



CMP1054 – ED I

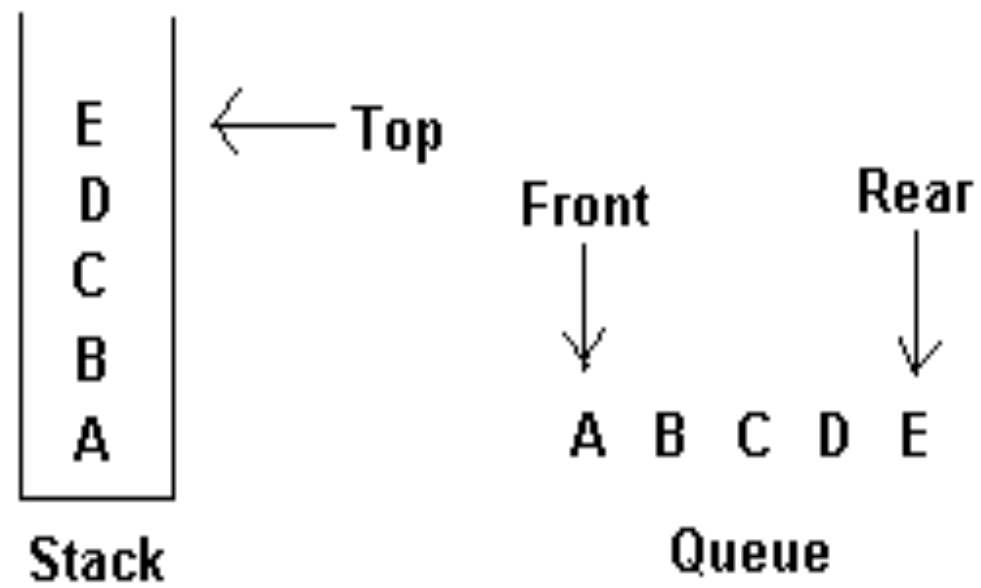
Filas usando arranjos

Java

Prof. Dr. José Olímpio Ferreira

Fila

- É uma lista linear em que todas as inserções são realizadas em um extremo da lista, e todas as retiradas são realizados no outro extremo da lista.
- O modelo intuitivo de uma fila é o de uma fila de espera em que as pessoas no início da fila são servidas primeiro e as pessoas que chegam entram no fim da fila.
- São chamadas listas fifo (“first-in”, “first-out”).



Fila

- Existe uma ordem linear para filas que é a “ordem de chegada”.
- São utilizadas quando desejamos processar itens de acordo com a ordem “primeiro-que-chega, primeiro-atendido”.
- Sistemas operacionais utilizam filas para regular a ordem na qual tarefas devem receber processamento e recursos devem ser alocados a processos.

TAD Filas

- Criar uma fila vazia.
- Enfileirar o item x no final da fila.
- Desenfileirar. Essa função retorna o item x no início da fila e o retira da fila.
- Verificar se a fila está vazia. Essa função retorna true se a fila está vazia; do contrário, retorna false.
- Verificar se a fila está cheia. Essa função retorna true se a fila está cheia; do contrário, retorna false.

Implementação de Filas por meio de Arranjos

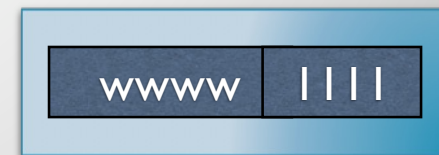
- Os itens são armazenados em posições contíguas de memória.
- A operação enfileira faz a parte de trás da fila expandir-se.
- A operação desenfileira faz a parte da frente da fila contrair-se.
- A fila tende a caminhar pela memória do computador, ocupando espaço na parte de trás e descartando espaço na parte da frente.
- Com poucas inserções e retiradas, a fila vai ao encontro do limite do espaço da memória alocado para ela.

Objetos do tipo Cliente

- **Atributos:**
 - string nome
 - int idade
- **Métodos:**
 - getNome, getIdade
 - setNome, setIdade

Classe Cliente

- nome: String
- idade: int (>0)
- + setNome(String): void
- + setIdade(int): boolean
- + getCliente(): String



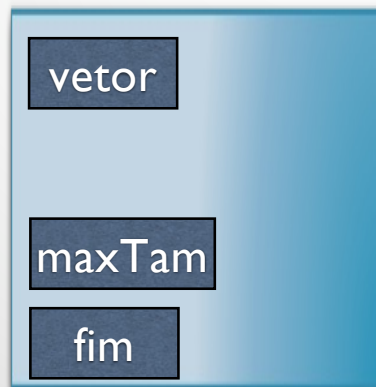
Objetos do tipo Fila

- Atributos:

- int fim, maxTam
- Cliente [] vetor

- Métodos:

- construtor
- enqueue
- dequeue
- filaVazia
- filaCheia
- getMaxTam

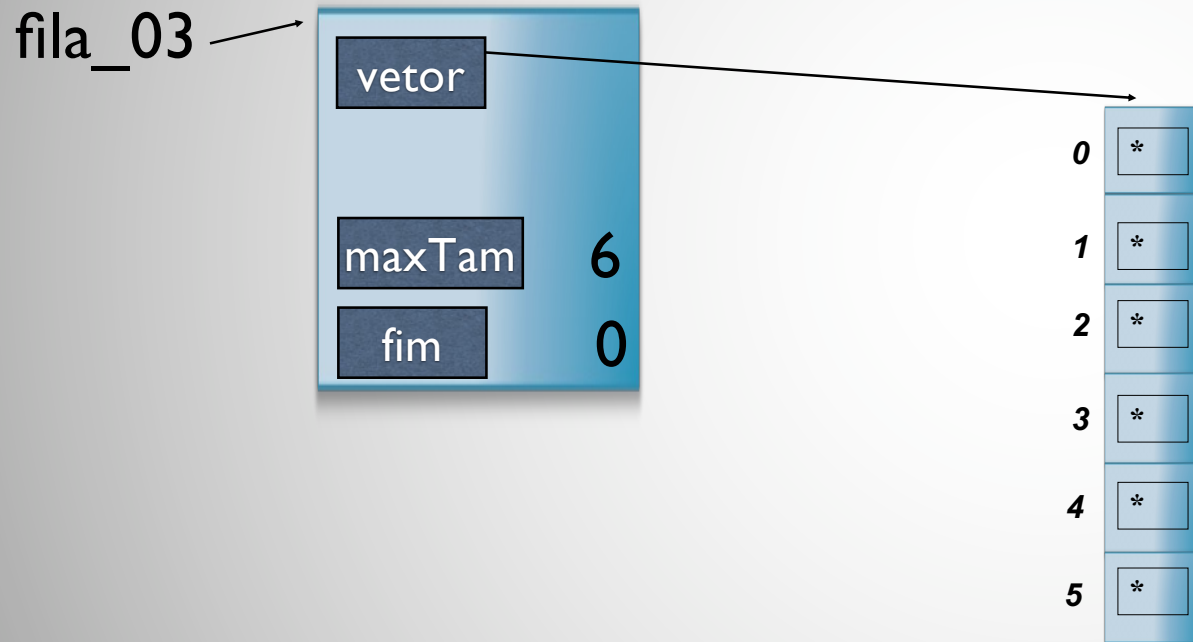


Classe Fila

- maxTam: int (≥ 2)
- fim: int
- vetor[]: Cliente
- + Fila()
- + enqueue(Cliente): boolean
- + dequeue(): Cliente
- + filaVazia(): boolean
- + filaCheia(): boolean
- + setMaxTam(int): boolean
- + getMaxTam(): int

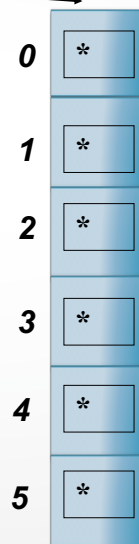
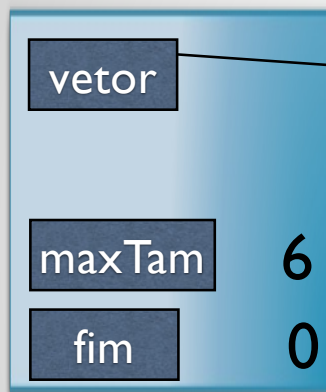
Fila(): Constrói fila vazia

boolean setMaxTam(int tam)

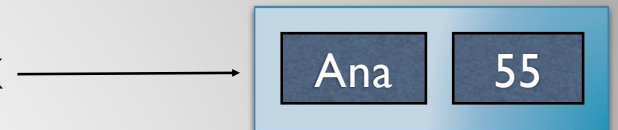


enqueue(x)

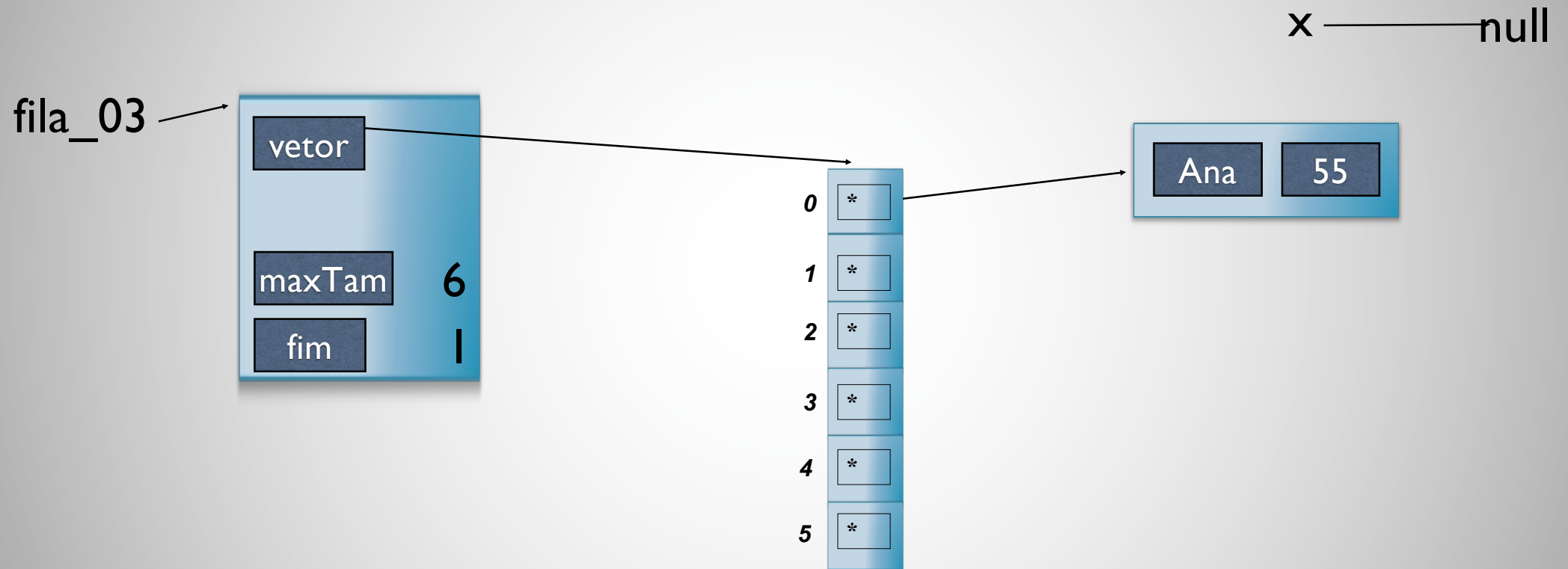
fila_03



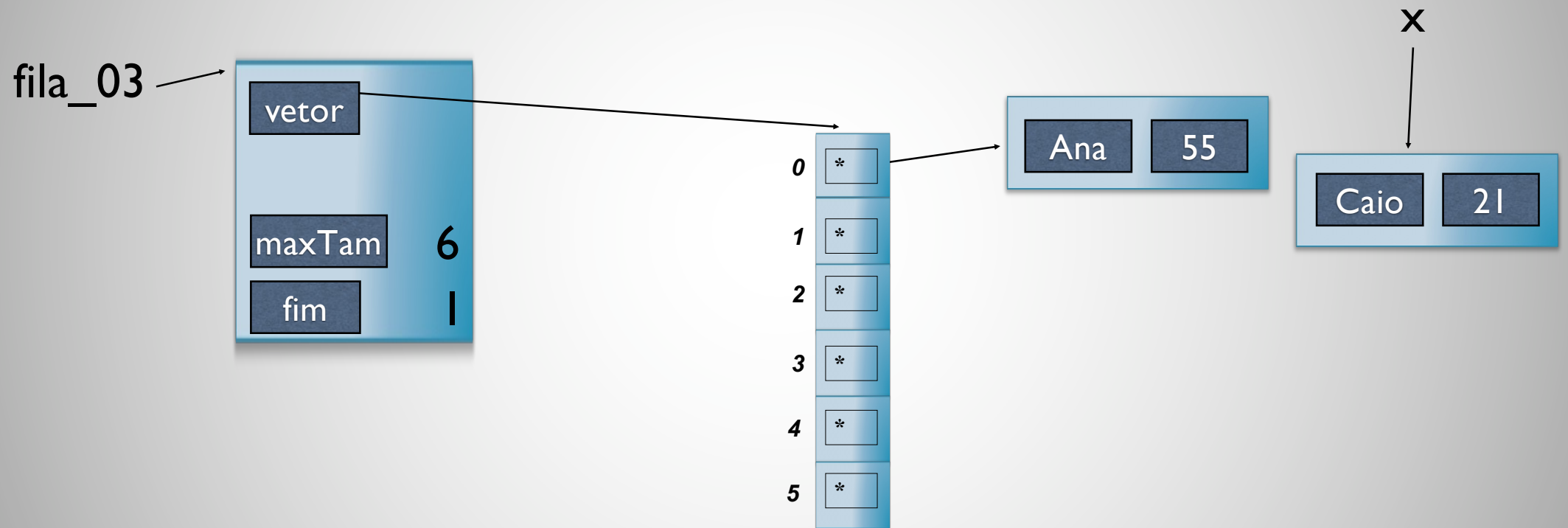
x



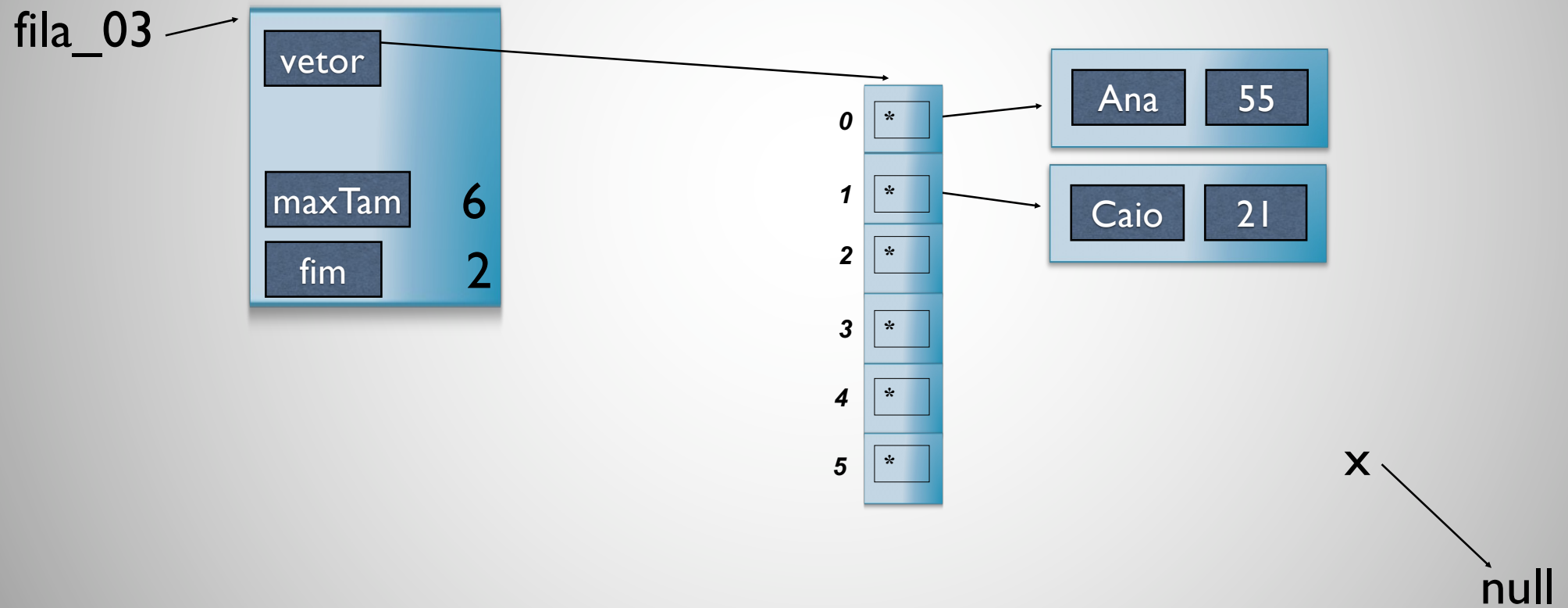
enqueue(x)



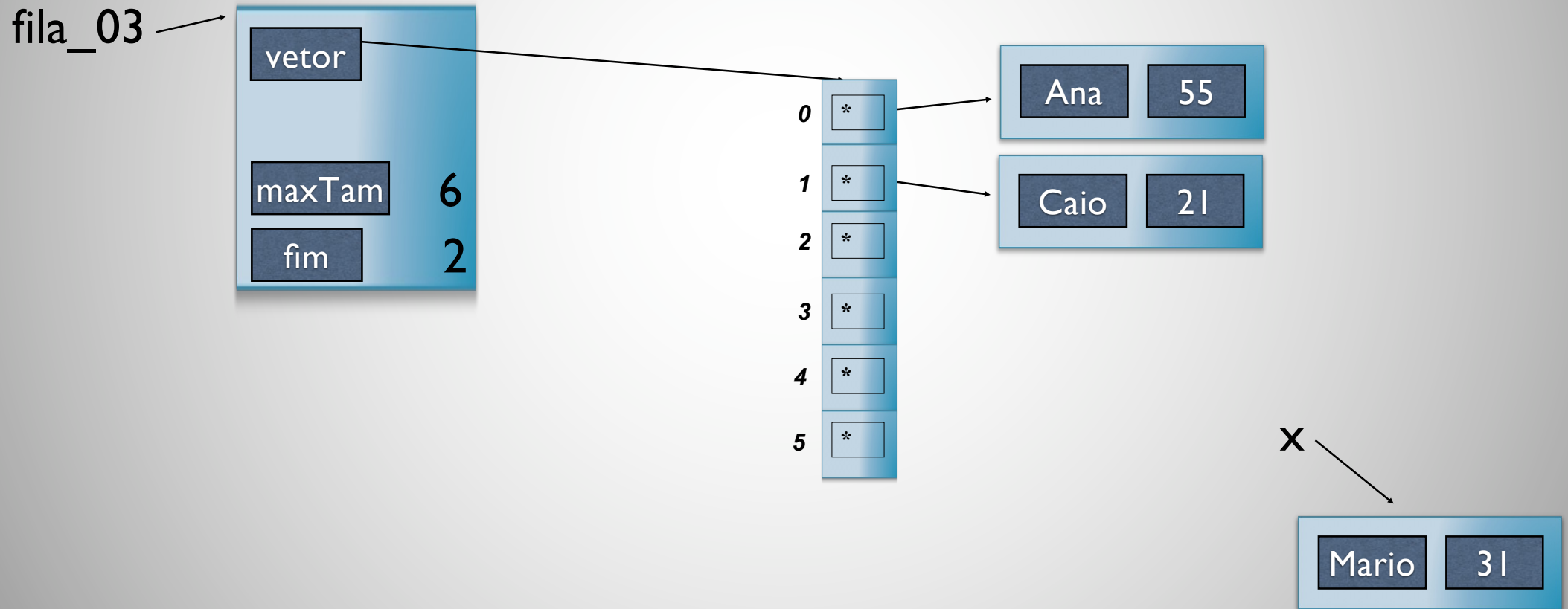
enfileirar(x)



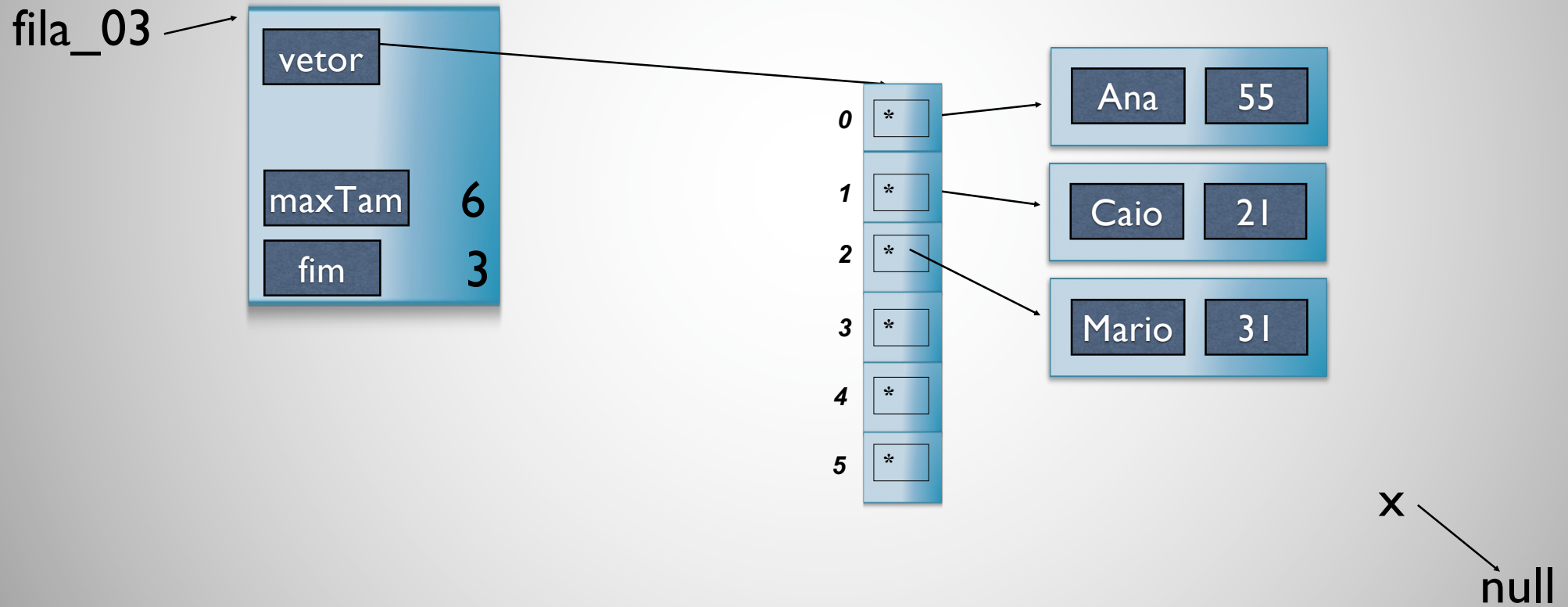
enqueue(x)



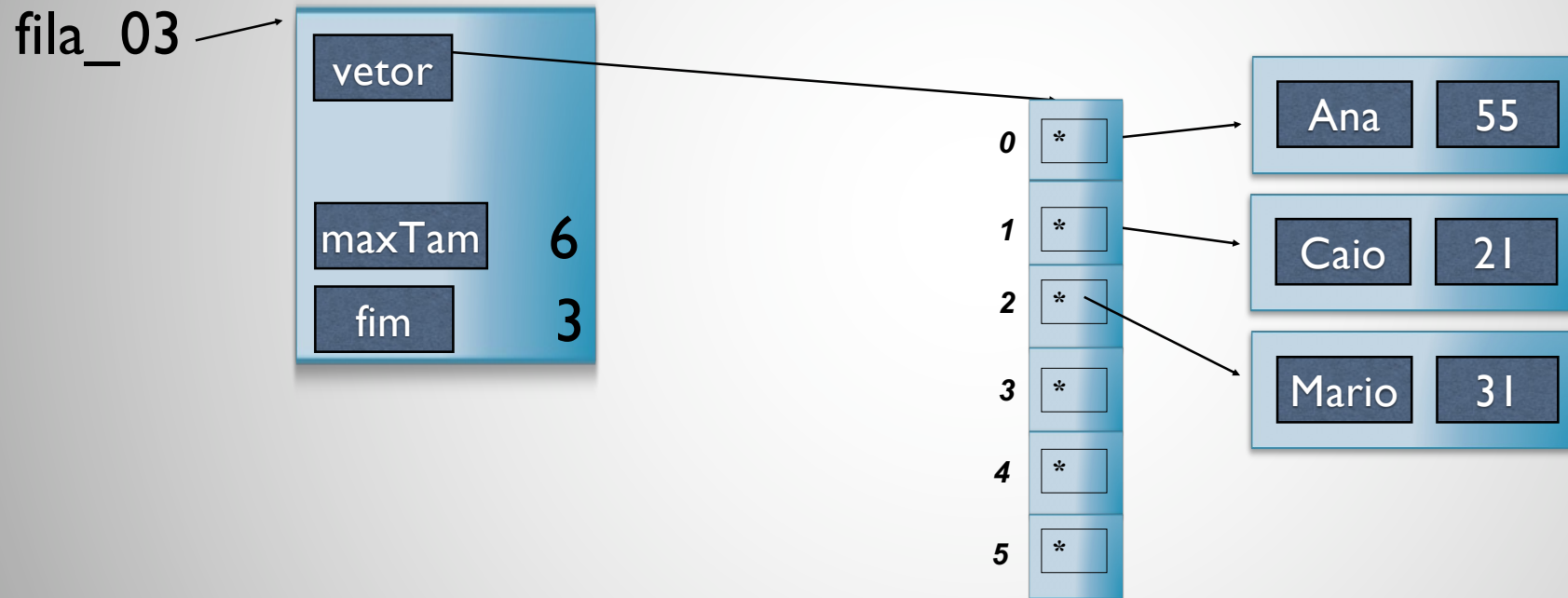
enqueue(x)



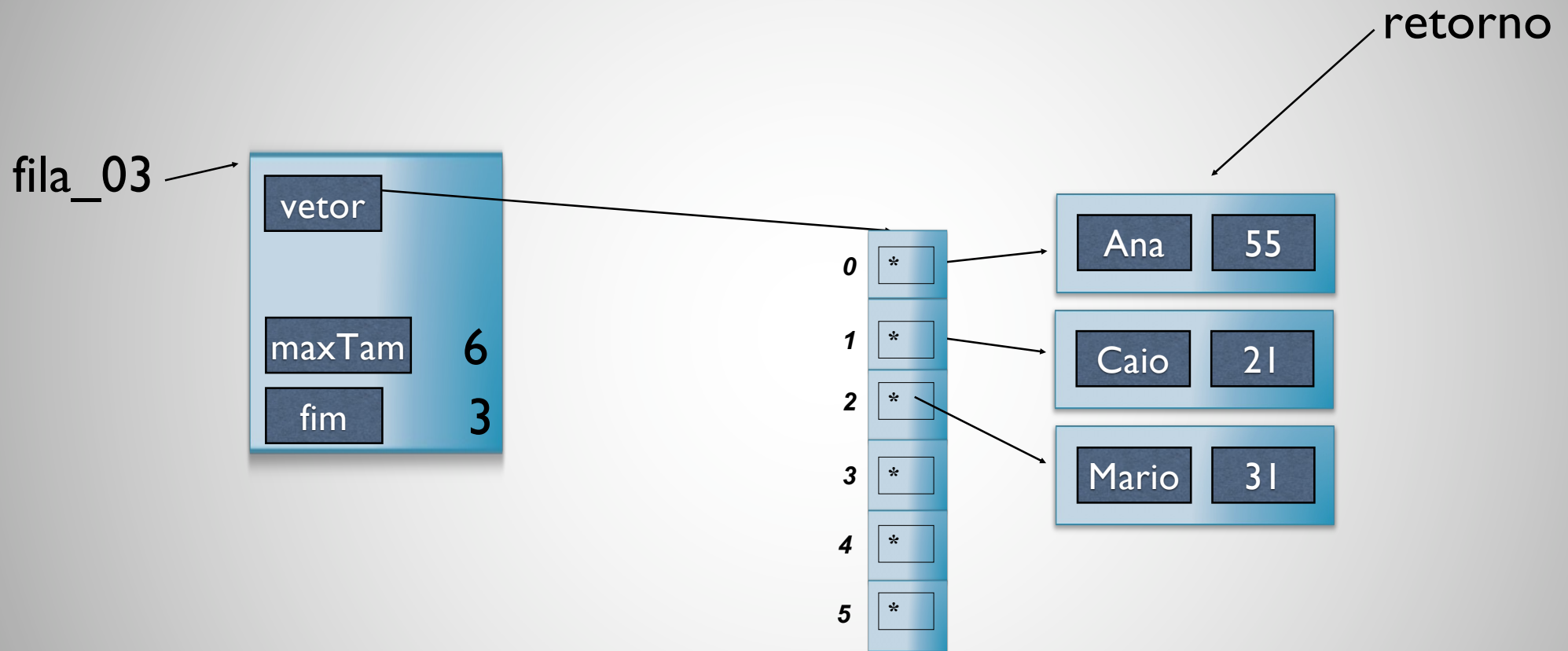
enqueue(x)



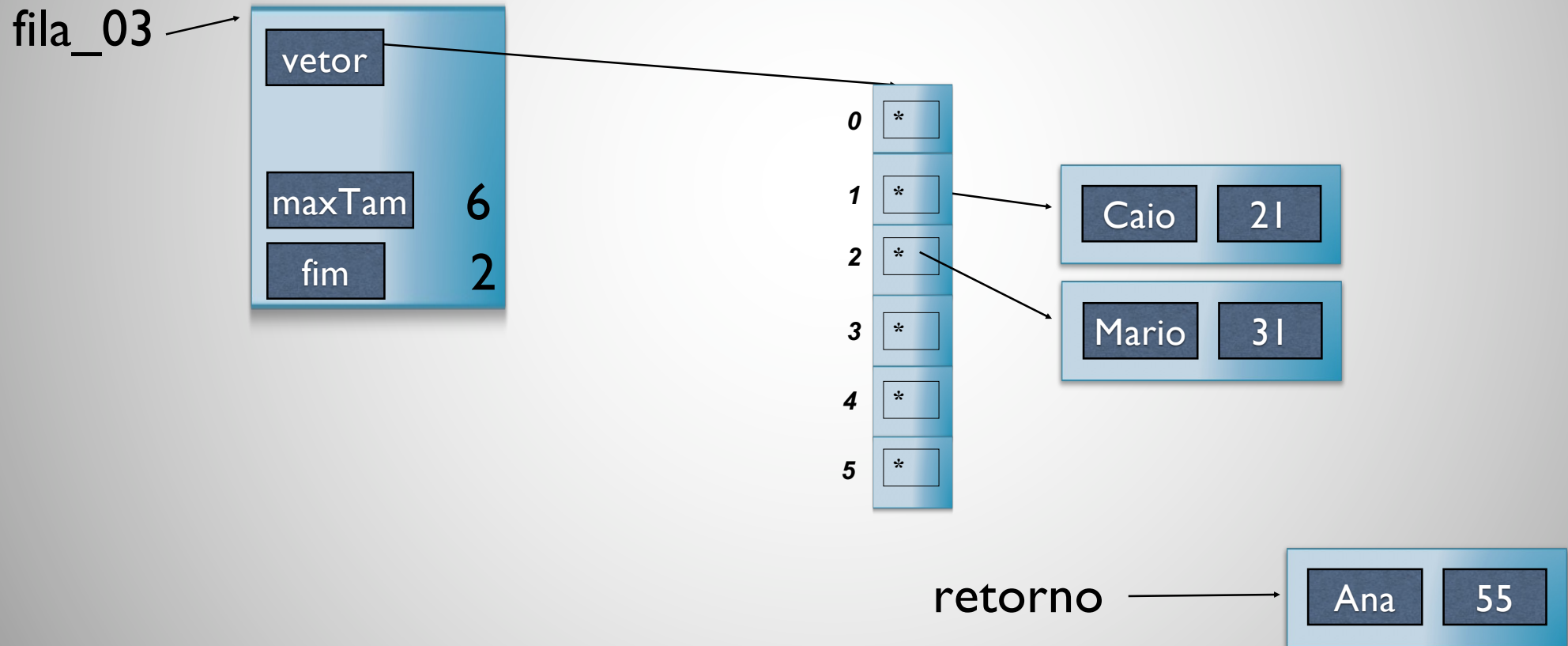
Estado da fila após 3 chegadas



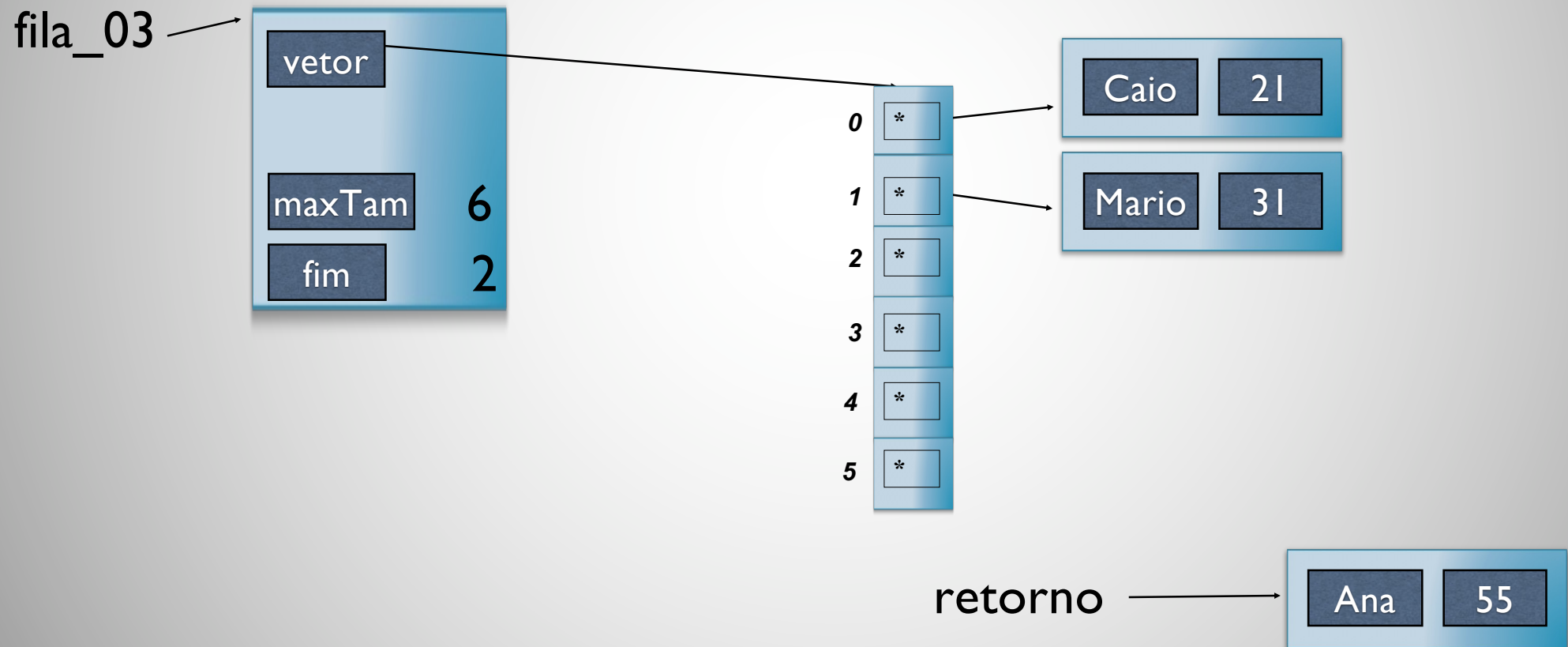
desenfileirar()



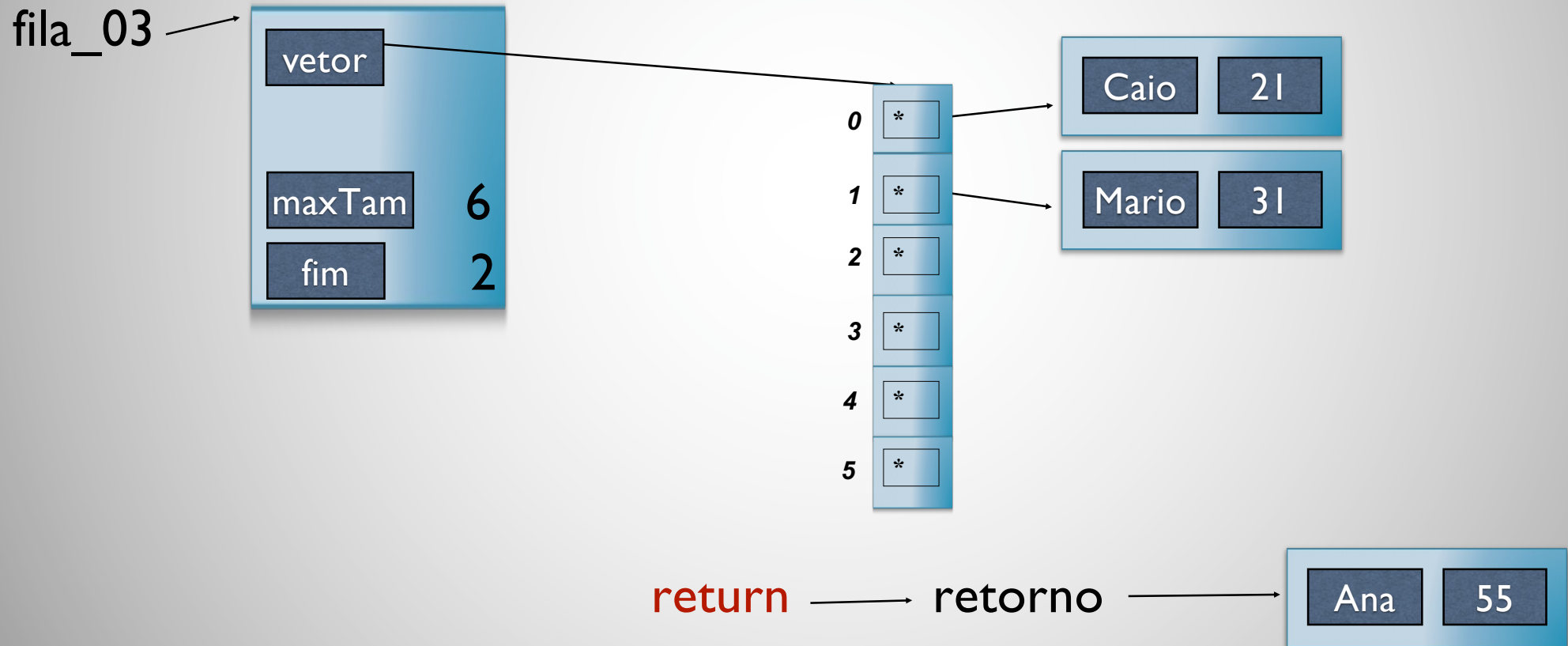
desenfileirar()



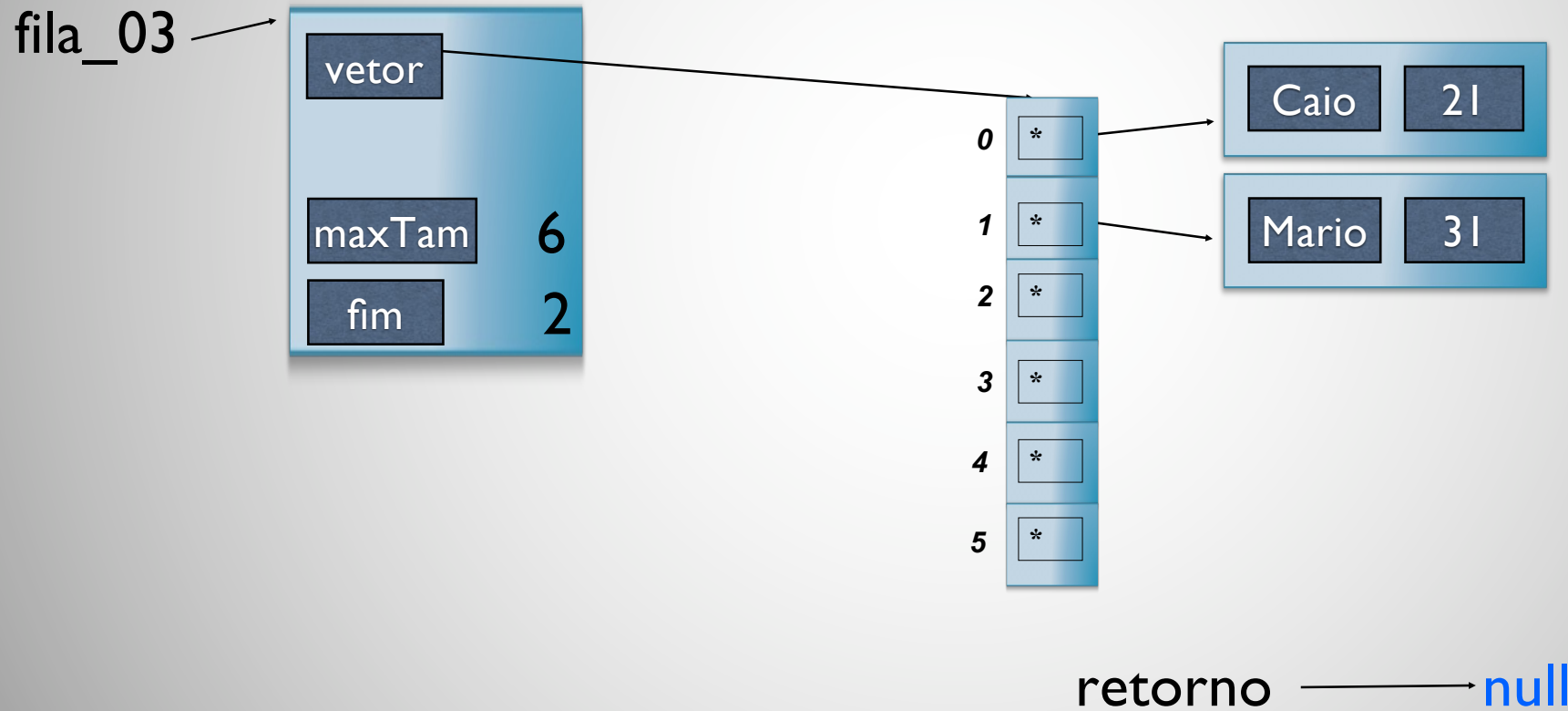
desenfileirar()

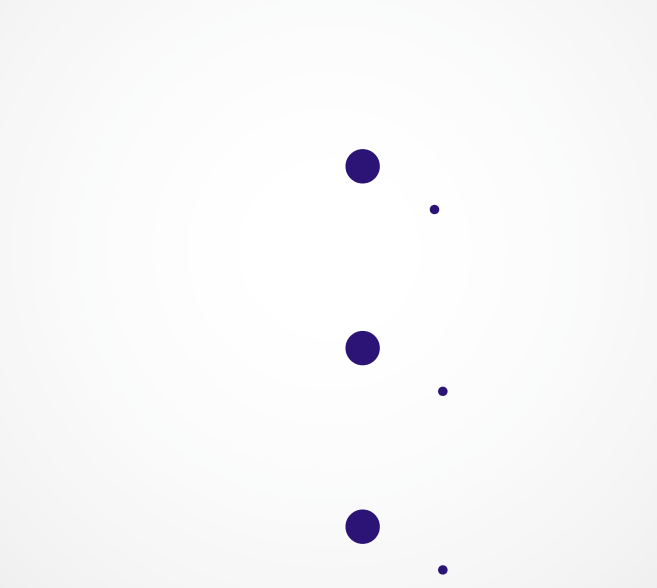


desenfileirar()

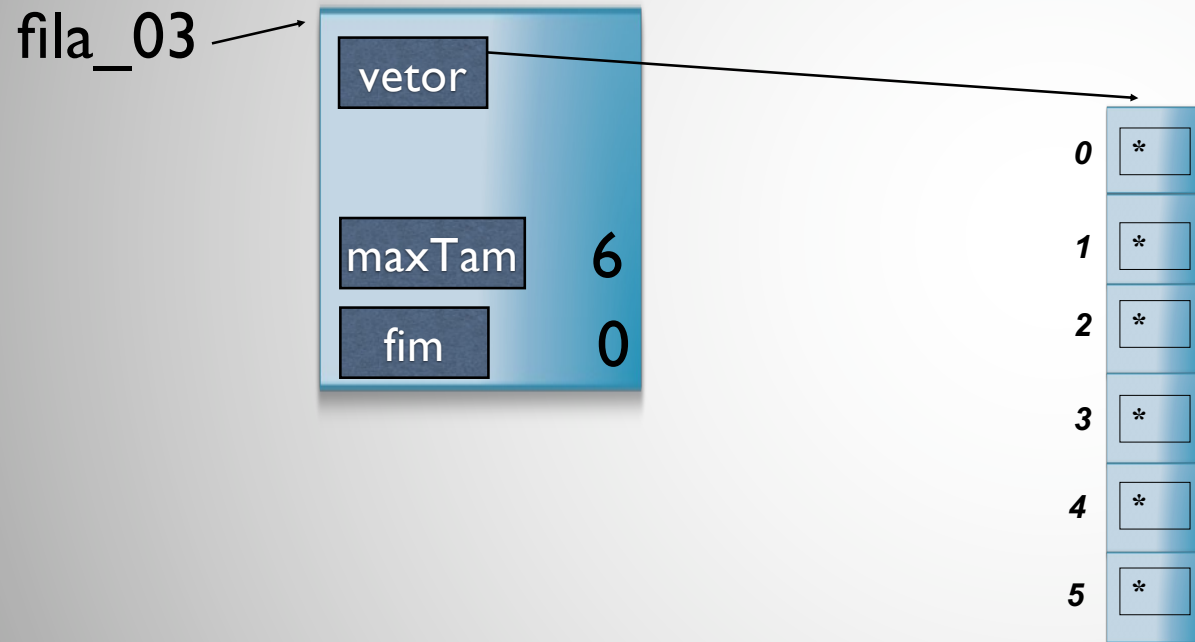


desenfileirar()





Estado da fila após 2 saídas



Objeto do tipo TestaFila (Base)

- Classe Main

- Atributos

- static Scanner leia

- Métodos

- static criarCliente()

- // cria e retorna um cliente

- static public main()

- // função principal

- // Fila fila // Objeto do tipo Fila

- // implementa um laço contendo um menu de escolhas (enfileirar , desenfileirar, encerrar)

Classe Main (Base)

- leia: static Scanner

+ criarCliente(): static Cliente

+ main(): static public

- obj: Fila

UML

Classe Cliente

- nome: String
- idade: int (>0)
- Cliente()
- + setNome(String): void
- + setIdade(int): boolean
- + getCliente(): String

Classe Main (Base)

- leia: static Scanner
- + criarCliente(): static Cliente
- + main(): static public
- obj: Fila

Classe Fila

- maxTam: int (≥ 2)
- fim: int
- vetor[]: Cliente
- + Fila()
- + enfileirar(Cliente): boolean
- + desenfileirar(): Cliente
- + filaVazia(): boolean
- + filaCheia(): boolean
- + setMaxTam(int): boolean
- + getMaxTam(): int