

# Python para Inteligência Artificial

Jean Phelipe de Oliveira Lima

jpdol.eng16@uea.edu.br

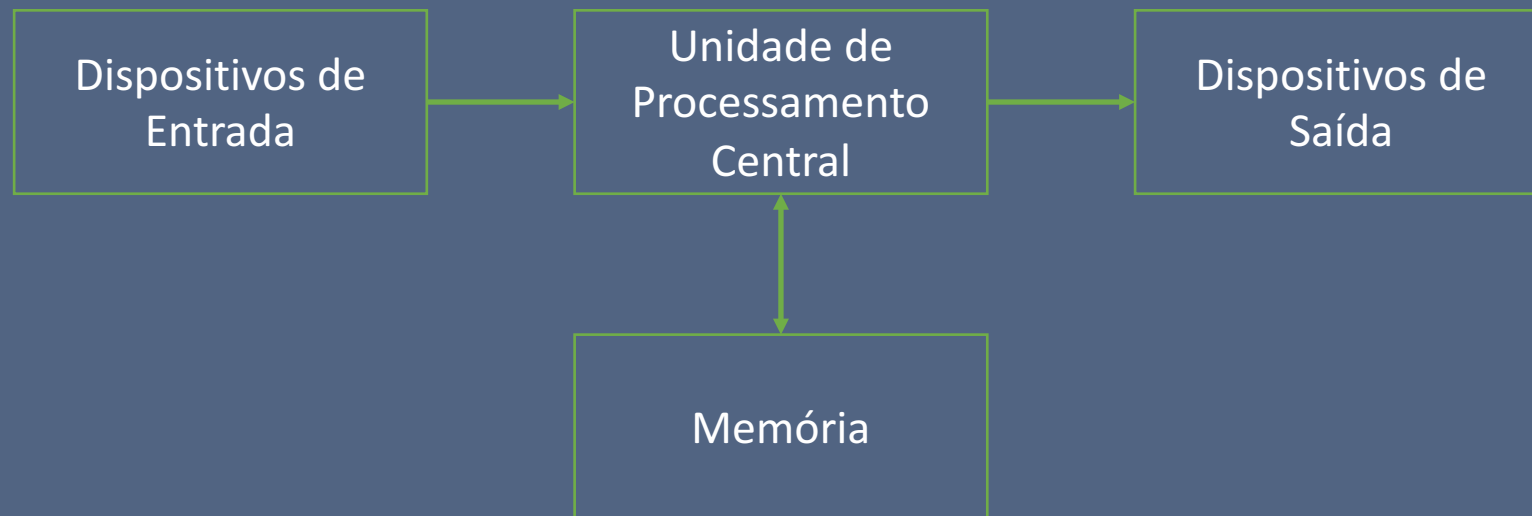
/jpdol

# Linguagem de Programação Python

- Conceitos Iniciais;
- Operações Aritméticas;
- Variáveis;
- Tipos de Dados;
- Estruturas de dados Básicas;
- Operações Lógicas;
- Estruturas Condicionais;
- Laços;
- Funções;

# Conceitos Iniciais

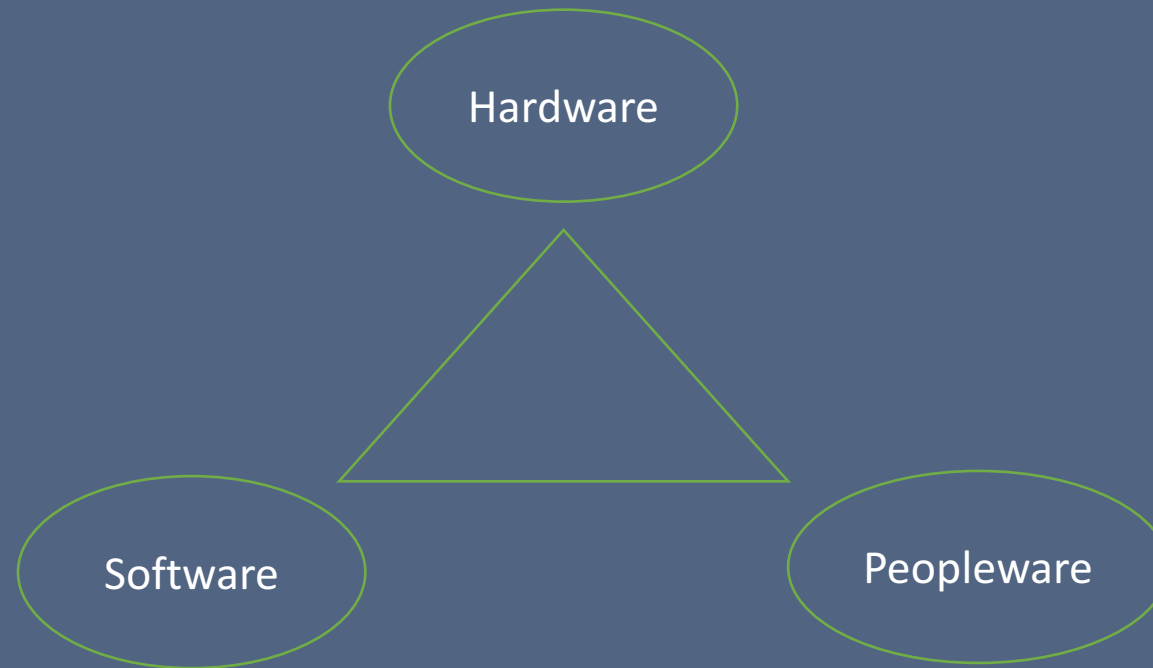
- Computador:



Arquitetura de Von Neumann

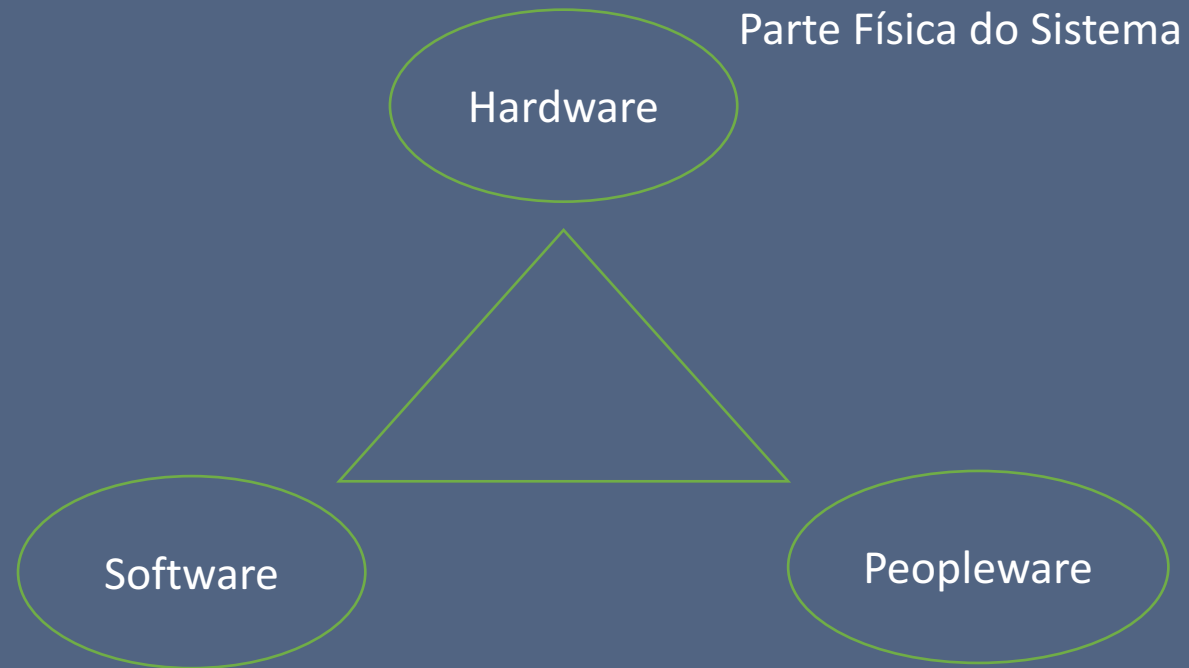
# Conceitos Iniciais

- Sistema Computacional:



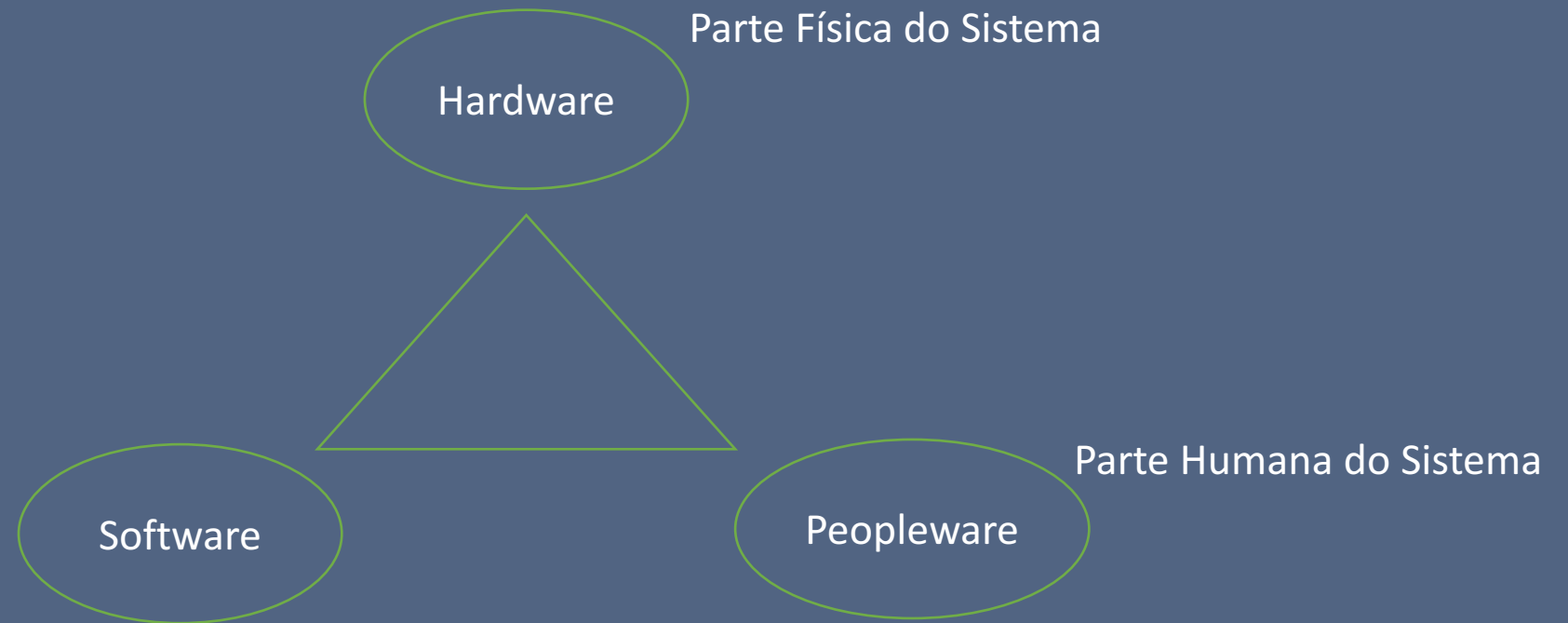
# Conceitos Iniciais

- Sistema Computacional:



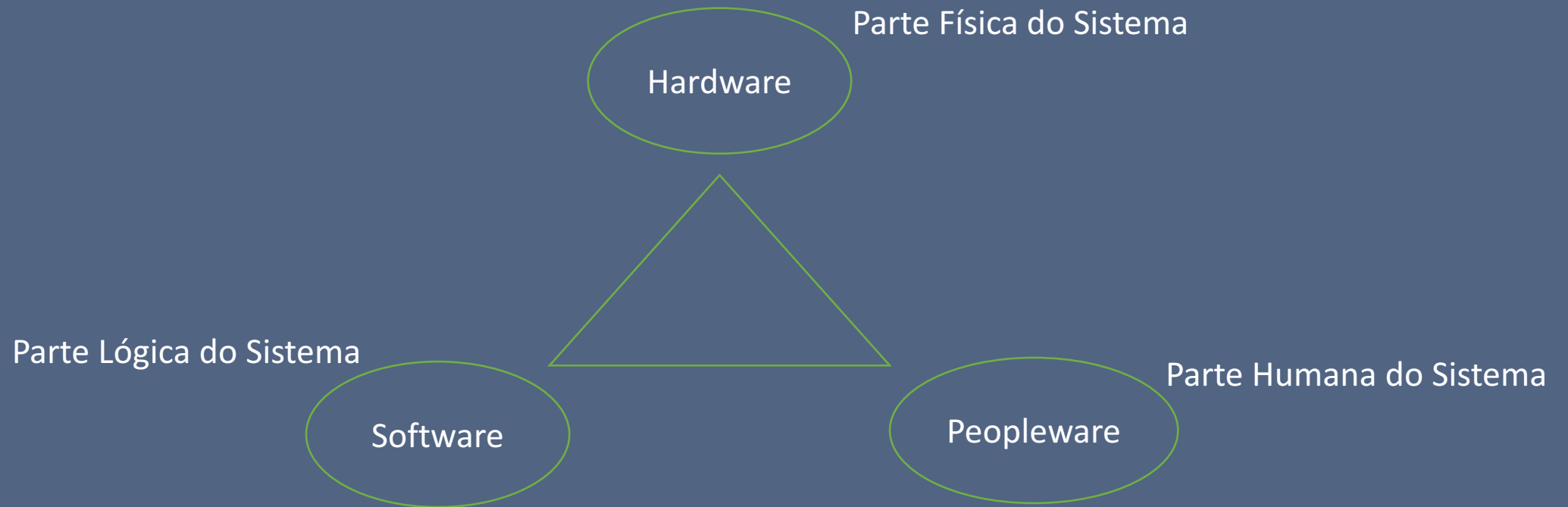
# Conceitos Iniciais

- Sistema Computacional:



# Conceitos Iniciais

- Sistema Computacional:



# Conceitos Iniciais

- Linguagem de Programação:
  - Ferramenta para Construir Programas.
- Programa:
  - Algoritmo implementado em uma LP para ser executado em um computador.
- Algoritmo:
  - Conjunto de passos para solução de um problema.



# Conceitos Iniciais

- Linguagem de Programação Python:
  - Alto Nível de Abstração
  - Linguagem Interpretada e de Script
    - Executa em VM
    - Shell executa comandos diretamente
  - Multiparadigma
    - Imperativa, Funcional, Orientada a Objetos
  - Portável, Resumida, Rápida, Flexível e Open Source.

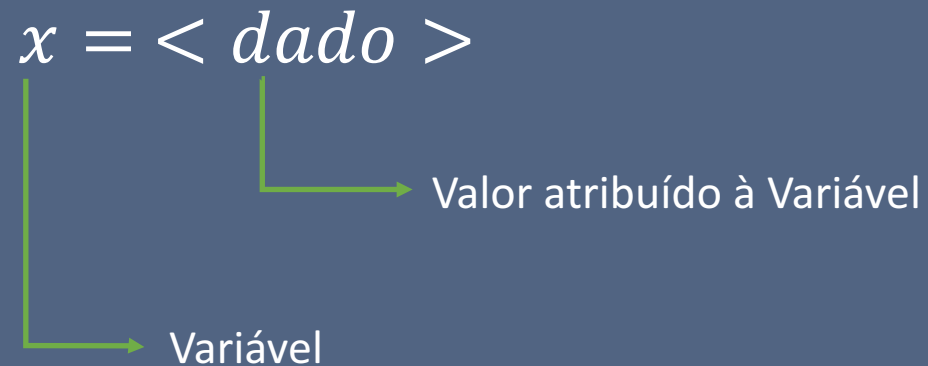


# Operações Aritméticas

- Adição:
  - $x + y$
- Subtração:
  - $x - y$
- Multiplicação:
  - $x * y$
- Divisão:
  - $x / y$
- Divisão Inteira:
  - $x // y$
- Módulo Aritmético (Resto da Divisão Inteira):
  - $x \% y$
- Valor Absoluto:
  - $abs(x)$
- Exponenciação:
  - $x ** y$

# Variáveis

- Variáveis são a abstração de endereços de memória para armazenamento de dados.



# Tipos de Dados

- Inteiro (*int*)
  - Ex: 4; -3; 98137.
- Ponto Flutuante (*float*)
  - Ex: 3.1415; 2.4124, -4.321.
- Booleano (*bool*)
  - *True*; *False*.
- Complexo (*complex*)
  - Ex: 4+3j; 2-1j; 4j.
- *String* (*str*)
  - Ex: "Python"; "Inteligência Artificial".

# Estruturas de Dados Básicas

- Lista (*list*)
  - Ex: []; [1, 2, 6]; [4, 3.2, "Maria"];.
- Tupla (*tuple*)
  - Ex: (2,4); (1,7,6,5,4).
- Dicionário (*dict*)
  - {}; {"Marca":"FIAT", "Modelo":"Siena", "Ano":2019}.

# Operações Lógicas

- Operações cujos resultados são do tipo de dado *bool*.
  - Comparação: `==`, `>=`, `<=`, `>`, `<`
  - Portas Lógicas: *and*, *or*, *not*

# Estruturas Condicionais

- Trechos do programa que são executados (ou não) dependendo de alguma condição.

```
if (<condição>):  
    <código>  
elif(<condição>):  
    <código>  
else:  
    <código>
```

# Laços (Estruturas de Repetição)

- Trechos do programa que são executados repetidas vezes.

Loop While:

```
while(<condição>):  
    <código>
```

Loop For:

```
for i in range(N):  
    <código>
```



# Funções

- Trechos do programa que normalmente são executados com frequência. Para evitar a repetição de código, isolam-se esses trechos. Cada trecho isolado é denominado Função.

```
def <nome_função> (<parâmetros>):  
    <código>  
    return <variáveis_retornadas>
```