

LINGUAGENS DE PROGRAMAÇÃO E SUAS CLASSIFICAÇÕES

Apresentado por: Felipe Barbosa



Será abordado:

*LINGUAGEM DE PROGRAMAÇÃO
E SUAS CLASSIFICAÇÕES*

- *PARADIGMA*
- *ESTRUTURA DE TIPOS*
- *GRAU DE ABSTRAÇÃO*
- *GERAÇÃO*



PARADIGMA

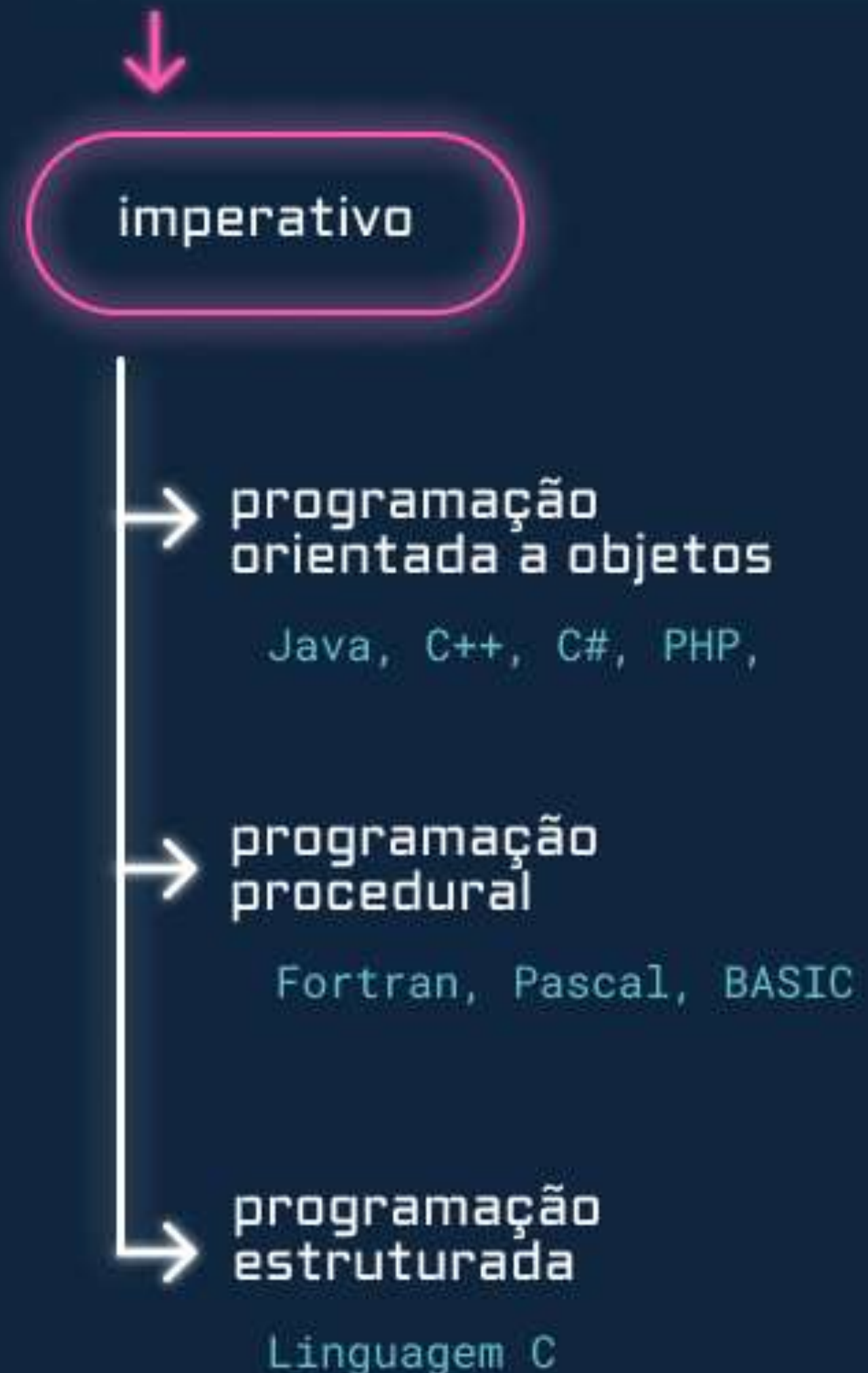
Um **paradigma** de programação é uma abordagem ou estilo que define como os programas são estruturados e operam.

- Cada paradigma oferece regras e métodos únicos para resolver problemas de software.
- Compreender os paradigmas é crucial para os programadores escolherem a melhor abordagem para cada problema e colaborarem efetivamente com outros desenvolvedores.

PARADIGMA

IMPERATIVO: Nesse paradigma, o foco está na especificação de como um programa opera. Ele descreve passo a passo quais instruções devem ser executadas.

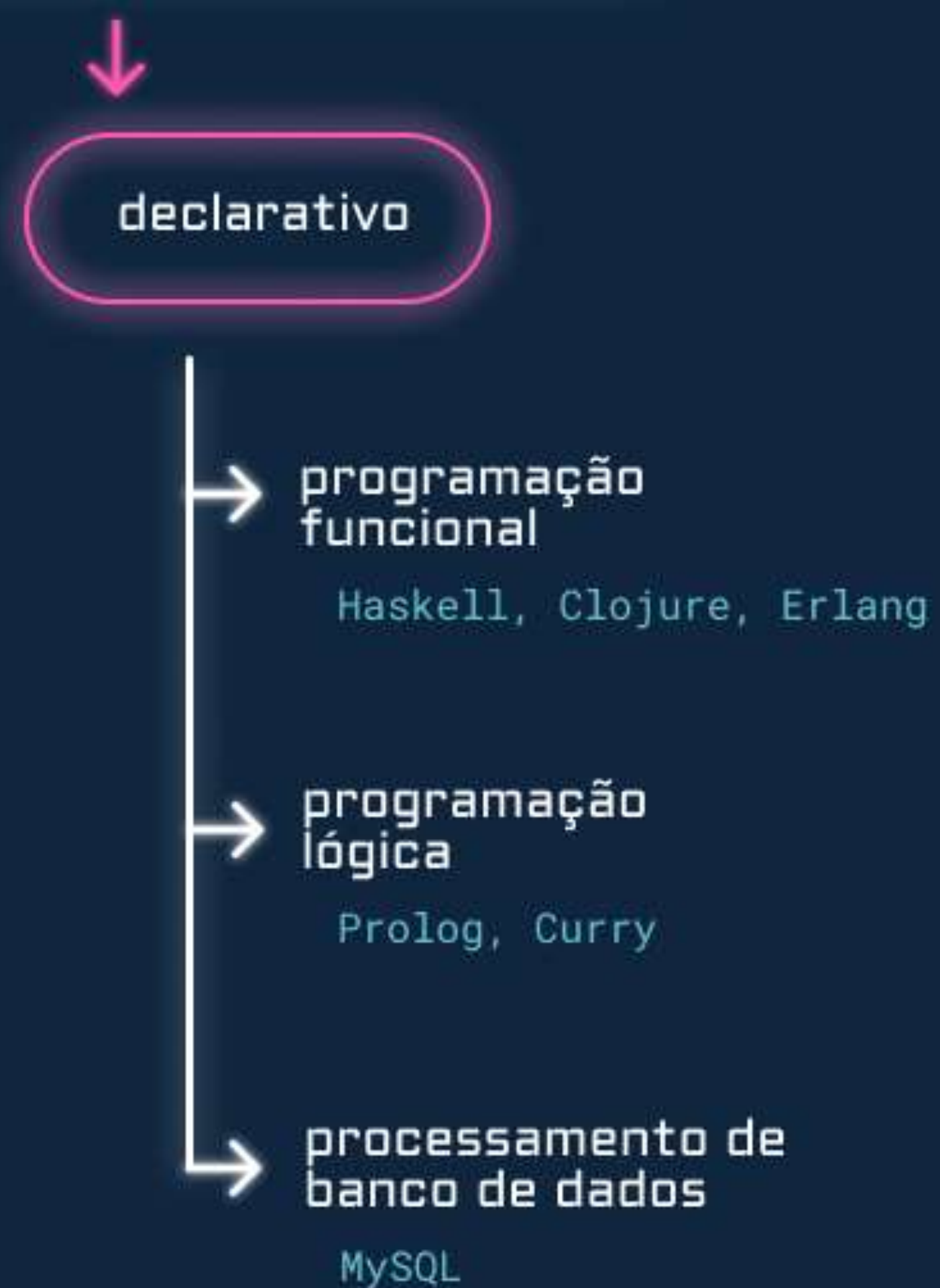
- PROCEDURAL: Organiza o código em procedimentos ou funções que podem ser chamados conforme necessário.
- ESTRUTURA DE BLOCOS: Utiliza blocos de código para controlar o fluxo de execução do programa.
- ORIENTAÇÃO A OBJETO: Baseia-se na ideia de objetos, que contêm dados na forma de campos (ou atributos) e código na forma de procedimentos (ou métodos).



PARADIGMA

DECLARATIVO: Aqui, o foco é na descrição do que um programa deve realizar, sem necessariamente definir como isso deve ser feito.

- FUNCIONAL: Os programas são construídos a partir de funções, que são tratadas como cidadãos de primeira classe.
- PROGRAMAÇÃO LÓGICA: O programa define relações lógicas entre diferentes variáveis e expressões.



RESUMINDO



P. IMPERATIVO

No paradigma imperativo, o desenvolvedor descreve passo a passo como o programa deve operar, usando instruções que modificam o estado do programa.



P. DECLARATIVO

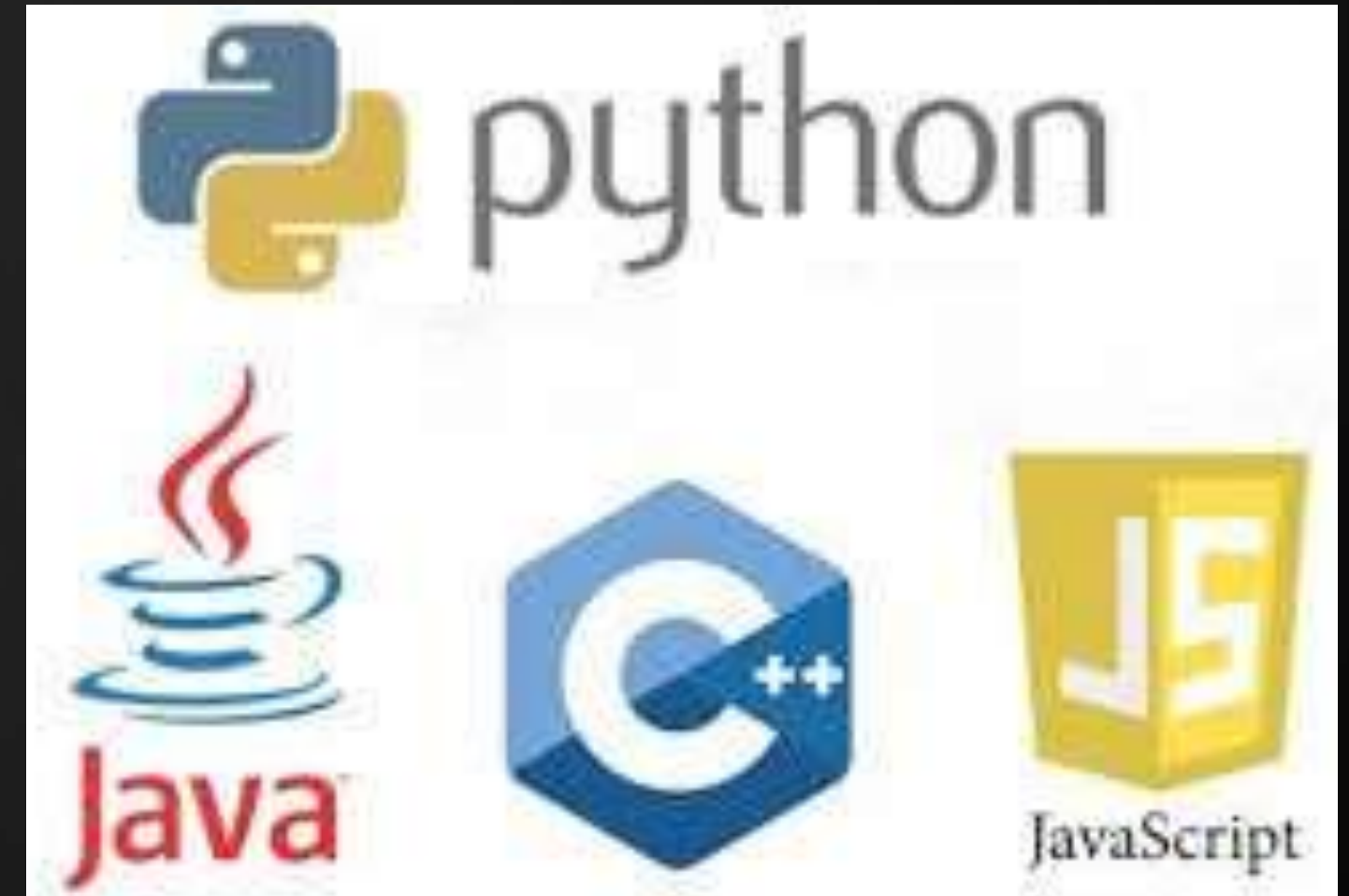
Em contraste, no paradigma declarativo, o foco é em descrever o que o programa deve alcançar, sem especificar passo a passo como isso deve ser feito.



ESTRUTURA DE TIPOS

"Tipada" descreve como uma linguagem de programação trata os tipos de dados.

- Em linguagens tipadas, como Java ou C++, os tipos de dados devem ser especificados e as operações entre tipos incompatíveis podem causar erros.
- Já em linguagens não tipadas, como JavaScript ou Python, as variáveis podem conter valores de qualquer tipo e as conversões entre tipos são feitas de forma mais flexível.

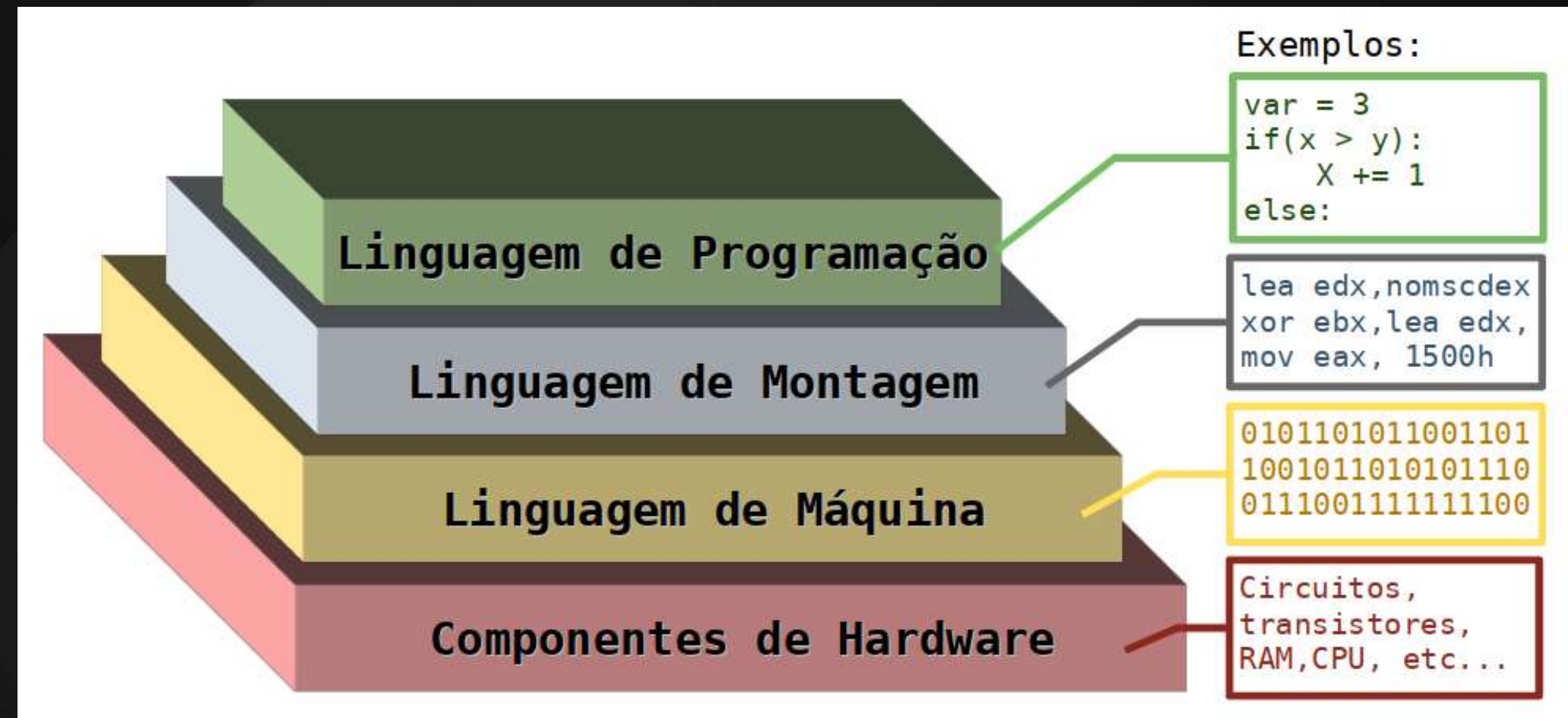


ESTRUTURA DE TIPOS

- **Fracamente tipada:** Permite que diferentes tipos de dados sejam usados sem muitas restrições ou conversões explícitas entre eles.
- **Fortemente tipada:** Exige que os tipos de dados sejam rigidamente seguidos, sem permitir operações entre tipos diferentes sem conversões explícitas.
- **Dinamicamente tipada:** Verifica os tipos de dados em tempo de execução, durante a execução do programa.
- **Estaticamente tipada:** Verifica os tipos de dados em tempo de compilação, antes da execução do programa.

GRAU DE ABSTRAÇÃO

O GRAU DE ABSTRAÇÃO EM LINGUAGENS DE PROGRAMAÇÃO DETERMINA O QUÃO PRÓXIMAS AS INSTRUÇÕES DE CÓDIGO ESTÃO DOS DETALHES DO HARDWARE DO COMPUTADOR.



BAIXO NÍVEL: As instruções são mais próximas da linguagem de máquina e exigem um entendimento detalhado da arquitetura do computador.

MÉDIO NÍVEL: Oferecem um equilíbrio entre a proximidade do hardware e a facilidade de programação.

ALTO NÍVEL: Permitem que os desenvolvedores escrevam programas em termos mais próximos de problemas humanos do que de detalhes de máquina.

GERAÇÃO:

1



PRIMEIRA

Computadores e linguagens de programação associadas a válvulas e transistores, como Assembly e linguagem de máquina.

2



SEGUNDA

Surgimento dos circuitos integrados, com linguagens como COBOL e Fortran.

3



TERCEIRA

Advento dos microprocessadores, com linguagens como C, Pascal e BASIC.

4



QUARTA

Linguagens orientadas a banco de dados e específicas de domínio.

5



QUINTA

Avanços em inteligência artificial, computação quântica e linguagens específicas de domínio.

Resumo do que estudamos



Importância

Essas classificações são essenciais para compreendermos as características e capacidades das linguagens de programação.



Influência

Cada categoria tem uma influência significativa na forma como os programas são desenvolvidos e compreendidos pelos desenvolvedores.



Compreensão

Compreender essas classificações é fundamental para os desenvolvedores escolherem as linguagens mais adequadas para suas necessidades específicas e explorarem todo o potencial da programação computacional.



OBRIGADO!

Pela sua presença e atenção

*Apresentado por: **Felipe Barbosa***

*Professor C.Currículo: **Marcelo Araújo***