Pilhas

Arthur Correia
Gabriel Lyrio
Gabriel Maiworm
Mathias
Nicolas
Pedro H. Francisco
Rebeca coelho

2º INFO: A 14/05/2025

Introdução

O que é: Empilhamento de dados linear

Funcionamento:

Segue o princípio LIFO (Last In, First Out)

Operações:

Push adiciona um novo elemento ao topo da pilha

Pop remove o elemento do topo e o retorna

Peek visualizar o elemento no topo da pilha sem removê-lo

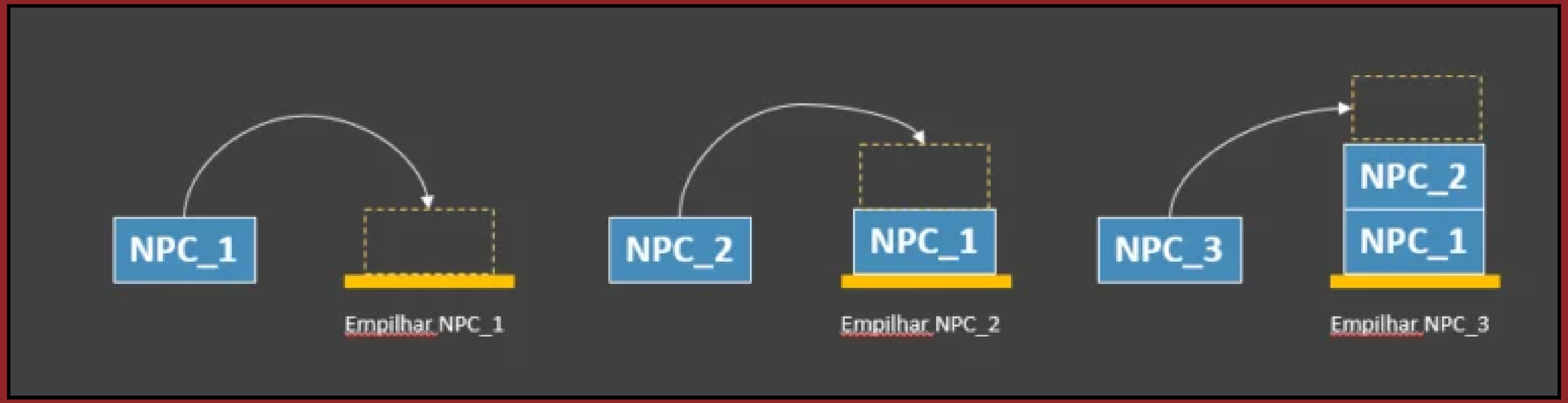
isEmpty verifica se a pilha está vazia

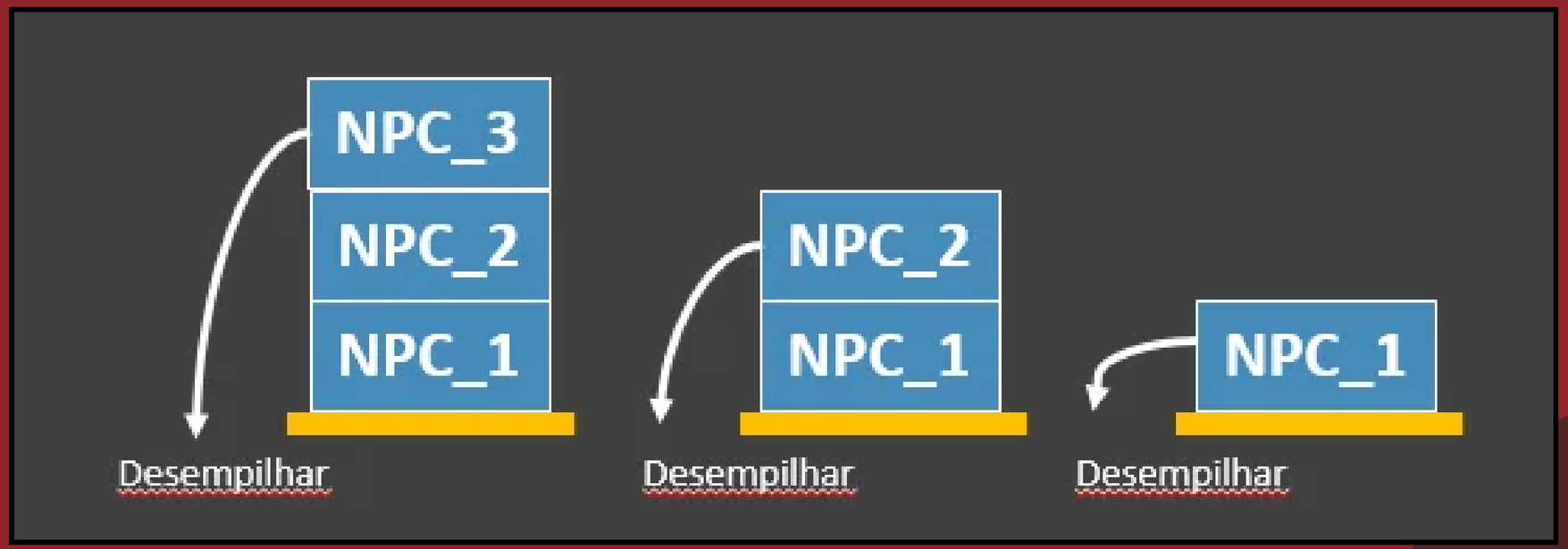
Size retorna o número de elementos na pilha

Funcionamento:

Arrays (elemento identificado por índices que indicam sua posição)

ou usando listas encadeadas (sequências de nós com 2 elementos)


NPC_1
Empilhar NPC_1
NPC_2
NPC_1
Empilhar NPC_2
NPC_3
NPC_2
NPC_1
Empilhar NPC_3


NPC_3
NPC_2
NPC_1
Desempilhar
NPC_2
NPC_1
Desempilhar
NPC_1
Desempilhar

Vantagens:

Simplicidade de implementação

Eficiência nas operações de inserção e remoção, ambas realizadas em tempo constante

Desvantagens:

Acesso restrito apenas ao elemento no topo, o que pode ser limitante para algumas aplicações.

Não permite a busca direta por elementos específicos, exigindo que todos os elementos sejam removidos até encontrar o desejado

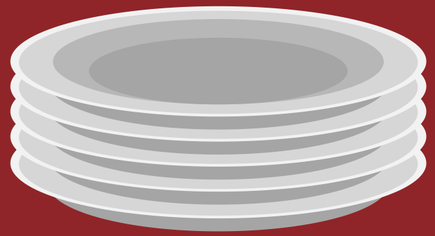
Diferença de fila x pilha:

Pilha trabalha no princípio LIFO(last in, first out) sendo o a organização que o último a entrar é o primeiro a sair, diferente da fila, que trabalha com o princípio FIFO (first in, first out) sendo a organização que o primeiro a entrar é o primeiro a sair.

Analogias

Pilha de livros

Pilha de pratos



Pilha de moedas

Histórico de navegação

pilha de copos

```python
## CODIGO 1 - Estrutura base
pilha = []

pilha.append("1")
pilha.append("2")
pilha.append("3")

print("Pilha:", pilha)

topo = pilha.pop()
print("Elemento removido:", topo)
print("Pilha após pop:", pilha)

print("Topo da pilha:", pilha[-1])
```

```
Pilha: ['1', '2', '3']
Elemento removido: 3
Pilha após pop: ['1', '2']
Topo da pilha: 2
```

```python
## CODIGO 2 - Pratos
pilha = []
pilha.append(object, /)
Append object to the end of the list.
pilha.append("Prato 3")
pilha.append("Prato 4")
pilha.append("Prato 5")

print("Pilha de pratos:", pilha)

topo = pilha.pop()
print("Prato servido:", topo)
print("Pilha após pop:", pilha)

print("Topo da pilha:", pilha[-1])
```

```
Pilha de pratos: ['Prato 1', 'Prato 2', 'Prato 3', 'Prato 4', 'Prato 5']
Prato servido: Prato 5
Pilha após pop: ['Prato 1', 'Prato 2', 'Prato 3', 'Prato 4']
Topo da pilha: Prato 4
```

```python
## CODIGO 3 - Estacionamento (com input)
pilha = []

pilha.append("Carro 1")
pilha.append("Carro 2")
pilha.append("Carro 3")
pilha.append("Carro 4")

print("Estacionamento:", pilha)
novo = input("Carro novo:")
topo = pilha.append(novo)
print("Estacionamento atual:", pilha)

print("Carro novo:", pilha[-1])
```

```
Estacionamento: ['Carro 1', 'Carro 2', 'Carro 3', 'Carro 4']
Carro novo:Carro 5
Estacionamento atual: ['Carro 1', 'Carro 2', 'Carro 3', 'Carro 4', 'Carro 5']
Carro novo: Carro 5
```

```python
##CODIGO 4 - Carros
pilha = []
pilha.append("Carro Verde")
pilha.append("Carro Azul")
pilha.append("Carro Amarelo")
pilha.append("Carro Branco")
def remver_carro(carro, pilha):
    if carro in pilha:
        pilha.remove(carro)
        print(f"{carro} removido da pilha.")
    else:
        print(f"{carro} não encontrado na pilha.")

print("Pilha de carros:", pilha)
carro_remover = input("Digite o carro a ser removido: ")
remver_carro(carro_remover, pilha)
print("Pilha atualizada:", pilha)
```

```
Pilha de carros: ['Carro Verde', 'Carro Azul', 'Carro Amarelo', 'Carro Branco']
```

```python
##CODIGO 5 - P.O.O. (Programação Orientada a Objeto)

class Pilha:
    def __init__(self):
        self._itens = []

    def push(self, item):
        self._itens.append(item)
        print(f"🔼 Inserido: {item}")

    def pop(self):
        if self.esta_vazia():
            print("⚠️ Pilha vazia. Nada a remover.")
            return None
        removido = self._itens.pop()
        print(f"🔽 Removido: {removido}")
        return removido

    def peek(self):
        if self.esta_vazia():
            print("👀 Pilha vazia.")
            return None
        topo = self._itens[-1]
        print(f"👁️ Topo da pilha: {topo}")
        return topo

    def esta_vazia(self):
        return len(self._itens) == 0

    def exibir(self):
        print("📦 Pilha atual (topo → base):")
        for item in reversed(self._itens):
            print(f"   - {item}")

# Demonstração
carros = Pilha()
carros.push("Carro Verde")
carros.push("Carro Rosa")
carros.push("Carro Azul")
carros.push("Carro Preto")
carros.push("Carro Branco")
carros.exibir()
carros.pop()  # Remove o último inserido (Carro Branco)
carros.peek()  # Mostra o novo topo (Carro Preto)
carros.exibir()
```

```
🔼 Inserido: Carro Verde
🔼 Inserido: Carro Rosa
🔼 Inserido: Carro Azul
🔼 Inserido: Carro Preto
🔼 Inserido: Carro Branco
📦 Pilha atual (topo → base):
   - Carro Branco
   - Carro Preto
   - Carro Azul
   - Carro Rosa
   - Carro Verde
🔽 Removido: Carro Branco
👁️ Topo da pilha: Carro Preto
📦 Pilha atual (topo → base):
   - Carro Preto
   - Carro Azul
   - Carro Rosa
   - Carro Verde
```

```python
##CODIGO 6 - Refrigerantes
from typing import List, Optional

class PilhaRefrigerantes:
    def __init__(self) -> None:
        self._itens: List[str] = []

    def push(self, refrigerante: str) -> None:
        self._itens.append(refrigerante)

    def pop(self) -> Optional[str]:
        return self._itens.pop() if not self.esta_vazia() else None

    def peek(self) -> Optional[str]:
        return self._itens[-1] if not self.esta_vazia() else None

    def esta_vazia(self) -> bool:
        return len(self._itens) == 0

    def __str__(self) -> str:
        if self.esta_vazia():
            return "🧃 Pilha de refrigerantes está vazia."
        return "🧃 Pilha de refrigerantes (topo → base):\n" + "\n".join(
            f"   - {item}" for item in reversed(self._itens)
        )

# Demonstração
if __name__ == "__main__":
    pilha = PilhaRefrigerantes()
    for refri in ["Fanta Uva", "Sprite", "Coca-Cola", "Pepsi", "Guaraná"]:
        pilha.push(refri)
```

```python
# Demonstração
if __name__ == "__main__":
    pilha = PilhaRefrigerantes()
    for refri in ["Fanta Uva", "Sprite", "Coca-Cola", "Pepsi", "Guaraná"]:
        pilha.push(refri)

    print(pilha)  # Exibe pilha
    removido = pilha.pop()
    print(f"\n🔽 Removido: {removido}")
    print(f"👁️ Topo atual: {pilha.peek()}\n")
    print(pilha)
```

```
🧃 Pilha de refrigerantes (topo → base):
  - Guaraná
  - Pepsi
  - Coca-Cola
  - Sprite
  - Fanta Uva

🔽 Removido: Guaraná
👁️ Topo atual: Pepsi

🧃 Pilha de refrigerantes (topo → base):
  - Pepsi
  - Coca-Cola
  - Sprite
  - Fanta Uva
```

```
[ ]  ## CODIGO 7 - copos
     pilha = []

     pilha.append("Copo 1")
     pilha.append("Copo 2")
     pilha.append("Copo 3")
     pilha.append("Copo 4")
     pilha.append("Copo 5")

     print("Pilha de Copos:", pilha)

     topo = pilha.pop()
     print("Copo retirado:", topo)
     print("Pilha de copos:", pilha)

     print("Topo da pilha:", pilha[-1])
```

```
Pilha de Copos: ['Copo 1', 'Copo 2', 'Copo 3', 'Copo 4', 'Copo 5']
Copo cheio: Copo 5
Pilha de copos: ['Copo 1', 'Copo 2', 'Copo 3', 'Copo 4']
Topo da pilha: Copo 4
```

```python
## CODIGO 8 - compras
pilha = []

pilha.append("Moeda 1")
pilha.append("Moeda 2")
pilha.append("Moeda 3")
pilha.append("Moeda 4")
pilha.append("Moeda 5")

print("Pilha:", pilha)

topo = pilha.pop()
print("moeda retirada:", topo)
print("Pilha após pop:", pilha)

print("Topo da pilha:", pilha[-1])
```

```
Pilha: ['Moeda 1', 'Moeda 2', 'Moeda 3', 'Moeda 4', 'Moeda 5']
moeda retirada: Moeda 5
Pilha após pop: ['Moeda 1', 'Moeda 2', 'Moeda 3', 'Moeda 4']
Topo da pilha: Moeda 4
```

```python
## CODIGO 9 - read dead redemption
pilha = []

pilha.append("read dead redemption 1")
pilha.append("read dead redemption 2")

print("temporadas:", pilha)
while pilha:
    topo = pilha.pop()
    print("Elemento removido:", topo)
    print("read dead redemption após pop:", pilha)
    if pilha:
        print("Topo da pilha:", pilha[-1])
```

```
temporadas: ['read dead redemption 1', 'read dead redemption 2']
Elemento removido: read dead redemption 2
read dead redemption após pop: ['read dead redemption 1']
Topo da pilha: read dead redemption 1
Elemento removido: read dead redemption 1
read dead redemption após pop: []
```

```python
## CODIGO 10 - ordem cronológica
pilha = []

pilha.append("the big bang theory")
pilha.append("young sheldon")

print("temporadas:", pilha)
while pilha:
    topo = pilha.pop()
    print("Elemento removido:", topo)
    print("ordem cronológica após pop:", pilha)
    if pilha:
        print("Topo da pilha:", pilha[-1])
```

```
temporadas: ['the big bang theory', 'young sheldon']
Elemento removido: young sheldon
ordem cronológica após pop: ['the big bang theory']
Topo da pilha: the big bang theory
Elemento removido: the big bang theory
ordem cronológica após pop: []
```

Colab

Muito Obrigado