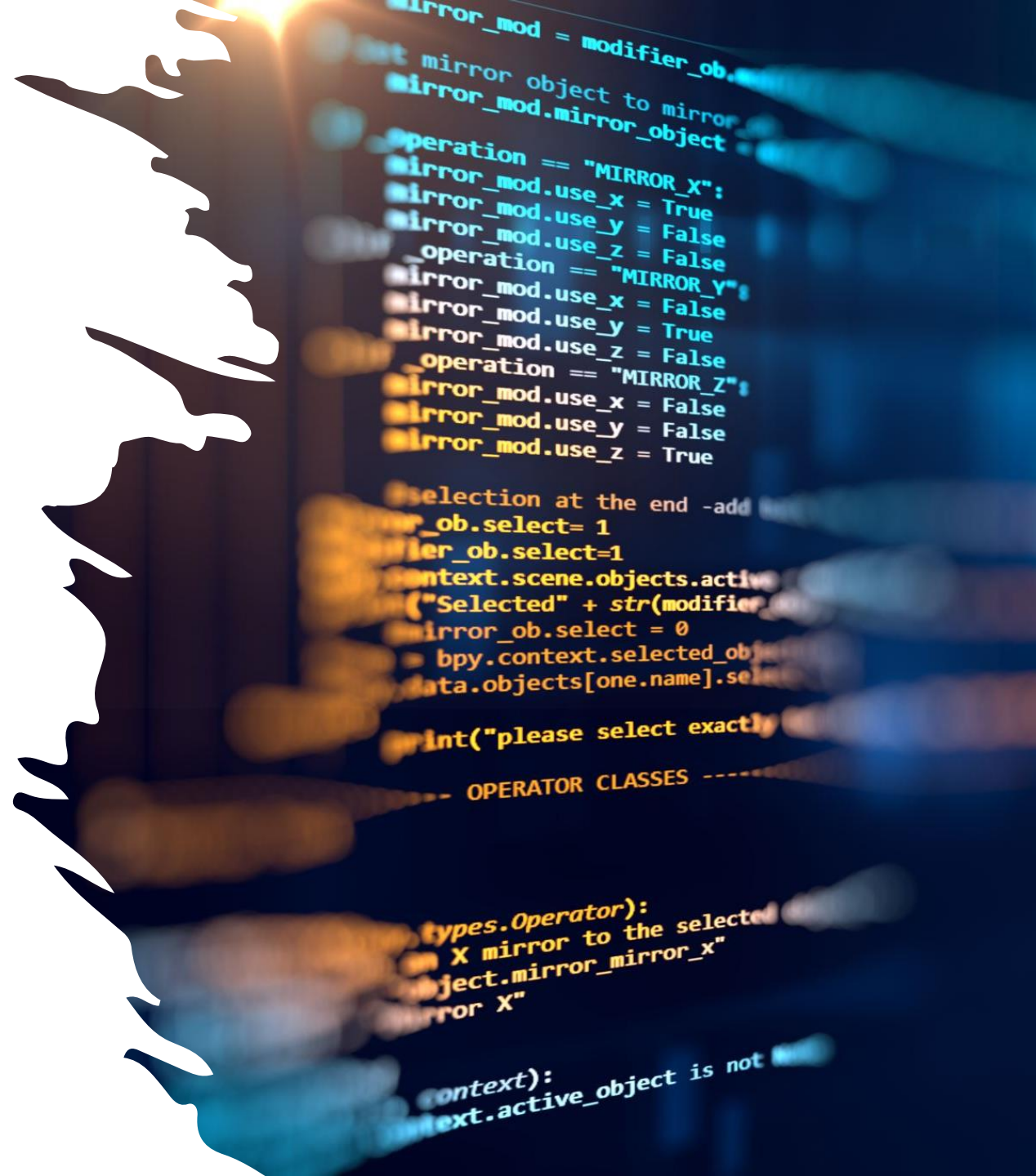


# Linguagens de Baixo Nível

Profª Ma. Cleane Nascimento

# Programação...

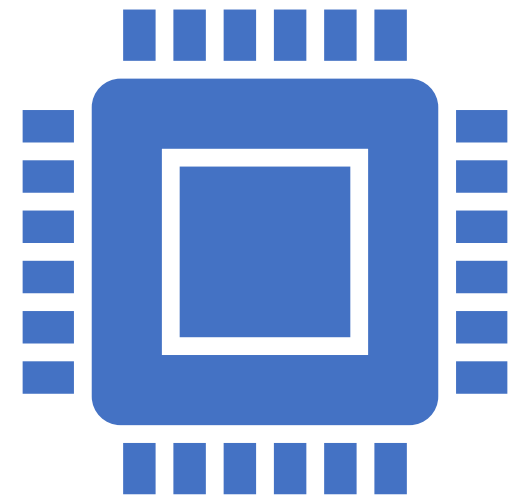
É o processo de projetar, escrever, provar, depurar e manter o código fonte de programas de computador. O código fonte é escrito em uma linguagem de programação. O propósito da programação é criar programas que executem um comportamento desejado.

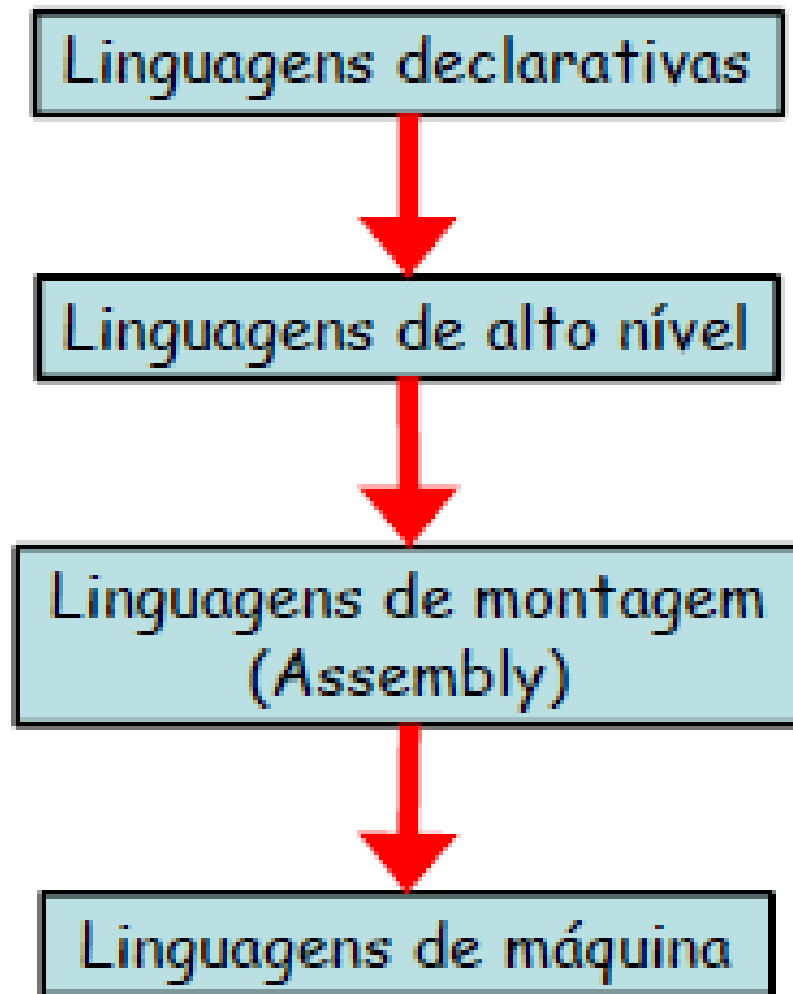


Para criar um programa que o computador interprete e execute as instruções escritas, deve-se usar uma linguagem de programação. No início, os computadores interpretavam somente instruções em uma linguagem específica, uma linguagem de programação de baixo nível conhecida como código máquina, excessivamente complicada para programar. Consiste somente em cadeias de números 1 e 0 (sistema binário).



Para facilitar o trabalho de programação os primeiros cientistas que trabalhavam na área decidiram substituir as instruções, sequências de um e zero, por palavras ou letras do inglês, codificando e criando assim uma linguagem de maior nível conhecida como Assembly. Por exemplo, para somar se usa a letra A, do inglês add. Realmente escrever em linguagem assembly é basicamente o mesmo que com a linguagem máquina, mas as letras e as palavras são mais fáceis de lembrar e entender que sequências de números binários.

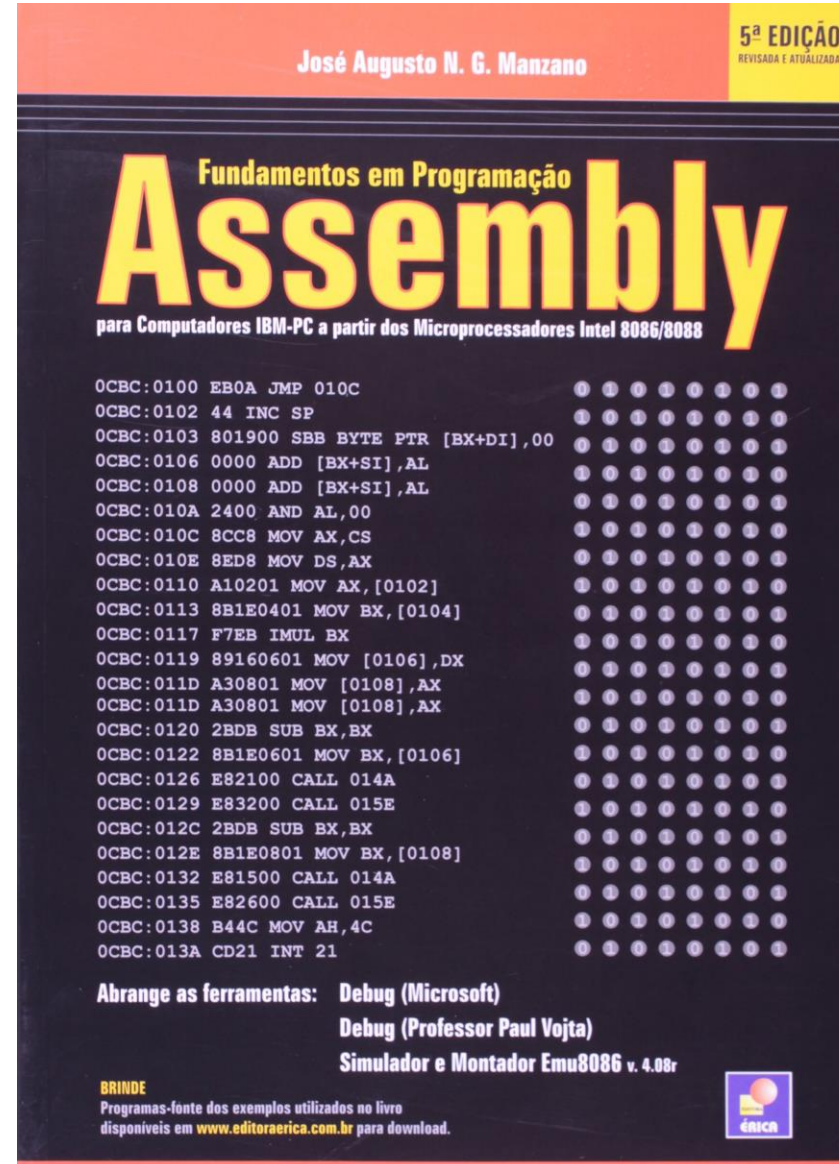






# Baixo Nível

- ❑ Voltada ao dispositivo, de modo a estar mais próxima da simbologia da máquina;
- ❑ Se comunica com o computador de maneira mais rápida e eficaz;
- ❑ Sintaxe complexa por não contar com comandos intuitivos.



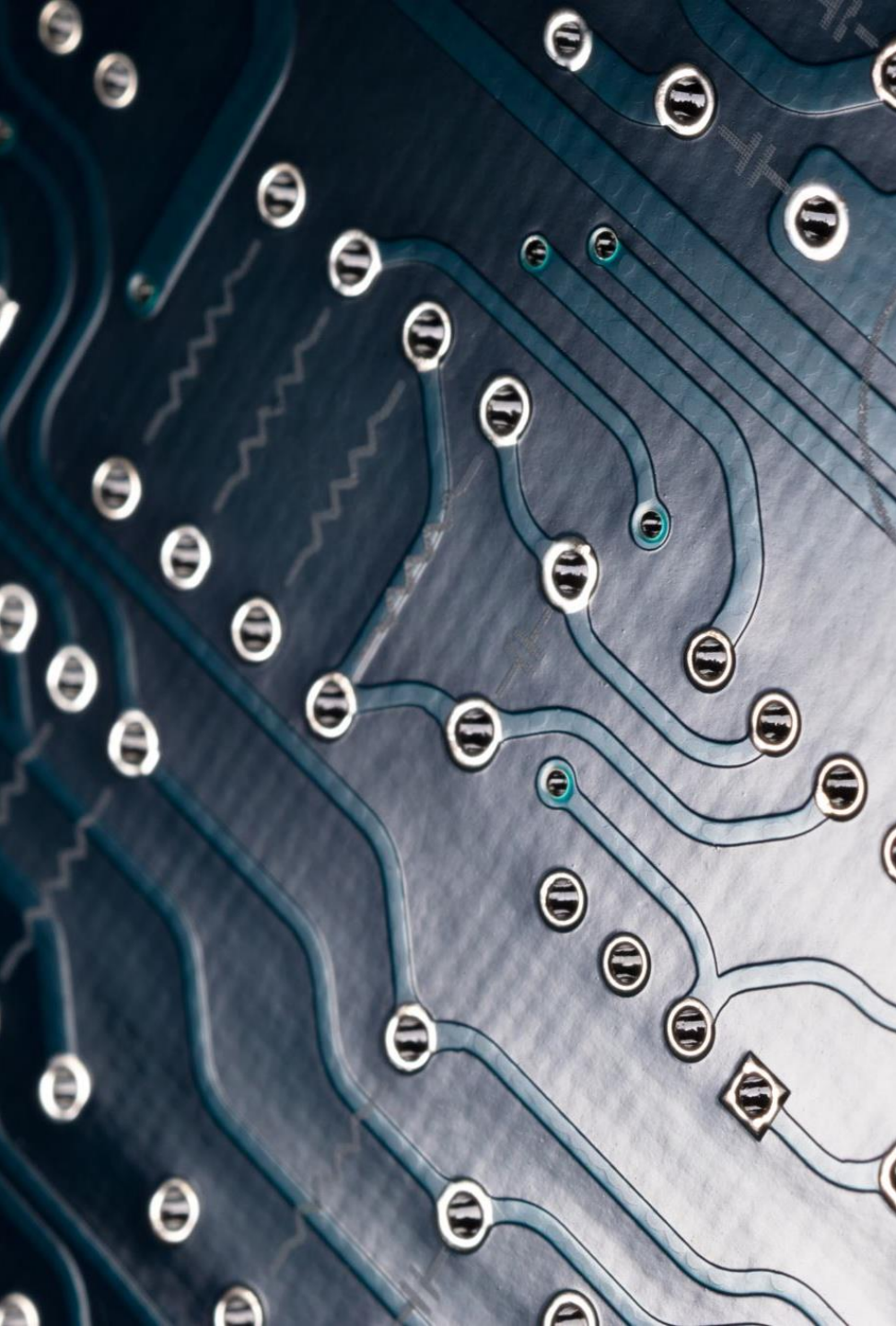
# *Alto nível*

- ❑ Mais compreensível pelo ser humano;
- ❑ Facilita a maneira de programar, por ser intuitiva, amigável e fácil de aprender, principalmente para iniciantes na área de programação.



Enquanto que uma tarefa tão simples como multiplicar dois números necessita um conjunto de instruções em linguagem assembly, em uma linguagem de alto nível basta com uma.





# Linguagem de Máquina

---

Sistema de códigos diretamente interpretável por um circuito microprogramável, como o microprocessador de um computador ou um microcontrolador. Um programa em código de máquina consiste em uma sequência de números que significam uma sequência de instruções a serem executadas. A linguagem máquina trabalha com dois níveis de voltagem. Tais níveis, por abstração, se simbolizam com o **zero (0)** e o **um (1)**, por isso a linguagem de máquina só utiliza estes signos.

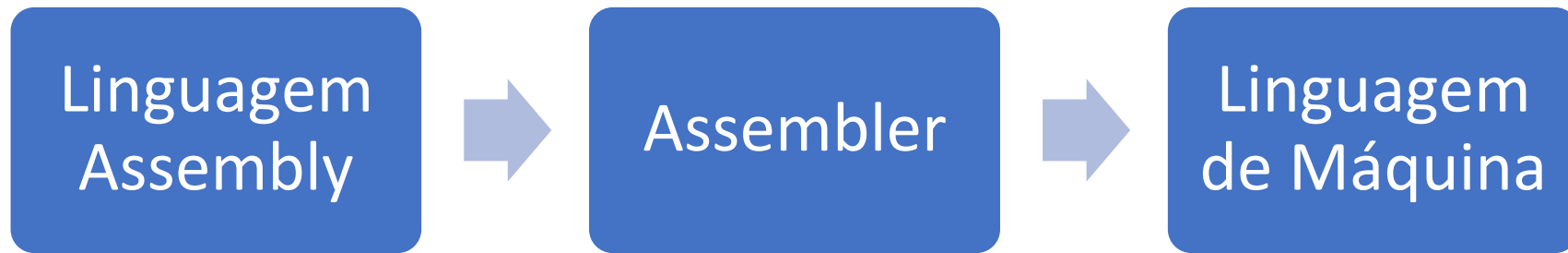
# Linguagem Assembly

É uma linguagem de programação de baixo nível para computadores, microcontroladores e outros circuitos integrados programáveis. A linguagem de máquina, que é um mero padrão de bits, torna-se legível pela substituição dos valores em bruto por símbolos chamados mnemónicos. Estes símbolos são geralmente definidos pelo fabricante do hardware e está baseada em códigos que simbolizam os passos do processamento (as instruções).



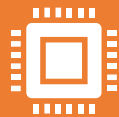
Uma linguagem assembly é portanto específica de cada arquitetura de computador, podendo ser usada somente por um microprocessador específico. Isso contrasta com a maioria das linguagens de programação de alto nível que idealmente são portáteis, o que significa que um programa pode ser executado em uma variedade de computadores.

**Assembly é linguagem de montagem e não linguagem de máquina.**



LINGUAGEM ASSEMBLY	LINGUAGEM DE MÁQUINA
só é compreensível para seres humanos, não para computadores.	só é compreensível para computadores.
os dados podem ser representados com a ajuda de mnemônicos como Mov, Add, Sub, End, etc	os dados são representados apenas com a ajuda do formato binário (0s e 1s), hexadecimal e octadecimal.
é fácil de entender pelo ser humano em comparação com a linguagem de máquina.	é muito difícil de ser entendida pelos seres humanos.
Modificações e correção de erros podem ser realizadas.	Modificações e correção de erros não podem ser feitas
É fácil memorizar a linguagem assembly porque alguns alfabetos e mnemônicos são usados.	é muito difícil de memorizar devido ao uso de formato binário (0s e 1s).
A execução é lenta em comparação com a linguagem de máquina.	A execução é rápida em linguagem de máquina porque todos os dados já estão presentes em formato binário.
Assembler é usado como um tradutor para converter mnemônicos em uma forma compreensível por máquina.	Não há necessidade de um tradutor. A linguagem de máquina já está em uma forma compreensível pela máquina.
depende da máquina e não é portátil.	depende do hardware.

# Instrução



É uma ação que determina a atividade que um computador deve realizar em certo instante.



Pode ser escrita a partir da definição do código de operação (opcode) e seus operandos.



O opcode transmite o dado e o que deve ser feito com o dado pelo processador.



Os operandos além de representar dados podem armazenar endereços de acesso a memória



As instruções  
podem ser  
utilizadas  
para:

- Efetuar ações matemáticas lógicas;
  - Efetuar movimentação dos dados no processador;
  - Efetuar entradas e a saída;
  - Efetuar o controle de desvios condicionais e laços.
- 
- A execução de instruções é efetivada com o auxílio de um dispositivo chamado registrador.

# Existem dois tipos de registradores (\$):

**Gerais**



Usados no armazenamento de dados e sua movimentação entre a memória e o processador.

**Propósito Especial**




Usados na execução e certa instrução, sendo normalmente gerenciados pela Unidade de Controle.

# O que é Assembly?

Linguagem de baixo nível, também conhecida como “linguagem de montagem”;

é atrelada à arquitetura de uma certa CPU, ou seja, ela depende completamente do hardware.



---

As primeiras linguagens Assembly surgiram na década de 50, na chamada segunda geração das linguagens de programação

---

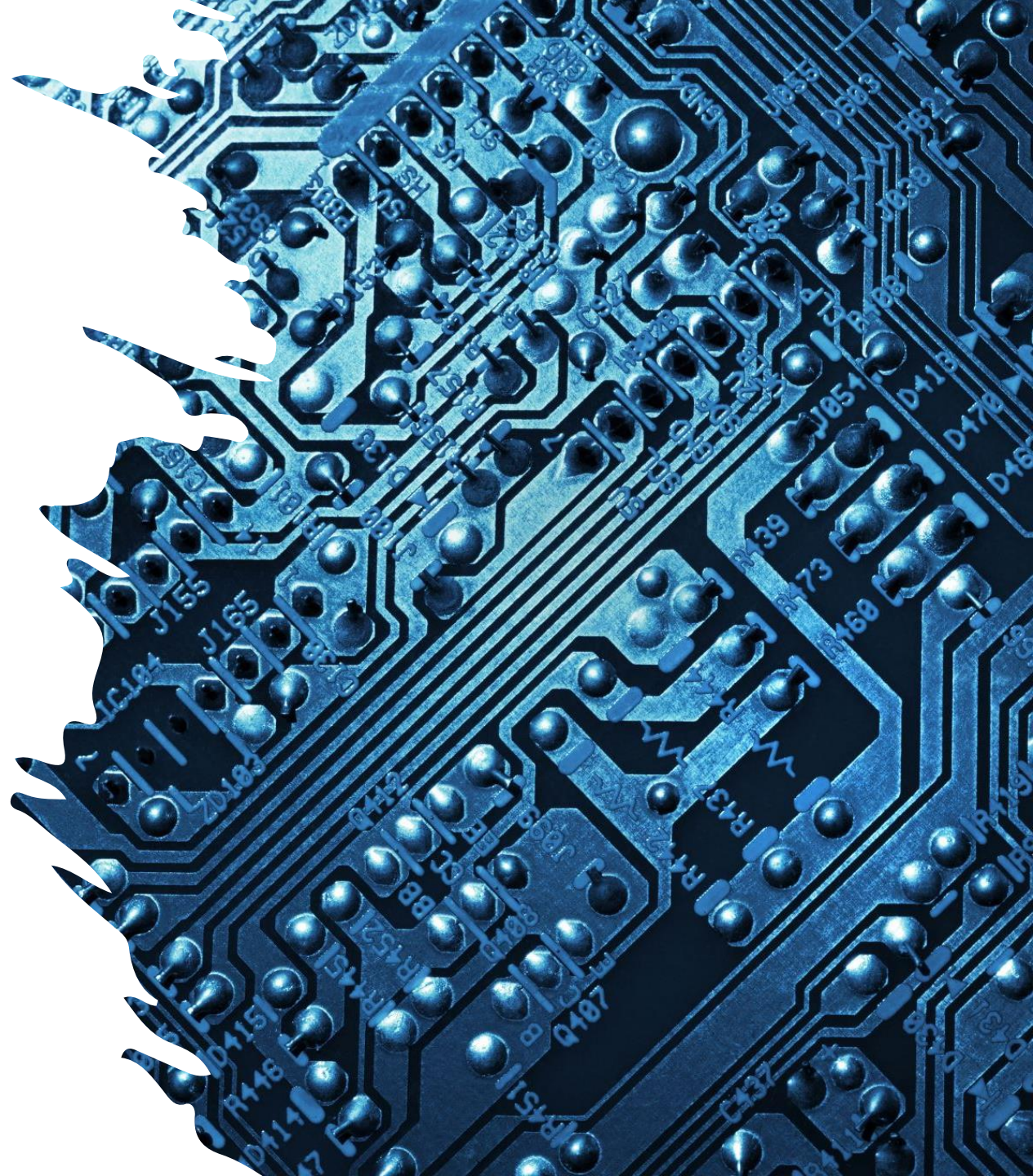
A segunda geração visou libertar os programadores de dificuldades como lembrar códigos numéricos e calcular endereços

---

muito usada para várias aplicações até os anos 80, quando foi substituída pelas linguagens de alto nível, devido a necessidade de aumento da produtividade de software



- Atualmente **Assembly** é usada para manipulação direta de hardware e para sistemas que necessitem de performance crítica, como: Device drivers, sistemas embarcados de baixo nível e sistemas de tempo real.







# Vamos praticar?

- Em dupla pesquisem e tragam para compartilhar como os colegas sobre a linguagem Assembly:
- Porque utilizar Assembly?
- Quais as vantagens e desvantagens?
- Cada dupla tem até 5 minutos para expor sobre o tema;
- Cada dupla deverá realizar 2 perguntas para outras duplas (1 por vez) e à escolha de quem pode perguntar será realizada no momento da aula.



# Referências

MANZANO, J. A. N. G. **Fundamentos em Programação Assembly**. 1. ed. Editora Érica, 2004.

HENNESSY, J. L.; PATTERSON D. A. **Organização e Projeto de Computadores: A Interface Hardware/Software**. 4. ed. Editora Elsevier, 2013.