



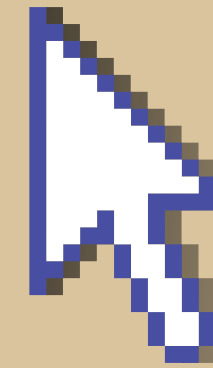
2º INFO B

# ESTRUTURA DE DADOS - FILAS

FABRÍCIO DA SILVA, GUILHERME SANCHES,  
GIOVANNA FELISARDO, JOÃO ACCÁCIO, JOÃO  
GABRIEL, LETÍCIA TEIXEIRA E LUCAS BENTO.



FIRJAN SENAI SESI 2025





# O QUE É ESTRUTURA DE DADOS?

- É a maneira como os dados são organizados, gerenciados e armazenados em um sistema.
- Ela permite que os programadores manipulem dados de forma eficiente, facilitando operações como inserção, exclusão, busca e atualização.
- Escolher a estrutura ideal pode impactar significativamente no desempenho de um algoritmo.





# CATEGORIAS



- Existem diversas categorias de estruturas de dados, essas podem ser classificadas em:

1. estruturas primárias que incluem tipos básicos como arrays, listas, pilhas e filas, que são fundamentais para a construção de algoritmos.
2. estruturas secundárias, como árvores, grafos e tabelas hash, são mais complexas e permitem a representação de dados em formatos mais sofisticados, adequados para aplicações específicas.





# O QUE SÃO FILAS?

Pilhas e filas são estruturas de dados que seguem princípios **diferentes** de organização. A pilha opera no princípio LIFO (Last In, First Out), onde o último elemento adicionado é o primeiro a ser removido. Isso é útil em situações como a execução de funções recursivas. As filas, por sua vez, seguem o princípio FIFO (First In, First Out), permitindo que o primeiro elemento adicionado seja o primeiro a ser removido, sendo ideal para gerenciamento de tarefas e processos em sistemas operacionais.





2º INFO B

# ORGANIZAÇÕES DE FILAS



## FIFO

FIFO: Os primeiros itens são os que saem primeiro

EX:

```
import queue
```

```
q1 = queue.Queue()
```



## LIFO

LIFO: Os itens mais recentes são os que saem primeiro

EX:

```
import queue
```

```
q1 = queue.LifoQueue()
```



FIRJAN SENAI SESI 2025



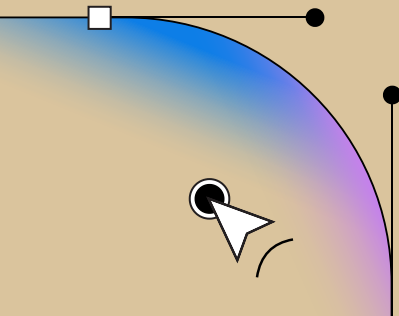


2º INFO B



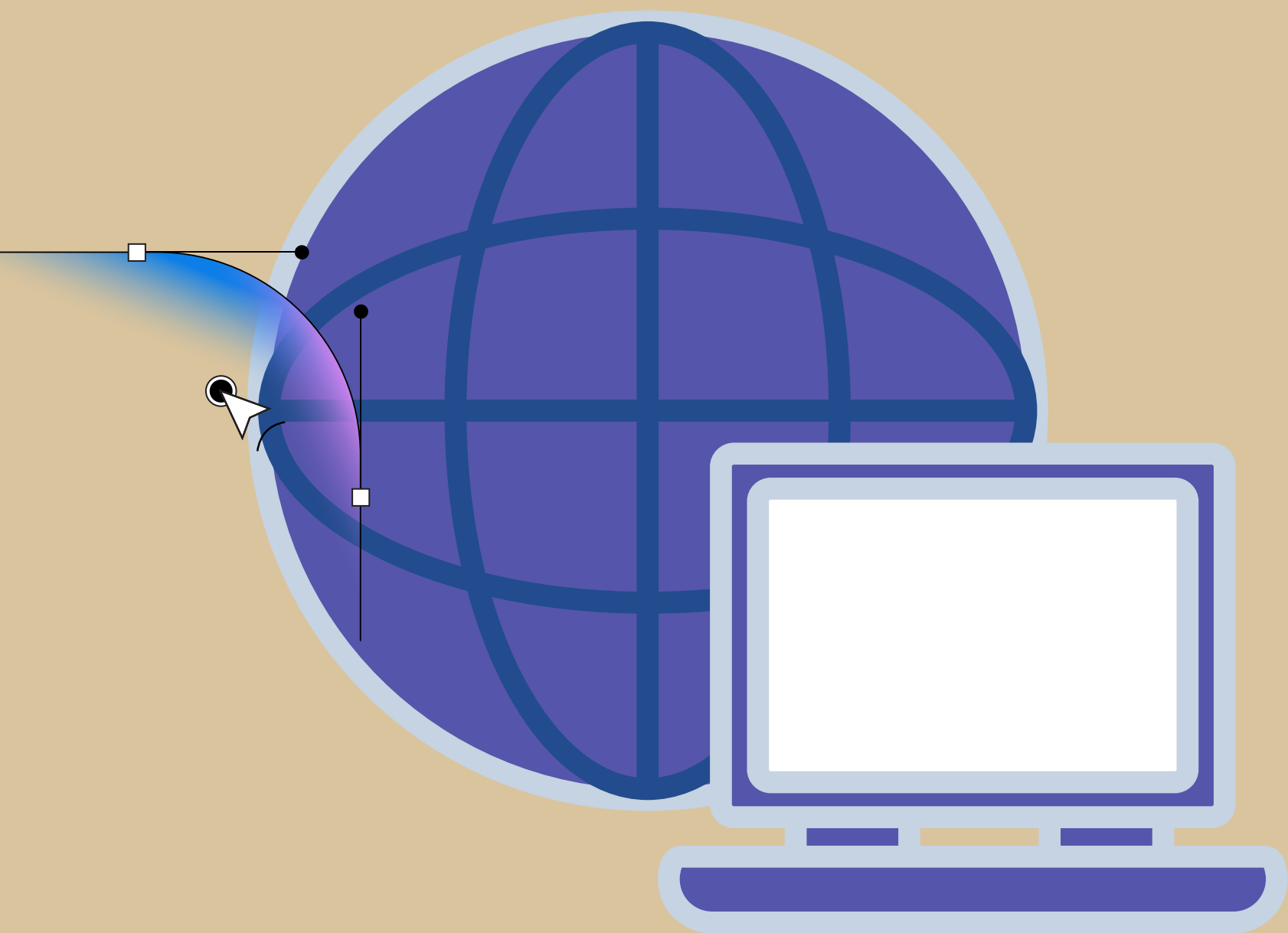
# ORGANIZAÇÕES DE FILAS

X	FIFO	LIFO
Ordem de Remoção	O Primeiro a chegar sai primeiro	O Último a chegar sai primeiro
Exemplo Prático	Fila de Banco	CTRL-Z
Uso em FILAS	Comum e Direto	Lógica Adaptada



FIRJAN SENAI SESI 2025





# ANALOGIAS

## Fila de banco

Quando você chega em um banco e entra na fila para ser atendido, você espera a sua vez conforme a ordem de chegada. Assim como na estrutura de dados fila, quem entrou primeiro na fila será o primeiro a ser atendido (removido da fila).

## Fila no drive-thru

No drive-thru de um fast food, os carros são atendidos na sequência em que chegam. O primeiro carro que chega é o primeiro a ser servido e sair, enquanto os demais aguardam atrás.





2° INFO B

## Fila de impressão

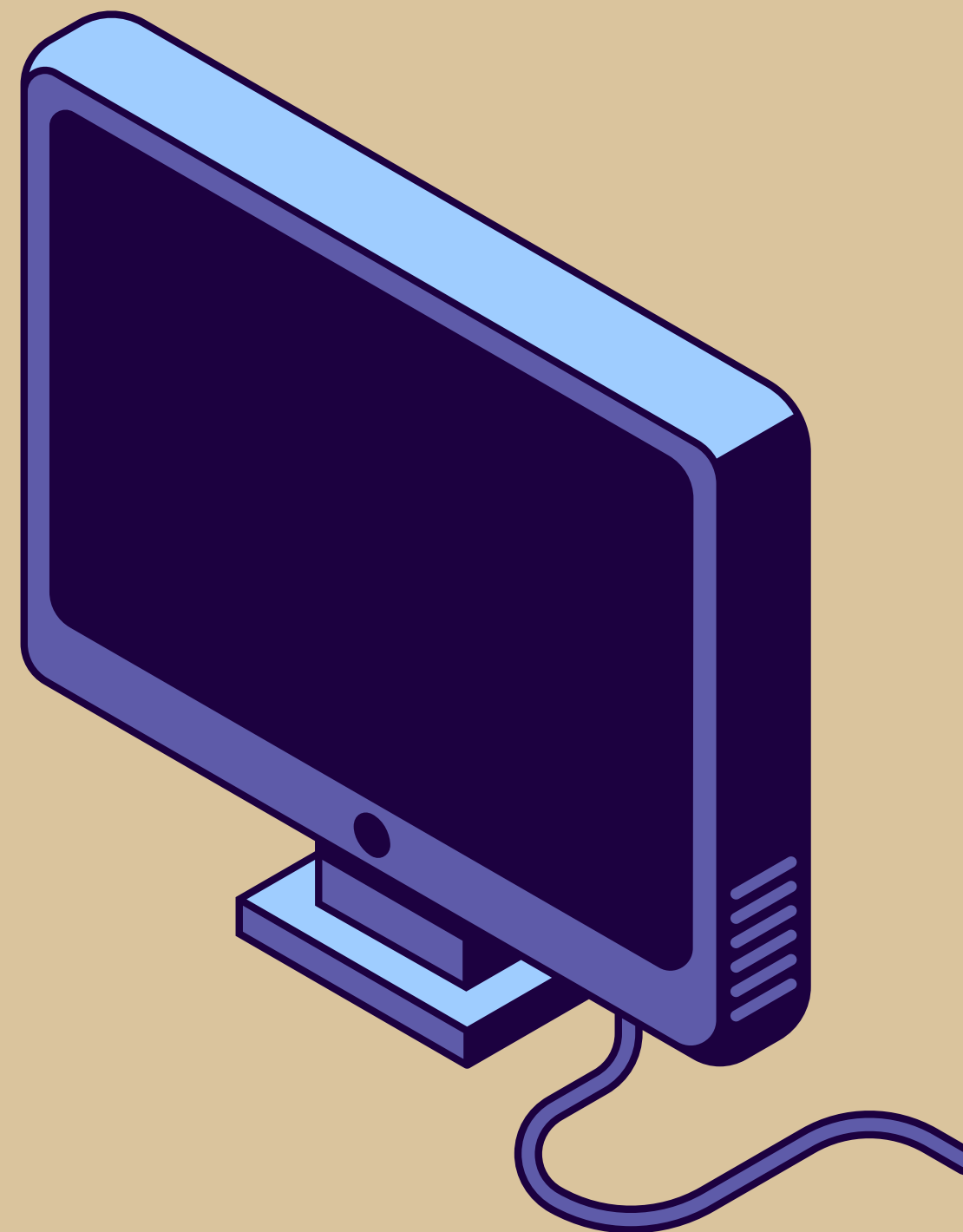
Quando vários documentos são enviados para a impressora, eles são impressos na ordem em que foram enviados.

O primeiro documento enviado (enqueue) será o primeiro a ser impresso (dequeue).

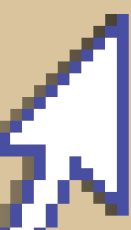
## Fila de ônibus

As pessoas esperam na fila do ponto de ônibus, e entram conforme a ordem de chegada.

O primeiro passageiro da fila é o primeiro a embarcar.



FIRJAN SENAI SESI 2025







2º INFO B

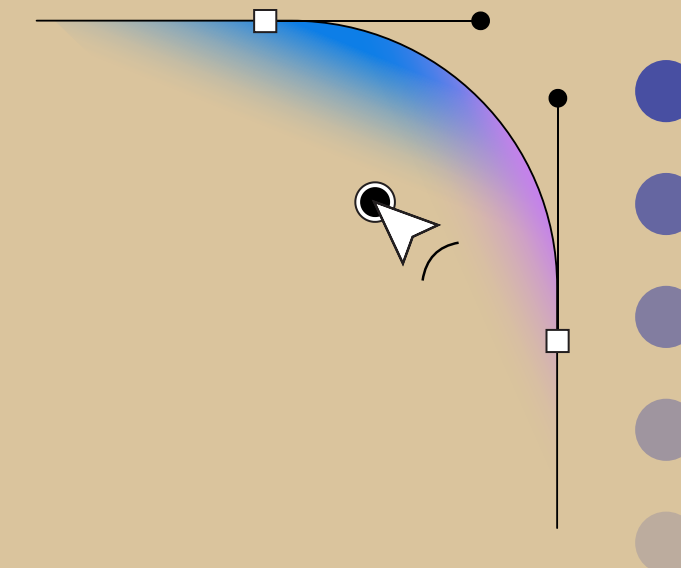


## Atendimento telefônico automatizado

Ao ligar para uma central de atendimento, você entra em uma fila de espera. A chamada é atendida conforme a ordem de entrada. As chamadas são enfileiradas e atendidas uma por vez, seguindo a ordem de chegada.



FIRJAN SENAI SESI 2025



# CÓDIGOS DE FIFO

```
from collections import deque
fb = deque()
fb.append(1)
fb.append(2)
fb.append(3)
print("fila de atendimento:", list(fb))
while fb:
    cliente_atendido = fb.popleft()
    print(f"Atendendo {cliente_atendido}")
```

Saída:  
fila de atendimento: [1, 2, 3]  
Atendendo 1  
Atendendo 2  
Atendendo 3

## Função do código acima:

Esse código permite o usuário visualizar uma fila e o atendimento de cada “componente da fila” pela ordem de chegada.

# CÓDIGOS FIFO

```
from collections import deque
nomes = deque()
i = 1
while True:
    nome = input(f"Insira o nome do aluno {i}: ")
    nomes.append(nome)
    if nome == "Acabou":
        break
    i += 1

j = 1
print("Primeiro:", nomes.popleft())

while len(nomes) > 1:
    j += 1
    pessoa = nomes.popleft()
    print(f"{j}: {pessoa}")
```

Saída:

Insira o nome do aluno 1: Accácio  
Insira o nome do aluno 2: João  
Gabriel Insira o nome do aluno  
3: Lucas Insira o nome do aluno  
4: Guilherme Insira o nome do  
aluno  
5: Giovanna Insira o nome do  
aluno  
6: Fabrício Insira o nome do  
aluno  
7: Letícia Insira o nome do aluno  
8: Acabou Insira o nome do  
aluno  
9: agdjcabou  
Primeiro: Accácio 2: João Gabriel  
3: Lucas 4: Guilherme 5:  
Giovanna 6: Fabrício 7: Letícia 8:  
Acabou

## Função do código acima:

Esse código permite o usuário adicionar nomes em uma lista, e no final ele imprime a lista por ordem de chegada





```
from collections import deque
fila = deque()
fila.append("A")
fila.append("B")
fila.append("C")
print(fila.popleft())
print(fila.popleft())
print(fila.popleft())
```

Saída:

A  
B  
C

```
fila = ()
fila.append("A")
fila.append("B")
fila.append("C")
print(fila.pop(0))
print(fila.pop(0))
print(fila)
```

Saída:

A  
B  
['C']



# CÓDIGOS FIFO

```
from collections import deque

fila = deque()

while True:
    comando = input("Digite um nome para entrar na fila, 'atender' para chamar alguém ou 'sair': ").strip()

    if comando.lower() == "sair":
        break
    elif comando.lower() == "atender":
        if fila:
            pessoa = fila.popleft()
            print(f"{pessoa} foi atendido.")
        else:
            print("Fila vazia!")
    else:
        fila.append(comando)
        print(f"{comando} entrou na fila.")

print("Programa encerrado.")
```

Saída:

Digite um nome para entrar na fila, 'atender' para chamar alguém ou 'sair': **atender**

Fila vazia!

Digite um nome para entrar na fila, 'atender' para chamar alguém ou 'sair': **giovanna**

giovanna entrou na fila.

Digite um nome para entrar na fila, 'atender' para chamar alguém ou 'sair': **sair**

Programa encerrado.



# CÓDIGOS LIFO

```
from collections import  
  
dequehistorico = []  
  
historico.append("Escreveu: Olá")  
historico.append("Escreveu:  
Mundo")  
historico.append("Deletou:  
Mundo")  
  
print("Histórico atual:", historico)  
  
ultima_acao = historico.pop()  
print("Desfez:", ultima_acao)  
  
print("Histórico após desfazer:",  
historico)
```

Saída:  
Histórico atual: ['Escreveu:  
Olá', 'Escreveu: Mundo',  
'Deletou: Mundo']

Desfez: Deletou: Mundo

Histórico após desfazer:  
['Escreveu: Olá', 'Escreveu:  
Mundo']



# CÓDIGOS LIFO

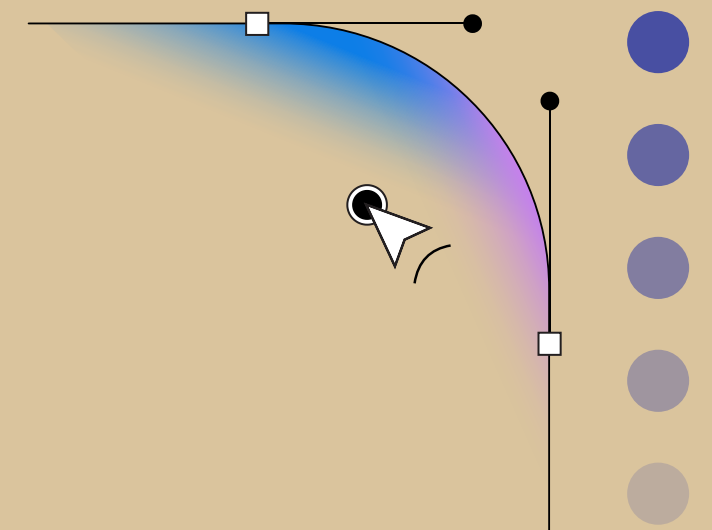
```
fila = []  
fila.append('A')  
fila.append('B')  
fila.append('C')  
print(fila.pop())  
print(fila.pop())  
print(fila.pop())
```

Saída:

C  
B  
A

## Função do código acima:

Esse código permite o usuário visualizar a saída de uma fila de trás para frente, ou seja, o componente mais recente é o componente que é executado primeiro.



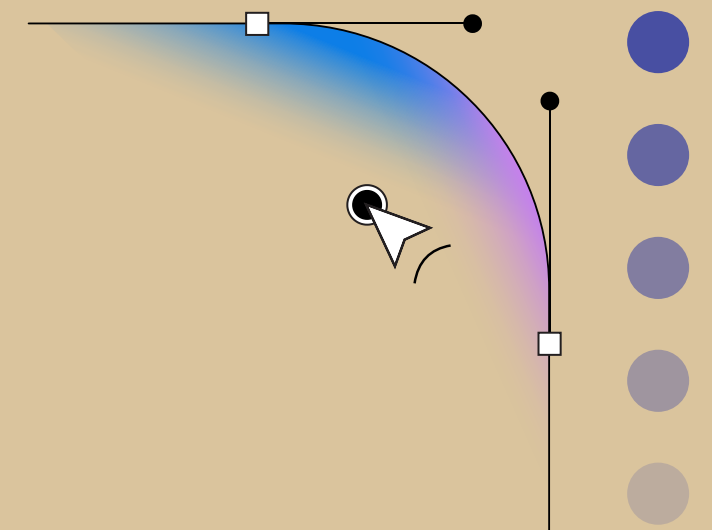
# CÓDIGOS LIFO

```
from collections import deque
q1 = deque()
q2 = deque()
q2.append(10)
while q1:
    q2.append(q1.popleft())
q1, q2 = q2, q1
print(q1.popleft())
```

Saída:  
10

## Função do código acima:

Esse código permite o usuário visualizar a saída de uma fila de trás para frente, ou seja, o componente mais recente é o componente que é executado primeiro.





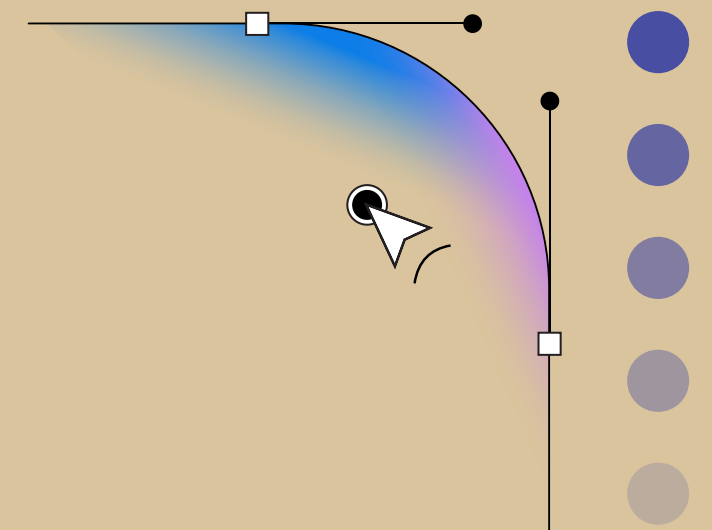
# CÓDIGOS LIFO

```
from collections import  
deque  
fila = deque()  
fila.append(1)  
fila.append(2)  
fila.append(3)  
for _ in range(len(fila) - 1):  
    fila.append(fila.popleft())  
print(fila.popleft())
```

Saída:  
3

## Função do código acima:

Esse código permite o usuário visualizar a saída de uma fila de trás para frente, ou seja, o componente mais recente é o componente que é executado primeiro.



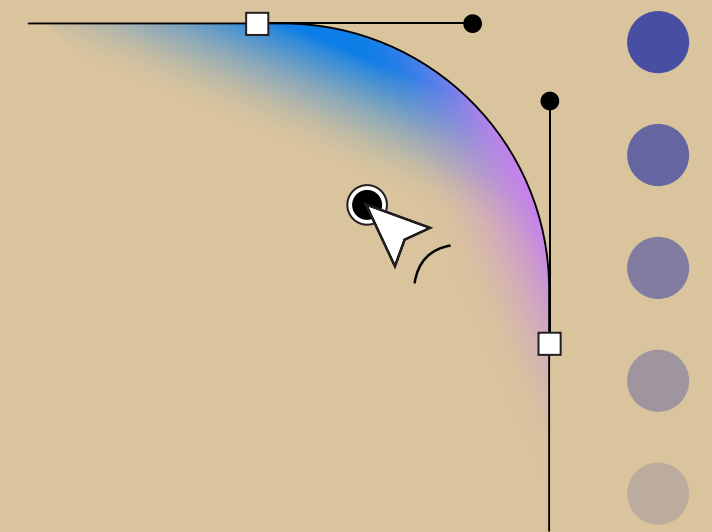
# CÓDIGOS LIFO

```
fila = []  
  
fila.append('A')  
fila.append('B')  
  
fila.append(fila.pop(0))  
  
print(fila.pop(0))
```

Saída:  
B

## Função do código acima:

Esse código permite o usuário visualizar a saída de uma fila de trás para frente, ou seja, o componente mais recente é o componente que é executado primeiro.



**TO BE CONTINUE...**

