

Educação Profissional Paulista

Técnico em
**Ciência de
Dados**

Lógica de programação e algoritmos

Pilhas e filas

Aula 1

Código da aula: [DADOS]ANO1C3B4S28A1

Lógica de
programação e
algoritmos

Mapa da Unidade 1 Componente 4

Prática de busca e
ordenação

semana

25

Algoritmos de
contagem e
acumulação

semana

27

semana

21

Recursividade

semana

23

Busca e ordenação

semana

28

Você está aqui!
Pilhas e filas



Lógica de
programação e
algoritmos

Mapa da Unidade 1 Componente 4

Você está aqui!

Pilhas e filas

Aula 1

Código da aula: [DADOS]ANO1C3B4S28A1

28



Objetivos da aula

- Introduzir os fundamentos das estruturas de dados pilhas e filas.



Recursos didáticos

- Recurso audiovisual para exibição de vídeos e imagens;
- Acesso ao laboratório de informática e/ou internet.



Duração da aula

50 minutos.



Competências técnicas

- Identificar e resolver problemas relacionados a dados e análises;
- Compreender e dominar técnicas de manipulação de dados.



Competências socioemocionais

- Adaptar-se a novas tecnologias, técnicas e tendências sem perder o foco, as metas e os objetivos da organização;
- Colaborar efetivamente com outros profissionais, como cientistas de dados e engenheiros de dados; trabalhar em equipes multifuncionais colaborando com colegas, gestores e clientes.

LIFO



Last in, first out

FIFO



First in, first out

Primeiras ideias

Analizando a imagem anterior, responda:

O que é uma pilha em termos de estrutura de dados?

O que é uma fila em termos de estrutura de dados?

Onde você pode encontrar exemplos de pilhas e filas no dia a dia?

Ponto de partida

Imagine que você está gerenciando uma pequena lanchonete. Durante o horário de pico, você precisa organizar o uso das bandejas de forma eficiente.

Algumas bandejas devem ser atendidas na ordem em que chegam (como clientes na fila do caixa), enquanto outras precisam ser organizadas de modo que a última bandeja colocada seja a primeira a ser retirada para uso ou lavagem.

Responda as questões ao lado:

- 1. Como você organizaria os pedidos dos clientes para garantir que todos sejam atendidos de forma justa e eficiente?**
- 2. Quais métodos ou ferramentas você usaria para garantir que o preparo dos sanduíches siga a ordem correta dos ingredientes?**

Definição de pilhas (stacks)

Uma pilha é uma estrutura de dados abstrata que segue o princípio LIFO (*last in, first out*). Isso significa que o último elemento inserido na pilha será o primeiro a ser removido. As operações principais realizadas em uma pilha são:

- ▶ **push:** adiciona um elemento ao topo da pilha.
- ▶ **pop:** remove o elemento do topo da pilha.
- ▶ **peek/top:** retorna o elemento no topo da pilha sem removê-lo.
- ▶ **isEmpty:** verifica se a pilha está vazia.

Veja a seguir alguns exemplos de aplicação de pilhas.

Construindo o **conceito**



Navegadores de internet

Pilhas são usadas para gerenciar a navegação de páginas (função de voltar).



Desfazer e refazer em editores de texto

Pilhas são usadas para implementar a funcionalidade de desfazer e refazer.



Avaliação de expressões

Pilhas são usadas na avaliação de expressões matemáticas e na conversão de notações infixa para pós-fixa.

Imagens © Getty Images

Definição de filas (queues)

Uma fila é uma estrutura de dados abstrata que segue o princípio FIFO (*first in, first out*). Isso significa que o primeiro elemento inserido na fila será o primeiro a ser removido. As operações principais realizadas em uma fila são:

- ▶ **enqueue:** adiciona um elemento ao final da fila.
- ▶ **dequeue:** remove o elemento do início da fila.
- ▶ **front:** retorna o elemento no início da fila sem removê-lo.
- ▶ **isEmpty:** verifica se a fila está vazia.

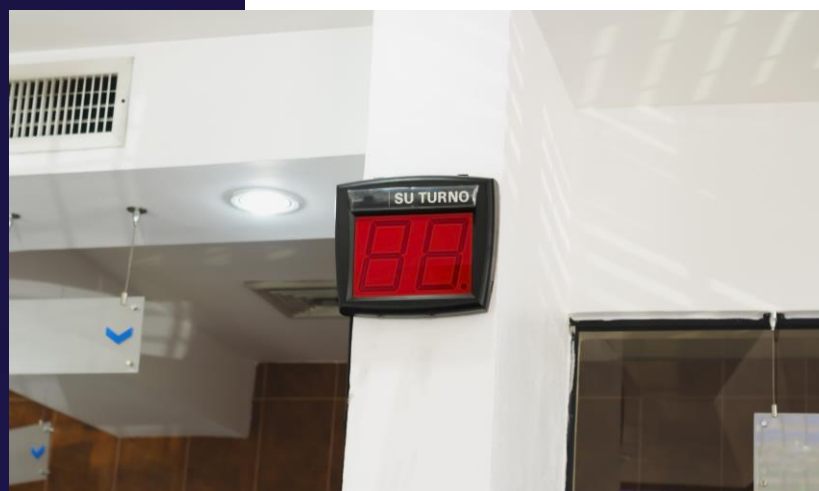
Veja a seguir alguns exemplos de aplicação de filas.

Construindo o **conceito**



Impressora

Filas são usadas para gerenciar a ordem de impressão dos documentos.



Sistemas de atendimento

Filas são usadas para gerenciar a ordem de atendimento de clientes em bancos e hospitais.



Processamento de tarefas

Filas são usadas para gerenciar tarefas em sistemas operacionais, como a fila de processos prontos para execução.

Imagens © Getty Images

Construindo o **conceito**

Exemplos práticos

Vamos praticar!

Implemente esse código utilizando Python.

Exemplo 1: pilha (stack)

Vamos criar uma classe Stack em Python que implementa as operações básicas de uma pilha: push, pop, peek e isEmpty.

```
class Stack:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return len(self.items) == 0

    def push(self, item):
        self.items.append(item)

    def pop(self):
        if not self.isEmpty():
            return self.items.pop()
        else:
            return None

    def peek(self):
        if not self.isEmpty():
            return self.items[-1]
        else:
            return None

    def size(self):
        return len(self.items)
```


Construindo
o **conceito**

Exemplos práticos

Vamos praticar!

Implemente esse código utilizando Python.

Exemplo 1: pilha (stack)

Vamos criar uma classe stack em Python que implementa as operações básicas de uma pilha: push, pop, peek e isEmpty.

```
# Exemplo de uso
stack = Stack()
stack.push(1)
stack.push(2)
stack.push(3)
print("Pilha após pushes:", stack.items)
print("Elemento no topo (peek):", stack.peek())
print("Elemento removido (pop):", stack.pop())
print("Pilha após pop:", stack.items)
print("A pilha está vazia?", stack.isEmpty())
```

Construindo o **conceito**

Exemplos práticos

Vamos praticar!

Implemente esse código utilizando Python.

Exemplo 1: fila (queue)

Vamos criar uma classe queue em Python que implementa as operações básicas de uma fila: enqueue, dequeue, front e isEmpty.

```
class Queue:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return len(self.items) == 0

    def enqueue(self, item):
        self.items.append(item)

    def dequeue(self):
        if not self.isEmpty():
            return self.items.pop(0)
        else:
            return None

    def front(self):
        if not self.isEmpty():
            return self.items[0]
        else:
            return None

    def size(self):
        return len(self.items)
```

Construindo
o **conceito**

Exemplos práticos

Vamos praticar!

Implemente esse código utilizando Python.

Exemplo 2: fila (queue)

Vamos criar uma classe queue em Python que implementa as operações básicas de uma fila: enqueue, dequeue, front e isEmpty.

```
# Exemplo de uso
queue = Queue()
queue.enqueue(1)
queue.enqueue(2)
queue.enqueue(3)
print("Fila após enqueues:", queue.items)
print("Elemento na frente (front):", queue.front())
print("Elemento removido (dequeue):", queue.dequeue())
print("Fila após dequeue:", queue.items)
print("A fila está vazia?", queue.isEmpty())
```




Vamos
fazer um
quiz

**Qual operação adiciona um
elemento ao topo da pilha?**

Push

Pop

Enqueue

Dequeue



Vamos
fazer um
quiz

Qual princípio as filas seguem?

LIFO

FILO

FIFO

LILO



Vamos
fazer um
quiz

**Qual operação remove um
elemento do início da fila?**

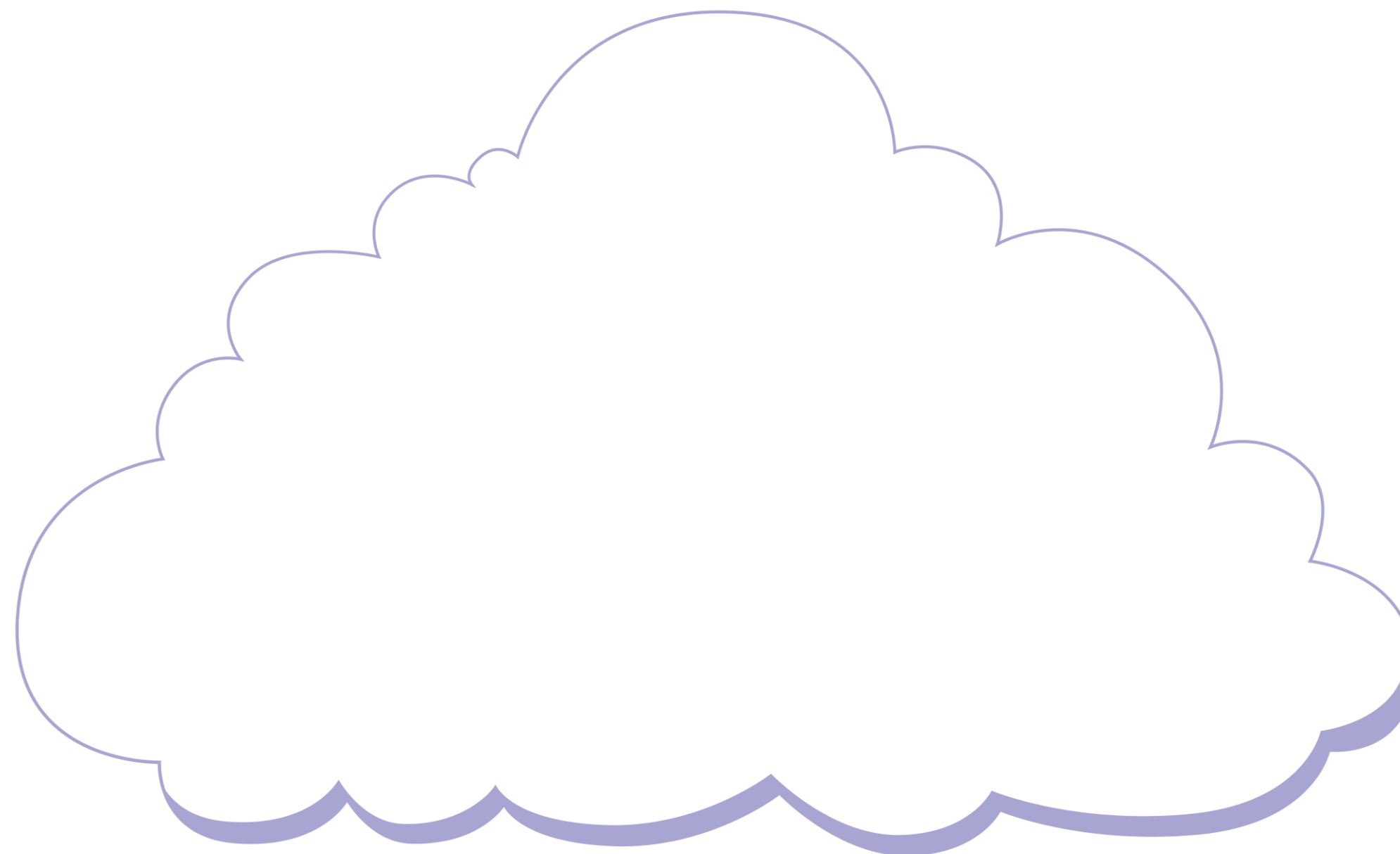
Push

Pop

Enqueue

Dequeue

Nuvem de palavras



© Getty Images

O que nós
**aprendemos
hoje?**



© Getty Images

O que nós
aprendemos
hoje?

Então ficamos assim...

- 1** Na primeira aula, exploramos os conceitos fundamentais de pilhas e filas, suas definições e princípios de funcionamento. Pilhas seguem o princípio LIFO (*last in, first out*), enquanto filas seguem o princípio FIFO (*first in, first out*).
- 2** Discutimos as operações básicas de pilhas, como push, pop e peek, e de filas, como enqueue, dequeue e front. Cada operação foi detalhadamente explicada com exemplos práticos em Python.
- 3** Exemplificamos aplicações reais dessas estruturas de dados, como gerenciamento de histórico em navegadores e sistemas de impressão. Além disso, incluímos um quiz interativo para reforçar o aprendizado dos conceitos abordados.

Saiba mais

Há tantos termos quando falamos de programação que é até difícil de memorizar tudo, não é mesmo?

Mas este artigo pode ajudar e muito! Veja o glossário de Back-end e nunca mais se perca nos nomes e siglas!

OLIVEIRA, A. Glossário Back-end. Alura, 11 ago. 2022. Disponível em: <https://www.alura.com.br/artigos/glossario-back-end>. Acesso em: 24 jul. 2024.

Referências da aula

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. **Lógica de programação**: a construção de algoritmos e estruturas de dados com aplicações em Python. Porto Alegre: Bookman, 2022.

Identidade visual: imagens © Getty Images.

Educação Profissional Paulista

Técnico em
**Ciência de
Dados**