ada Lovelace
  - Herman Hollerith
  - Alan Turing
  - John von Neumann
  - Grace Hopper

# Breve história da computação

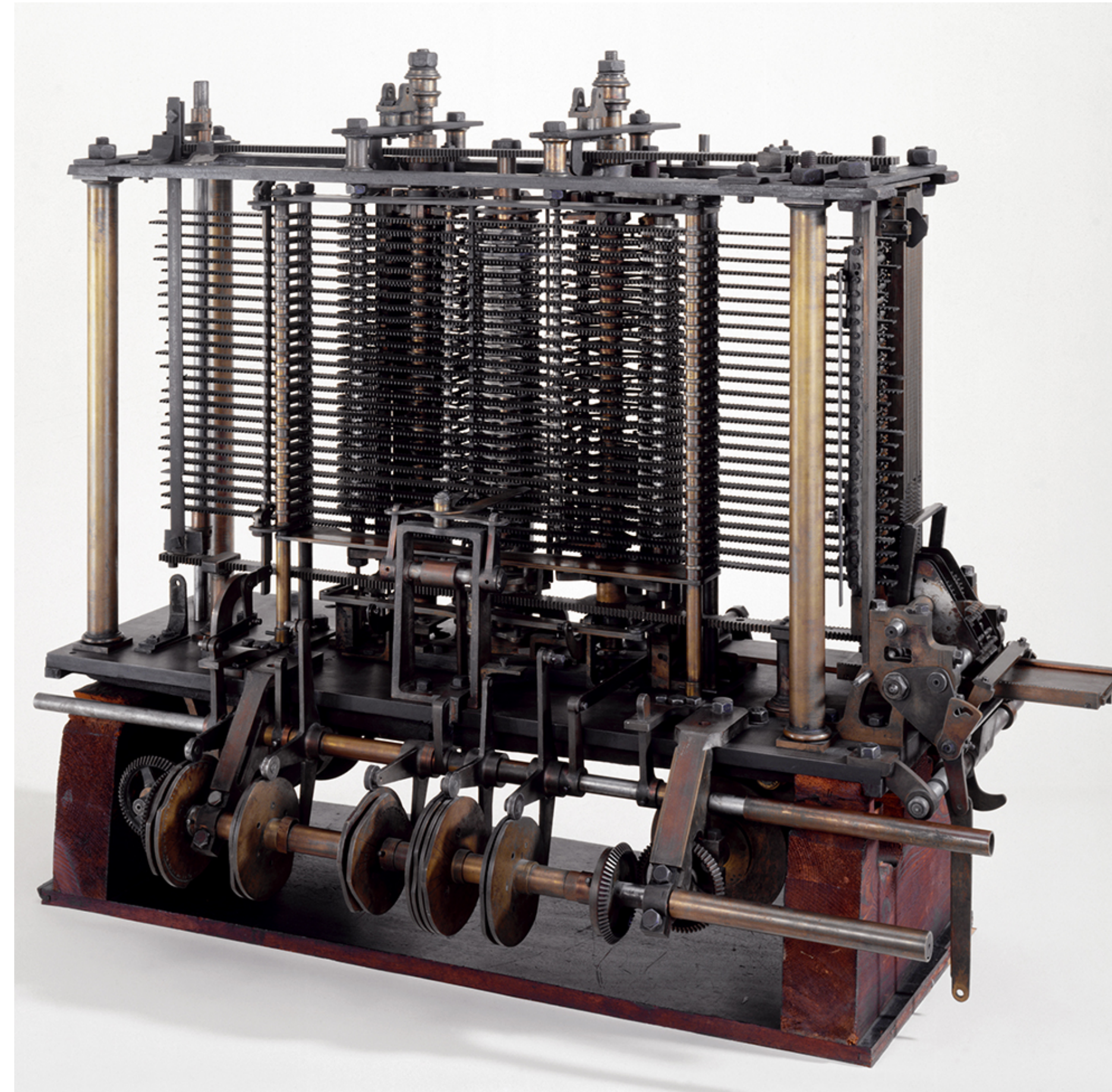
## Charles Babbage (1791 - 1871)

- Matemático inglês do século XIX
- **Projetou a Máquina Analítica, considerada o primeiro conceito de computador programável**
- A máquina incluía:
  - unidade de processamento
  - memória
  - dispositivos de entrada e saída
- Ideias muito semelhantes aos computadores atuais.

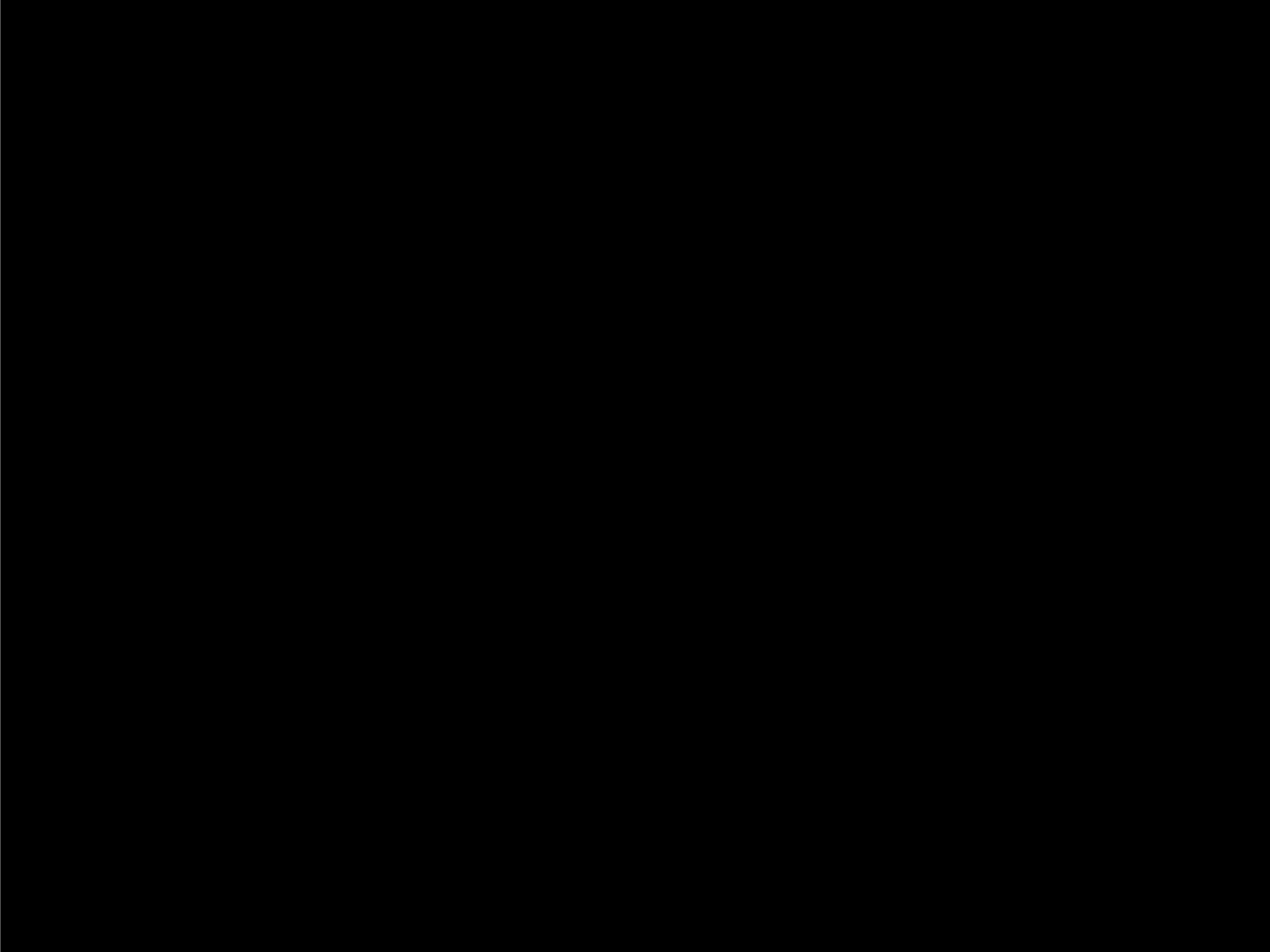


Fonte: [Wikipedia](#)

# Breve história da computação



Fonte: [Wikipedia](#)



# Breve história da computação

## Ada Lovelace (1815 - 1852)

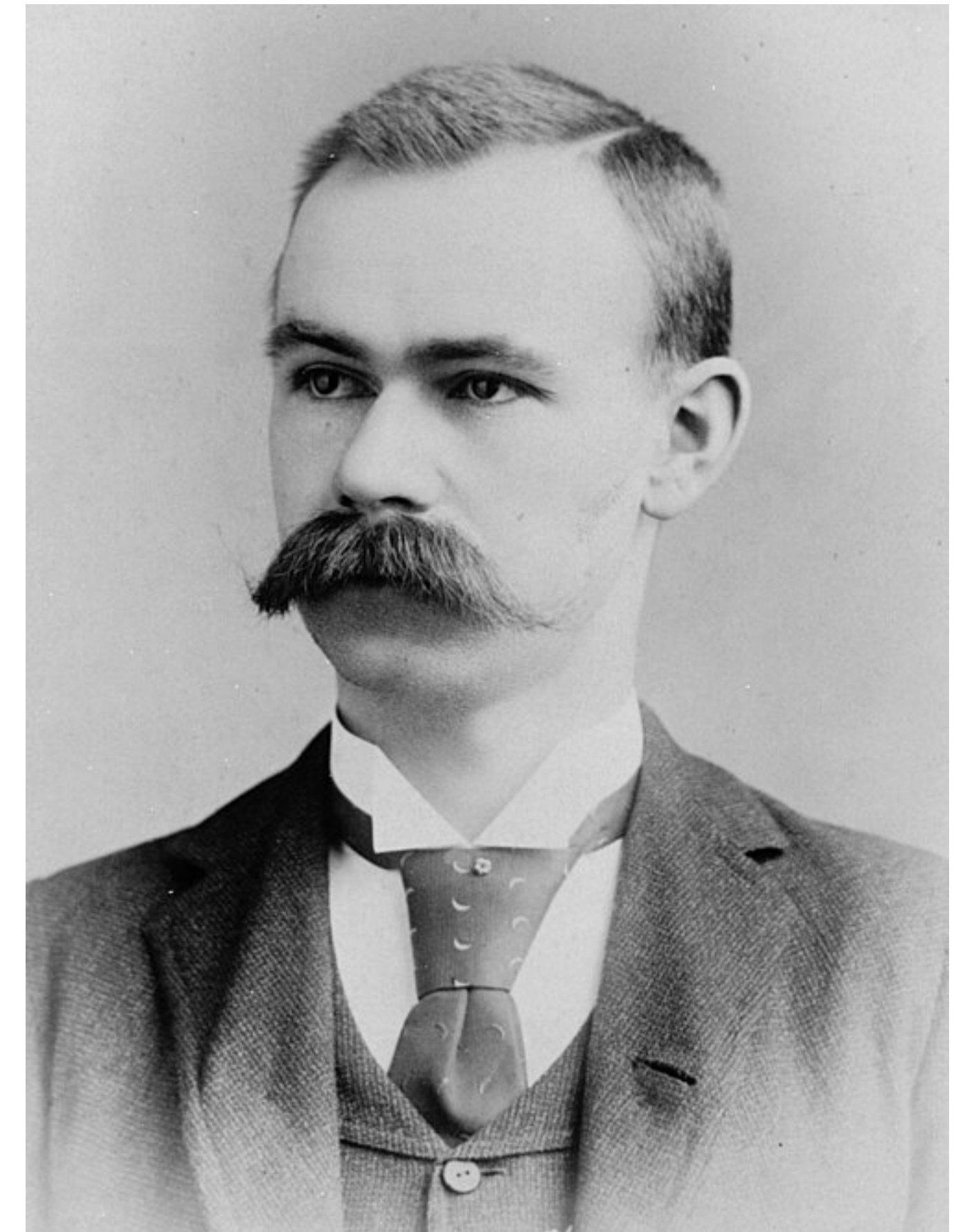
- Matemática inglesa
- Trabalhou com Babbage
- Escreveu o primeiro algoritmo destinado a ser executado por uma máquina
- Por isso é considerada:
  - A primeira programadora da história



# Breve história da computação

## Herman Hollerith

- Empresário americano
- **O Problema:**
  - O **Censo** de 1880 dos EUA **demorou 7 anos** para ter seu resultado publicado
- **A Solução:**
  - Cartões perfurados para tabulação automática
- Foi o precursor da indústria de processamento de dados

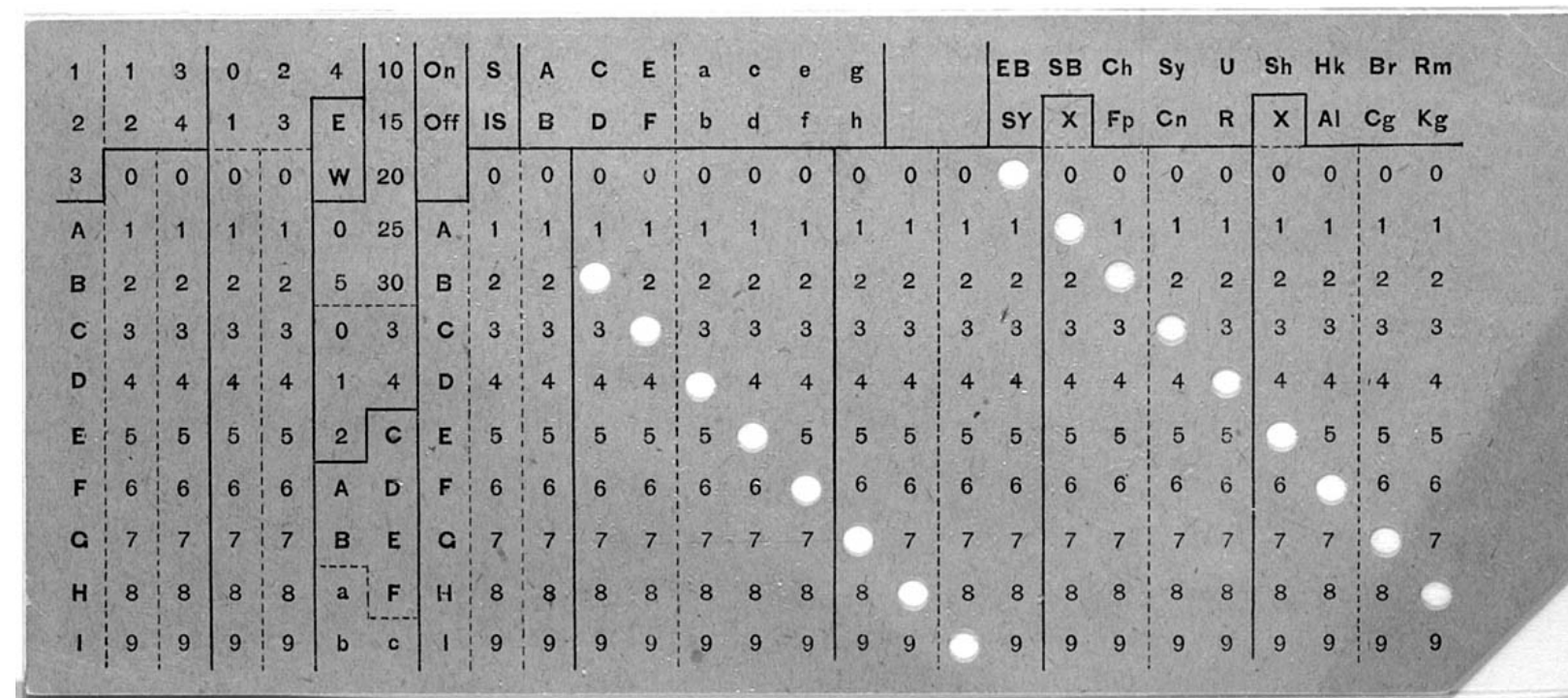


Fonte: [Wikipedia](#)

# Breve história da computação

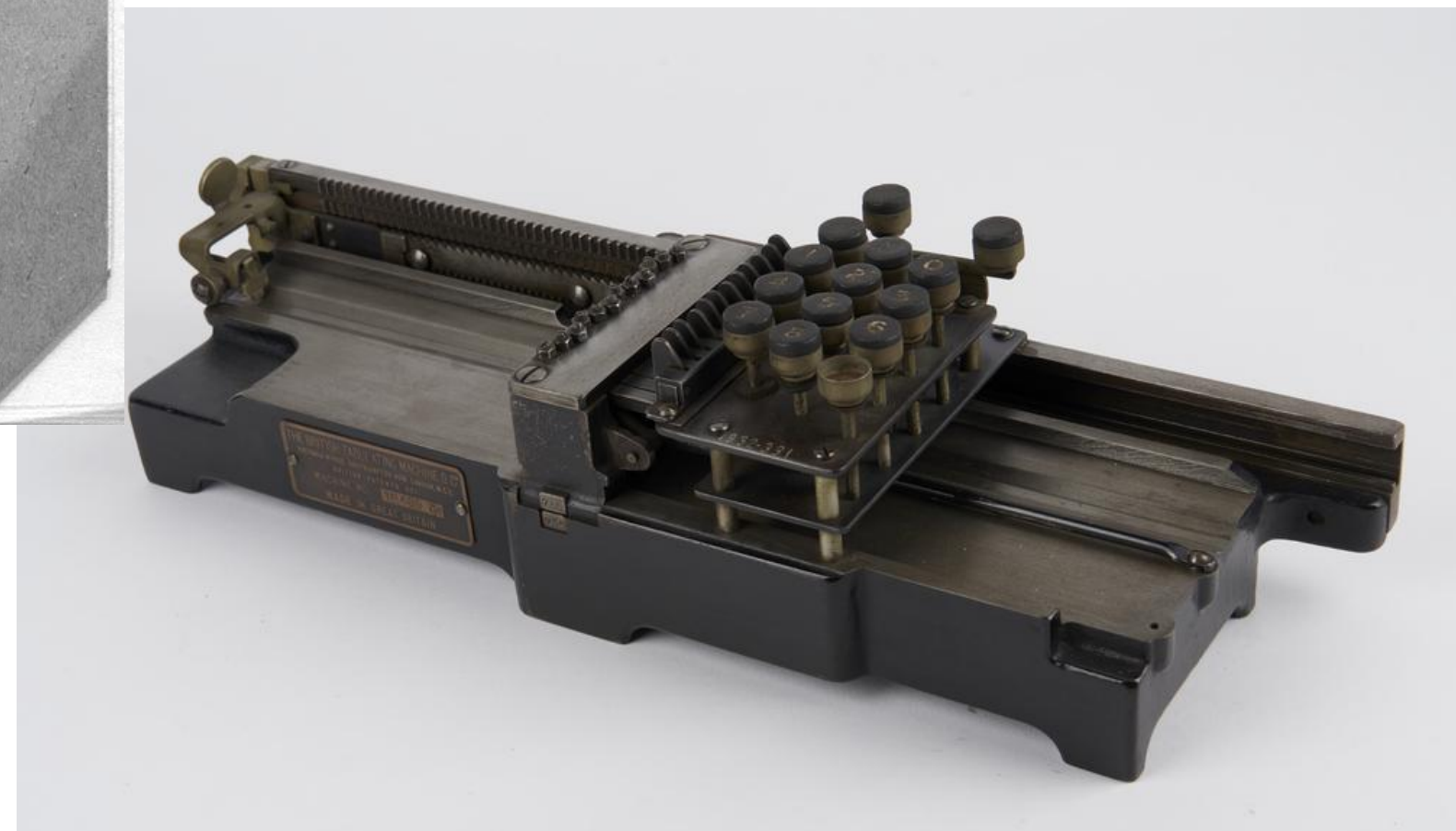
## De Cartões Perfurados à IBM

- Herman Hollerith criou cartões perfurados e três máquinas que revolucionaram o processamento de dados:
  - **Keypunch, Sorter, Tabulator**



1	2	3	4	5	6	7	8	9	10	On	S	A	C	E	a	c	e	g	EB	SB	Ch	Sy	U	Sh	Hk	Br	Rm
2	2	4	1	3	E	15	Off	IS	B	D	F	b	d	f	h	SY	X	Fp	Cn	R	X	Al	Cg	Kg			
3	0	0	0	0	W	20			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
A	1	1	1	1	0	25	A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
B	2	2	2	2	5	30	B	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
C	3	3	3	3	0	3	C	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
D	4	4	4	4	1	4	D	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
E	5	5	5	5	2	C	E	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
F	6	6	6	6	A	D	F	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
Q	7	7	7	7	B	E	Q	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	
H	8	8	8	8	a	F	H	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	
I	9	9	9	9	b	c	I	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	

Cartão perfurado da marca Hollerith



Máquina perfuradora de cartão



Réplica de uma máquina de tabulação

# Breve história da computação

## Alan Turing (1912 - 1954)

- Matemático britânico
- Criou o conceito de **Máquina de Turing**, um modelo teórico que define o que pode ser computado
- Também teve papel crucial na **quebra de códigos durante a Segunda Guerra Mundial**



Fonte: [Wikipedia](#)

# Breve história da computação

## John von Neumann 1903 – 1957

- Matemático húngaro americano
- Propôs a arquitetura de computador usada até hoje
- Essa arquitetura define que um computador possui:
  - memória
  - processador
  - dispositivos de entrada e saída



Fonte: [Wikipedia](#)

# Breve história da computação

## Grace Hopper (1906 - 1992)

- Matemática e Cientista da Computação americana
- Membro da marinha americana (Almirante)
- Trabalhou no Mark I, um computador que é considerado a primeira calculadora automática de larga escala e o precursor dos computadores eletrônicos modernos
- Trabalho nas versões melhoradas, Mark II e Mark III
- Coautora do COBOL



Fonte: [Wikipedia](#)

9/9

0800  
1000

Antan started

" stopped - antan ✓

13" sec (032) MP - MC

(033) PRO 2

connect

~~1.982647000~~

~~2.130476415~~

2.130476415

2.130676415

{ 1.2700 · 9.037 847 025

9.037 846 995 connect

4.615925059(-2)

Relays 6-2 in 033 failed special speed test  
in relay " 10.000 test -

Relays changed

1700

Started Cosine Tape (Sine check)

1525

Started Mult + Adder Test.

1545



Relay #70 Panel F  
(moth) in relay.

First actual case of bug being found.

~~1630~~

Antan started.

1700

closed down.

Relay  
2145  
Relay 3370

Como funciona um computador?

# Como funciona um computador?

## Arquitetura de von Neumann

- CPU(Unidade de processamento central):
  - Executa o programa
  - A CPU está sempre se perguntando: “O que fazer agora”?
  - Não exatamente um cérebro - muito “burro”, mas muito rápido
- Dispositivos de entrada:
  - Teclado, Mouse, Tela sensível ao toque.
- Dispositivos de saída:
  - Monitor, Auto-falantes, Impressora, Gravador de DVD

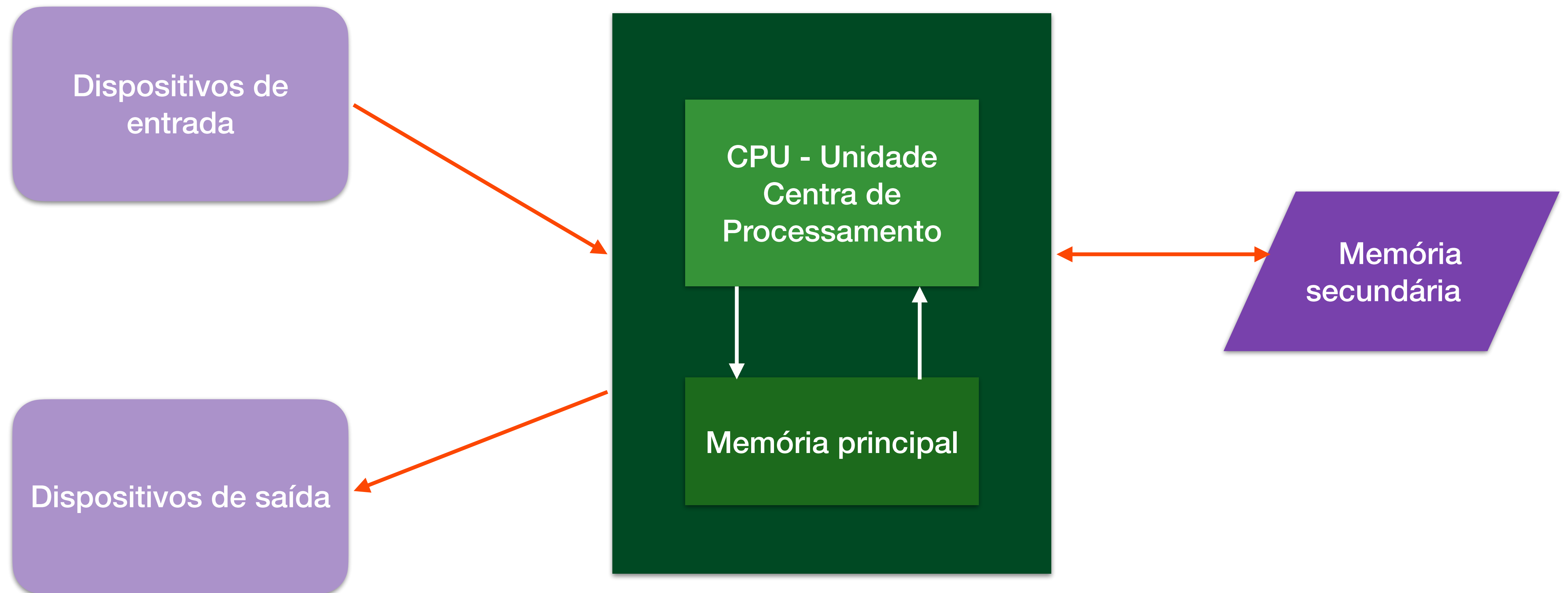
# Como funciona um computador?

## Computador genérico

- Memória principal:
  - Bastante rápida
  - Baixa capacidade de armazenamento
  - Armazenamento temporário
  - Também conhecida por RAM(Memória de acesso aleatório)
- Memória secundária:
  - Devagar, grande e armazenamento permanente - dura até ser deletado - unidade de disco / cartão de memória

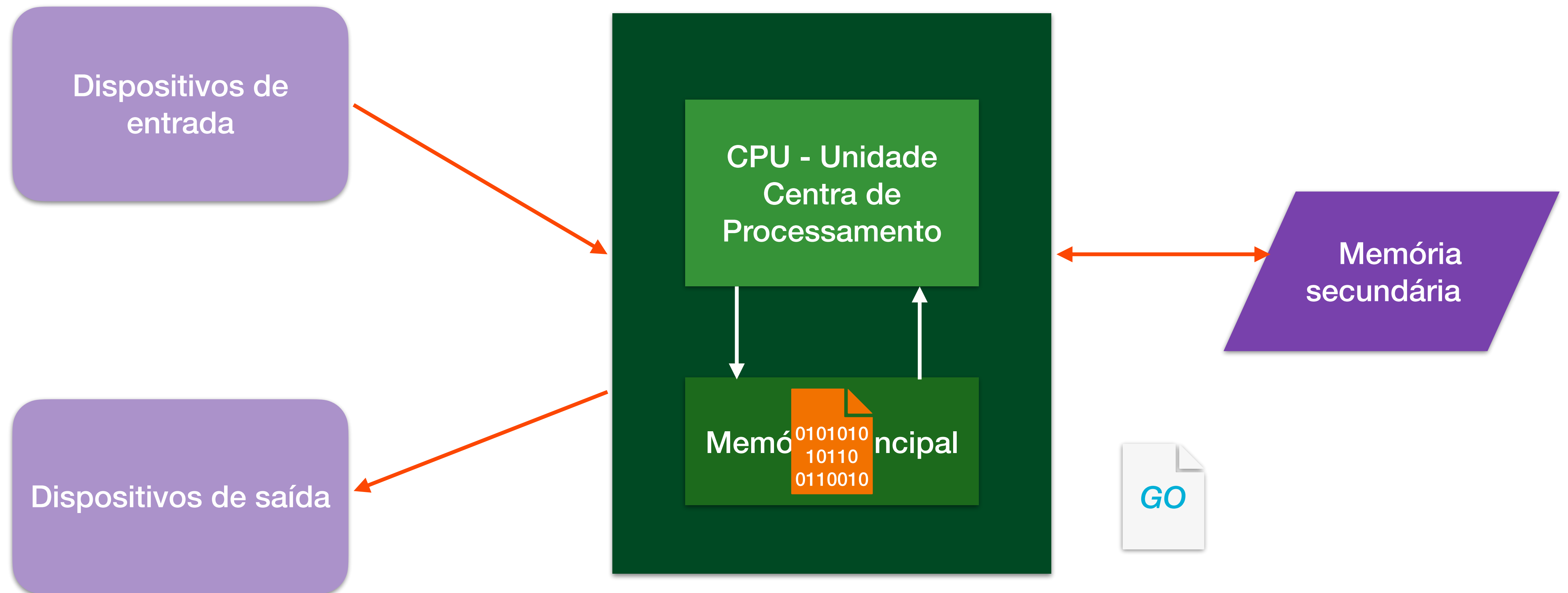
# Como funciona um computador?

## Computador genérico



# Como funciona um computador?

## Como um programa é executado



O que é um algoritmo?

# O que é um algoritmo?

## Introdução

- Da antiguidade, cerca de 300-400aC, já se tem notícias de algoritmos sendo desenvolvidos embora, obviamente, não explicitamente tratados com tal.
- Euclides, já propunha um algoritmo, para se resolver o problema de calcular o máximo divisor comum entre dois inteiros positivos
- Embora as vezes não percebemos, utilizamos algoritmos no nosso dia-a-dia e não sabemos
  - Para a execução de alguma tarefa ou mesmo resolver algum problema, muitas vezes inconscientemente executamos algoritmos

# O que é um algoritmo?

## Introdução

- Matemática
  - Descreve o processo de cálculo, ou resolução de um grupo de problemas semelhantes, em que se estipulam, com generalidade e sem restrições formas gerais para obtenção do resultado, ou da solução do problema.
- Computação
  - Descreve um conjunto bem definido de regras processos destinados à solução de um problema com um conjunto finito de etapas

# O que é um algoritmo?

## Introdução

- Um algoritmo **pode ser comparado:**
  - Receita culinária
  - Passo a passo para montar um móvel
- Uma sequência de passos que devem ser seguidos para atingir um determinado objetivo
  - Caso um dos passos seja desconsiderado, o objetivo não será alcançado

# O que é um algoritmo?

## Introdução

- Não representa necessariamente um programa de computador, e sim os passos necessários para realizar uma tarefa
- Algoritmo não é a solução do problema
  - Um problema pode ter mais de uma solução
- O algoritmo é um caminho para a solução do problema
  - Em geral existem muitos caminhos que levam a solução

# O que é um algoritmo?

## Formas de representar um algoritmo

- Algoritmos podem ser representados de várias formas:
  - Linguagem natural
  - Pseudocódigo
  - Fluxogramas
  - Código em linguagem de programação

# O que é um algoritmo?

## Linguagem Natural

- Algoritmo para: Determinar se o aluno foi aprovado
  - Obter as notas das duas provas
  - Calcular a média aritmética
  - Se a média for maior ou igual a 7
    - Aluno foi aprovado
  - Senão
    - Aluno foi reprovado

# O que é um algoritmo?

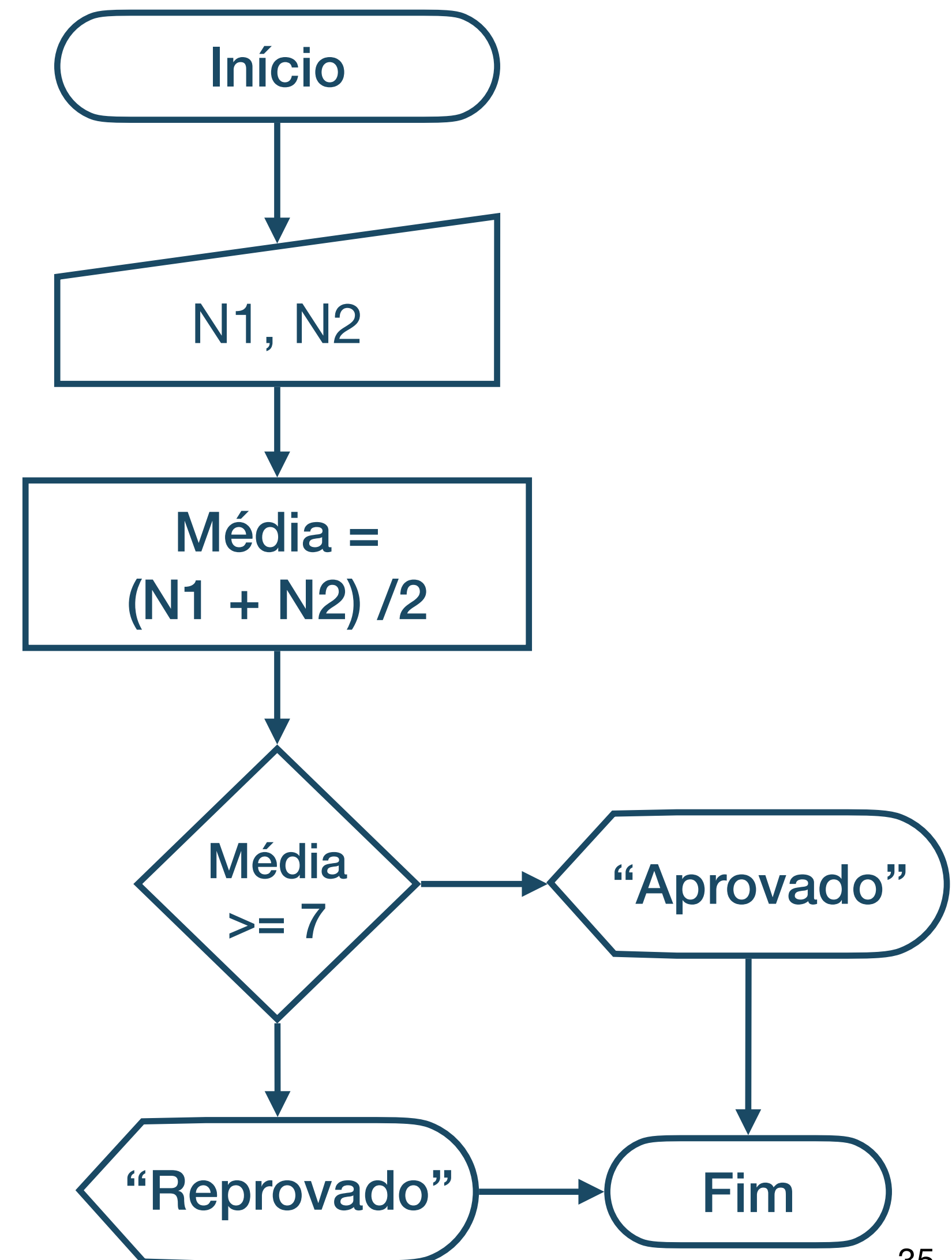
## Linguagem Natural

- Pouco utilizada na prática
  - Ambiguidade
  - Má interpretação
  - Imprecisa
- Exemplo: Afrouxar ligeiramente (quanto é ligeiramente?)
  - Formato preciso: Afrouxar a porca girando 30° graus no sentido anti-horário

# O que é um algoritmo?

## Fluxograma

- Algoritmo para: Determinar se o aluno foi aprovado
  - Obter as notas da primeira e da segunda prova
  - Calcular a média aritmética entre as duas
  - Se a média for maior ou igual a 7
    - Aluno foi aprovado
  - Senão
    - Aluno foi reprovado



# O que é um algoritmo?

## Pseudo-código

**Algoritmo** Média de duas notas

**Var** n1, n2, media: **real**

**Início**

**leia** n1

**leia** n2

  media  $\leftarrow$  (n1 + n2)/2

  Se media  $\geq$  7

  Então **imprima** "Aprovado"

  Senão **imprima** "Reprovado"

**Fim**

# O que é um algoritmo?

## Linguagens de programação

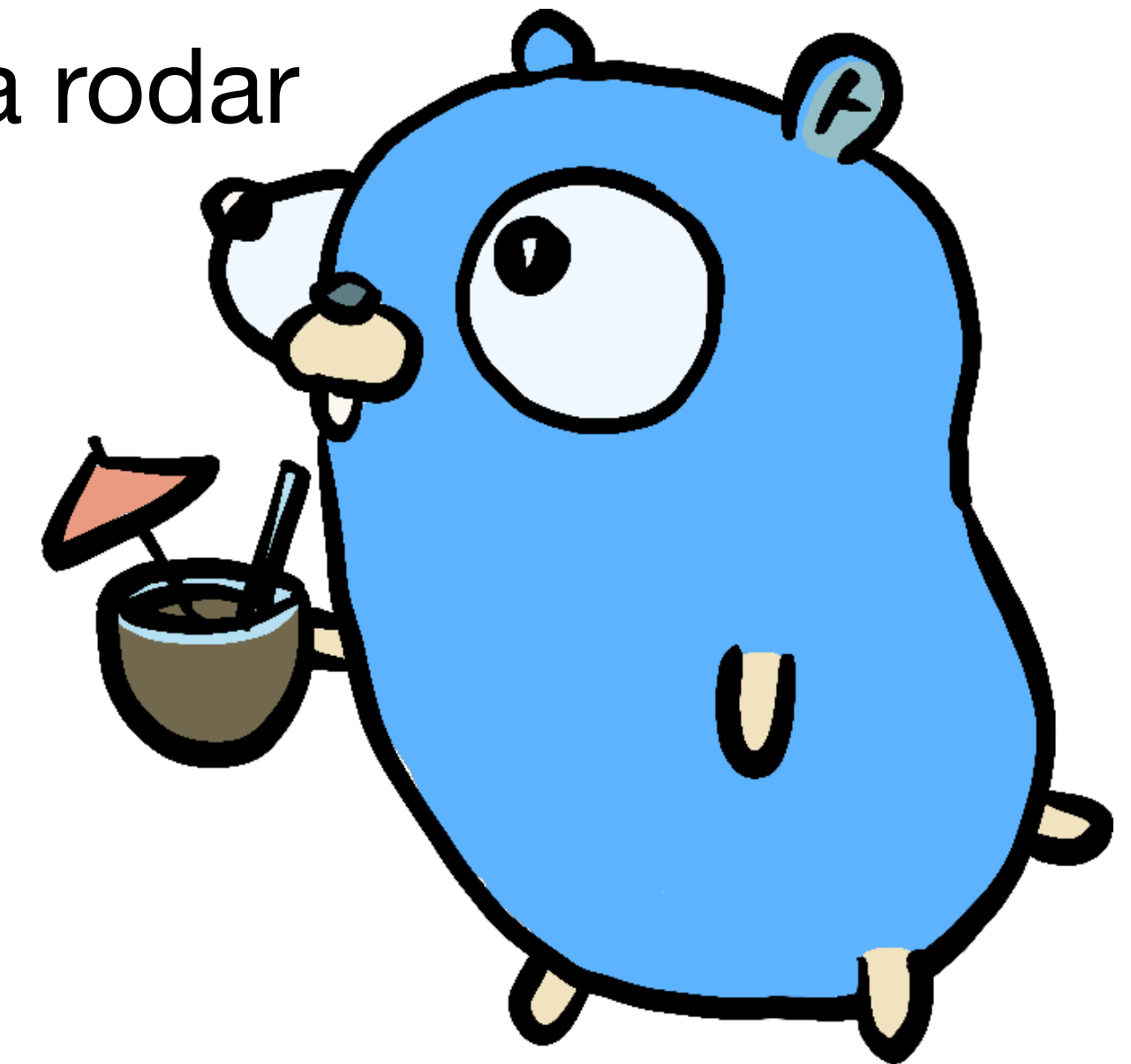
- Para que um computador execute um algoritmo, usamos linguagens de programação.
- Alguns exemplos:
  - C
  - Java
  - Python
  - JavaScript
  - Go

**Instalando Go**

# Introdução à linguagem Go

## O que é Go (Golang) ?

- Criada por Robert Griesemer, Rob Pike e **Ken Thompson** (o criador do **B**, antecessor do **C**) em 2009 pelo Google
- O problema: C++ era lento para compilar, Python era lento para rodar
- Foco em:
  - Simplicidade
  - Performance
  - Concorrência nativa
  - Código limpo



Fonte: [Go Google Source](#)

# Introdução à linguagem Go

## Onde Go é usada?

- Go é utilizado em grandes sistemas como:
  - infraestrutura de nuvem
  - servidores web
  - sistemas distribuídos
  - ferramentas de linha de comando

# Introdução à linguagem Go

## Onde Go é usada?

- Exemplos de projetos famosos escritos em Go:



Fonte: [Documentação oficial](#)



Fonte: [Wikipedia](#)



Fonte: [Documentação oficial](#)

# Introdução à linguagem Go

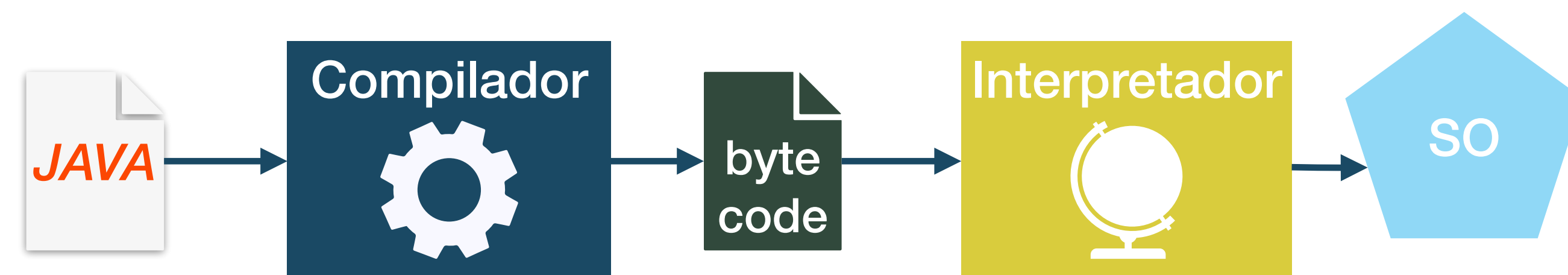
## Características Principais

- **Compilada:**
  - Gera um arquivo executável direto para a máquina (sem necessidade de máquina virtual pesada como o Java).
- **Estaticamente Tipada:**
  - O tipo da variável é verificado antes do programa rodar, evitando erros bobos.
- **Garbage Collected:**
  - O programador não precisa se desesperar com `free()` ou liberar memória manualmente.
- **Simples:**
  - Possui apenas 25 palavras-chave (C tem 32, C++ tem mais de 80).

# Introdução à linguagem Go

## Compilado vs Interpretado

- Linguagens compiladas
  - código é convertido para executável
  - exemplo: C, Go
- Linguagens interpretadas
  - O interpretador se comunica com o S.O
  - exemplo: Python
- Linguagens híbridas
  - compilação para bytecode + máquina virtual
  - exemplo: Java



# Introdução à linguagem Go Go

- Download em: <https://go.dev>
- Verificar instalação:
  - `go version`

# Introdução à linguagem Go Go

## Hello World

Execute *go run nome\_do\_ser\_arquivo.go* para compilar e executar o programa

basic compilation and execution of the program

```
package main  
  
import "fmt"  
  
func main() {  
    fmt.Println("Hello, World!")  
}
```

Define o programa principal

Importa a biblioteca de entrada e saída

Função principal

Função que imprime o texto na saída

**Ambiente de desenvolvimento**

# Ambiente de desenvolvimento

## Linguagens de programação

- Nesta disciplina utilizaremos:



Fonte: [Página oficial](#)



Fonte: [Github Blog](#)

- Isso permite:
  - programar direto no navegador
  - compartilhar código facilmente
  - entregar atividades



Fonte: [Página oficial](#)

# Referências

- Introdução ilustrada à computação – Larry Gonnick
- Como o termo “bug” foi implementado no nosso vocabulário de tecnologia?  
A história de Grace Hopper