



Estrutura de dados

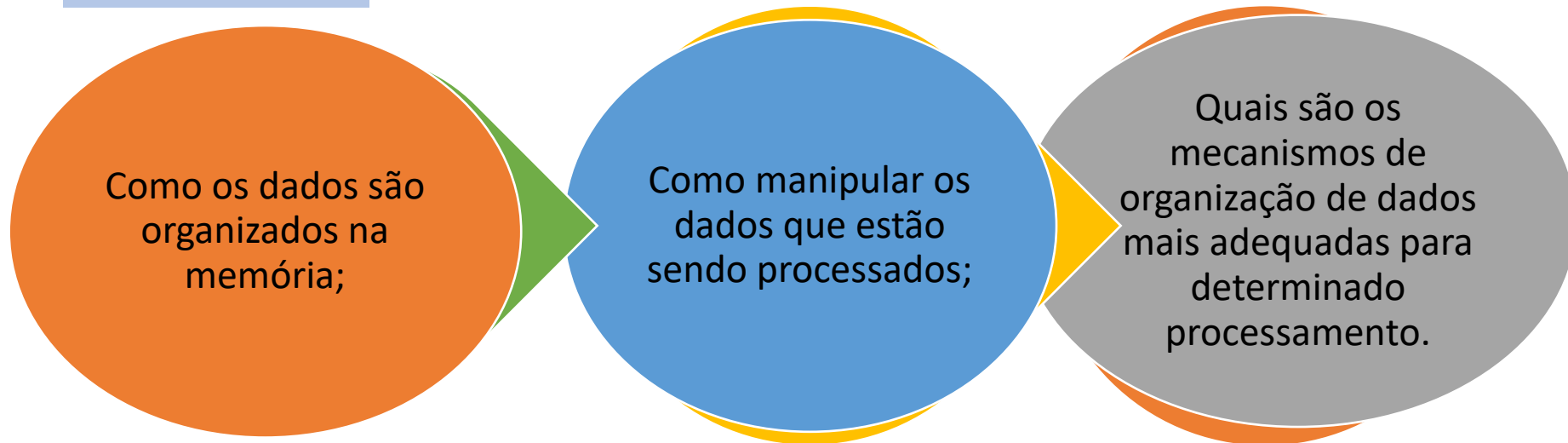
Prof. Ismar

Estrutura de dados é o ramo da computação que estuda os diversos mecanismos de organização de dados para atender aos diferentes requisitos de processamento.

Observação:

Não se trata de banco de dados!

Trata-se de :

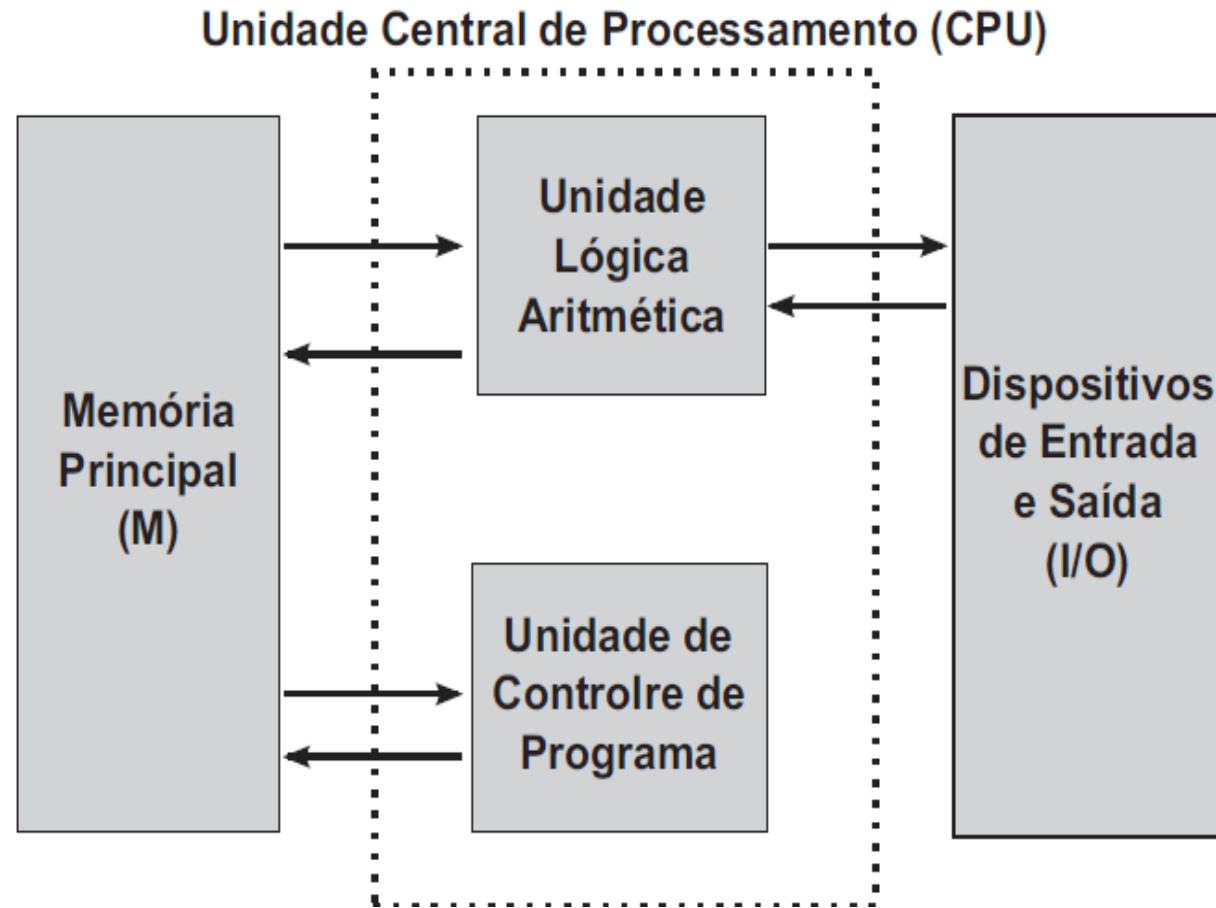




Estrutura de dados se refere a uma forma organizada de armazenamento e manipulação de informações em um computador. É uma área fundamental da ciência da computação e é amplamente utilizada em algoritmos, programação e engenharia de software.

Arquitetura de um Computador

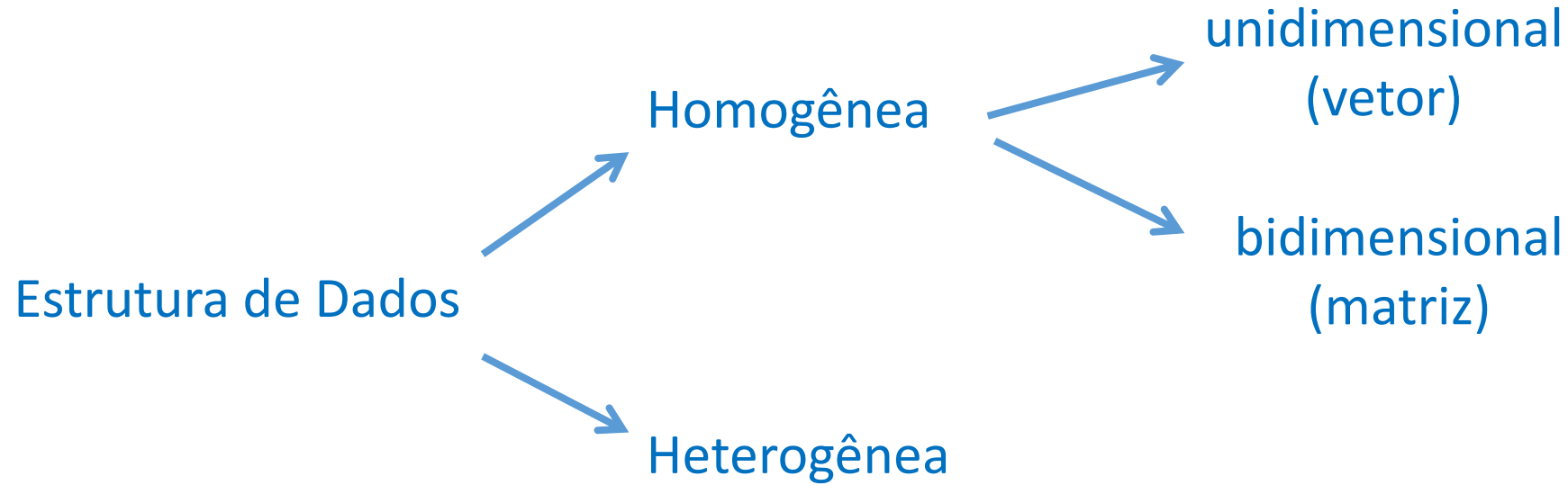
Máquina de Von Neumann



As estruturas de dados envolvem o armazenamento de dados organizados na memória de uma maneira mais sofisticada do que pela utilização de variáveis básicas usadas em algoritmos simples.

Quando se deseja representar dados de forma direta, a referência a eles é feita normalmente por meio de variáveis, mas quando se utiliza uma estrutura de dados, a referência aos dados é feita normalmente por índices.

Uma estrutura de dados permite executar operações e fazer a manipulação de dados. Ela indica como os dados são representados e como se torna possível a sua manipulação.



Cada uma dessas estruturas de dados é adequada para diferentes tipos de problemas e é importante escolher a estrutura certa para cada situação, a fim de otimizar a eficiência e a performance do programa.

Estrutura de Dados Heterogêneas Homogêneas

Quando uma estrutura de dados possui elementos do mesmo tipo é chamada de **homogênea**.

Uma estrutura de dados homogênea é representada por um vetor (quando se trata de uma estrutura unidimensional) e por uma matriz (quando se trata de uma estrutura bidimensional).

Quando uma estrutura de dados possui elementos de tipos diferentes, é chamada de **heterogênea**. Esse tipo de estrutura é também chamado de registro.

Então...

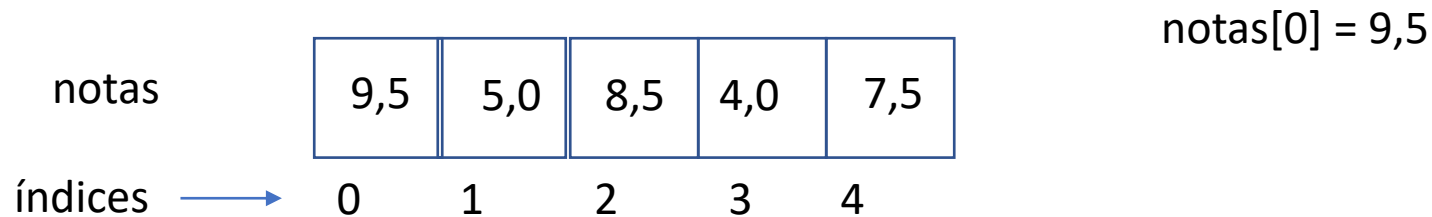
Registro é uma estrutura de dados que faz agrupamento de dados com tipos diferentes entre si. Em outras palavras, registro é algo que agrupa elementos que não são do mesmo tipo, mas que têm vínculo lógico.

Vetores

Um vetor é uma estrutura de dados utilizada para armazenar uma lista de valores do mesmo tipo

Uma estrutura do tipo vetor tem as seguintes características:

- É alocado de forma estática, ou seja, no momento da sua declaração, deve-se conhecer o tamanho (quantidade de posições que ele terá).
- É uma estrutura de dados homogênea.
- Cada posição do vetor contém somente um valor.
- Os tamanhos dos valores de cada posição, por serem do mesmo tipo, são iguais em número de bytes.
- Faz a alocação dos dados de forma sequencial.



Matrizes

Um vetor é uma estrutura de dados bidimensional (é necessário dois índices para definir uma posição)

Uma estrutura do tipo matriz tem as seguintes características:

- É alocada de forma estática, no momento da sua declaração, deve-se conhecer o tamanho que ela terá.
- É uma estrutura de dados homogênea.
- Cada posição do matriz contém somente um valor.
- Os tamanhos dos valores de cada posição, por serem do mesmo tipo, são iguais.
- Faz a alocação dos dados de forma sequencial.

		colunas				
		0	1	2	3	4
linhas	0	10	20	30	40	50
	1	60	70	80	90	100
	2	110	120	130	140	50

`mat[1][3] = 90`



Noções de Python

A linguagem de programação Python é uma linguagem de alto nível e de uso geral, conhecida por sua sintaxe simples e legibilidade. Foi criada por Guido van Rossum e lançada pela primeira vez em 1991. Python tem sido amplamente adotada tanto por desenvolvedores iniciantes como por profissionais, devido à sua facilidade de aprendizado e sua vasta gama de aplicações.

Python é uma linguagem interpretada, o que significa que o código fonte é executado linha por linha por um interpretador. Isso torna o desenvolvimento rápido e fácil, além de ser multiplataforma, funcionando em sistemas operacionais como Windows, macOS e Linux.

principais características da sintaxe do Python

A sintaxe de uma linguagem são as regras que nos dizem como escrever um código para que ele seja compilado sem erros.

variável

Nas linguagens de programação, a variável é um local na memória que reservamos para armazenar dados. Para **criar uma variável no Python** basta informar nome e valor

```
1 | total_alunos = 10  
2 | print(total_alunos)
```

variável

Python é uma linguagem case-sensitive, que faz diferenciação entre letras maiúsculas e minúsculas. Sendo assim, `total_alunos` e `Total_alunos` seriam nomes de variáveis diferentes.

```
1 total_aulas = 5
2 TOTAL_AULAS = 10
3
4 print(total_aulas)
5 print(TOTAL_AULAS)
```

Palavras reservadas ou Palavras-chave

Python, como toda linguagem, possui um conjunto de palavras reservadas, também chamadas de palavras-chave, que são termos que possuem significado especial para o compilador e que não podem utilizados como nomes. Abaixo temos uma lista destes termos:

```
1 | and, as, assert, break, class, continue, def, del, elif, else, except, False, finally,  
2 | for, from, global, if, import, in, is, lambda, None, nonlocal, not, or, pass, raise,  
3 | return, True, try, while, with, yield
```

Indentação

A indentação é uma parte fundamental na linguagem Python. Diferente de outras linguagens de programação, onde a indentação é apenas uma questão de legibilidade do código, em Python ela tem um significado sintático.

Em Python, a indentação é usada para indicar blocos de código, como loops, condicionais e funções. É necessário ser consistente com a quantidade de espaços ou tabulações em todo o código.

É importante notar que a escolha entre espaços ou tabulações é uma questão de preferência pessoal. No entanto, é recomendado utilizar 4 espaços como padrão de indentação em Python, conforme definido pela PEP 8 (guia de estilo da linguagem). Certifique-se de manter uma consistência na indentação do seu código para evitar erros de sintaxe. A falta de indentação ou uma indentação inconsistente pode causar erros como "IndentationError" em Python

```
if condicao:
    # Início do bloco if, indentado com 4 espaços
    codigo1
    codigo2
    ...
    # Fim do bloco if
else:
    # Início do bloco else, indentado com 4 espaços
    codigo3
    codigo4
    ...
    # Fim do bloco else
```

Nomes de variáveis em Python

Em Python, os nomes de variáveis também devem obedecer a seguintes regras:

- sequência de letras ($a \rightarrow z$, $A \rightarrow Z$) e números ($0 \rightarrow 9$),
- sempre começar com uma letra.
- Apenas letras comuns são permitidas. Letras acentuadas, cedilhas, espaços, caracteres especiais como \$, #, @, etc. são proibidos, exceto para o caractere _ (underline).

Convenção: escrever a maioria dos nomes de variáveis em caracteres minúsculos (incluindo a primeira letra). Isso é amplamente respeitado.

Use letras maiúsculas dentro nome da variável, para possivelmente aumentar a legibilidade.

Ex: `tabelaDeMateriasPrimas`.

Tipos de variáveis disponíveis no Python

Python é uma linguagem dinamicamente tipada:

Significa que não é necessário declarar o tipo de variável ou fazer *casting* (mudar o tipo de variável), pois o Interpretador se encarrega disso.

Isso significa também que o tipo da variável poder variar durante a execução do programa.

- Inteiro (int)
- Ponto Flutuante ou Decimal (float)
- String (str)
- Boolean (bool)
- List (list)
- Tuple
- Dictionary (dic)

Exemplo de uma estrutura em C:

Esta é a forma de definir uma estrutura de dados na linguagem de programação C:

```
struct nome_da_estrutura {  
    tipo_do_campo1 nome_do_campo1;  
    tipo_do_campo2 nome_do_campo2;  
    tipo_do_campo3 nome_do_campo3;  
    ...  
};
```

```
struct boletim {  
    char nome_aluno (50);  
    char disciplina (50);  
    int qtd_faltas;  
    float nota1;  
    float nota2;  
};
```

Ponteiros

Um ponteiro é um tipo de dado que faz uma referência indireta a um valor, ou seja, ele permite referenciar a posição de um objeto na memória (endereço em hexadecimal), ou seja, Pontoeiro é uma variável especial que armazena o endereço de memória de uma outra variável.

Ex:

```
Int var1 = 10;  (Suponha que esta variável esteja alocada na posição de memória 1000)
```

```
Int *p;
```

```
p = &var1;
```

```
printf("Conteúdo da variável var1: %d", var1);  (a resposta é 10)
```

```
printf("Endereço da variável var1: %x", &var1); (a resposta é 1000)
```

```
printf("Conteúdo da variável p: %x", p);        (a resposta é 1000)
```

Prioridade das operações

Quando há mais de um operador em uma expressão, a ordem em que as operações devem ser executadas depende das regras de precedência. Em Python, as regras de prioridade são as mesmas usuais da matemática:

- parênteses têm maior prioridade. Eles permitem que você “force” a avaliação de uma expressão na ordem desejada.

$$2 * (3-1) = 4$$

$$(1 + 1) ** (5-2) = 8'.$$

- exponenciação é avaliado antes de outras operações.

$$2 ** 1 + 1 = 3$$

$$3 * 1 ** 10 = 3$$

Prioridade das operações

- Multiplicação e divisão têm a mesma prioridade. Eles são avaliados antes da adição e da subtração.

$$2 * 3 - 1 = 5$$

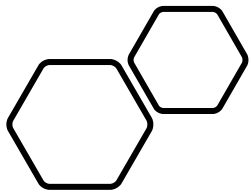
$$2 / 3 - 1 = -0,3333$$

Se dois operadores tiverem a mesma prioridade, o cálculo é feito da esquerda para a direita.

Ex:

$$2 * 3 // 2 = 3 \text{ e não } 2$$

Obs: (lembre-se que o operador // executa uma divisão inteira)



Estrutura de dados

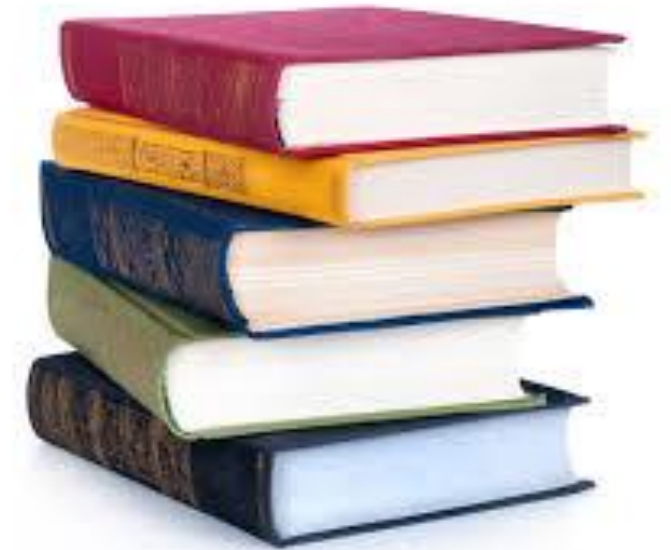
Episódio: Pilhas

Pilhas

- Uma pilha é uma estrutura de dados que pode ser acessada somente por uma de suas extremidades para armazenar e recuperar dados.
- Por essa razão, uma pilha é chamada de estrutura *LIFO* (*last in / first out*).

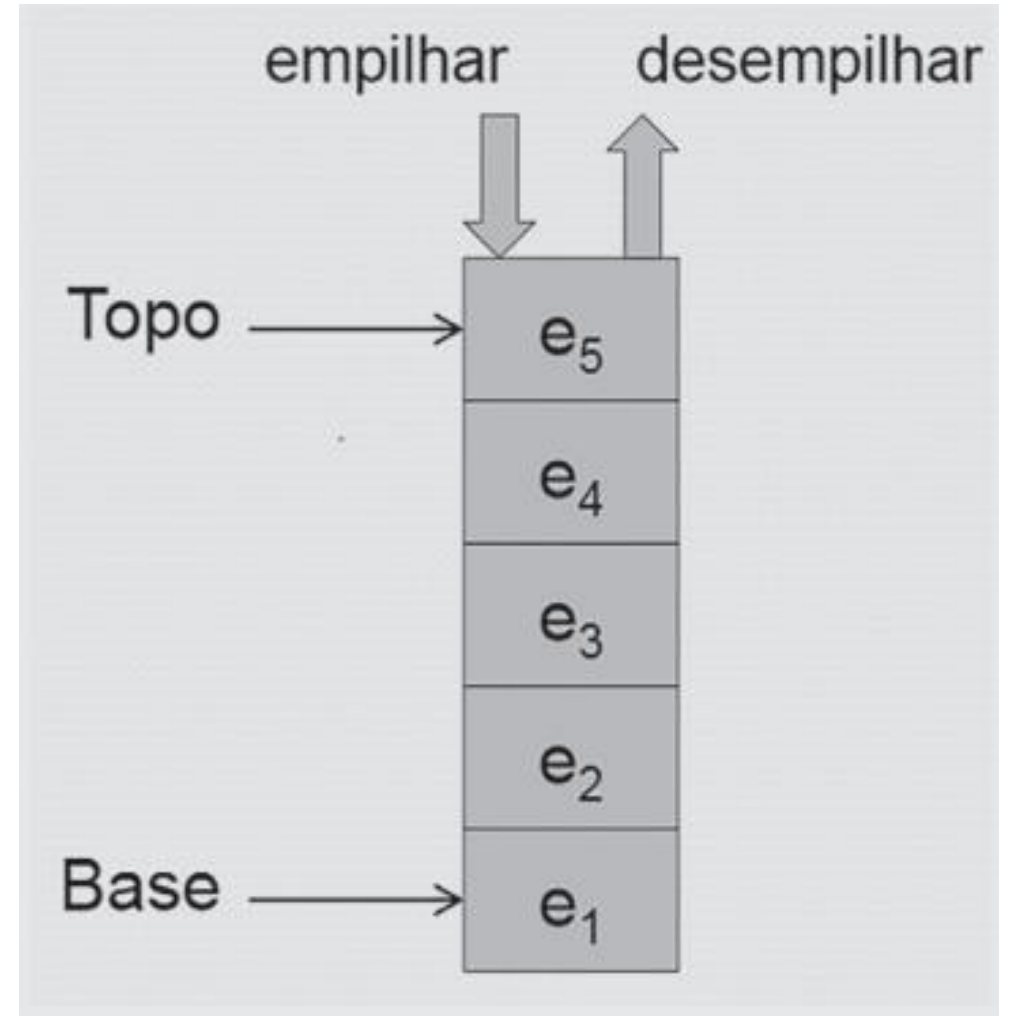
Operações sobre pilhas:

- pull – cria uma pilha;
- push – adiciona novo item na pilha (inserção);
- pop – Remove o último item adicionado e o retorna.
- top – acessa um elemento;
- isEmpty – verifica se a pilha está vazia;



Extremidades da Pilha

Apenas na extremidade chamada topo, é possível realizar a manipulação dos dados. A outra extremidade da estrutura chamamos de base.



Operações sobre pilhas:

- push – adiciona novo item na pilha (inserção);
- pop – Remove o último item adicionado e o retorna.
- top – acessa um elemento;
- isEmpty – verifica se a pilha está vazia;
- size – exibe o quantidade de elementos da pilha.



Exemplos:

Estado inicial	Operação	Estado final						
<table><tr><td></td></tr><tr><td></td></tr><tr><td>10</td></tr></table>			10	push(6)	<table><tr><td></td></tr><tr><td>6</td></tr><tr><td>10</td></tr></table>		6	10
10								
6								
10								
<table><tr><td></td></tr><tr><td>6</td></tr><tr><td>10</td></tr></table>		6	10	push(15)	<table><tr><td>15</td></tr><tr><td>6</td></tr><tr><td>10</td></tr></table>	15	6	10
6								
10								
15								
6								
10								



Exemplos:

Estado inicial	Operação	Estado final						
<table><tr><td>15</td></tr><tr><td>6</td></tr><tr><td>10</td></tr></table>	15	6	10	pop()	<table><tr><td></td></tr><tr><td>6</td></tr><tr><td>10</td></tr></table>		6	10
15								
6								
10								
6								
10								
<table><tr><td></td></tr><tr><td>6</td></tr><tr><td>10</td></tr></table>		6	10	pop()	<table><tr><td></td></tr><tr><td></td></tr><tr><td>10</td></tr></table>			10
6								
10								
10								



Exemplos:

Estado inicial	Operação	Estado final						
<table><tr><td>30</td></tr><tr><td>20</td></tr><tr><td>10</td></tr></table>	30	20	10	pop(), pop(), push(15)	<table><tr><td></td></tr><tr><td>15</td></tr><tr><td>10</td></tr></table>		15	10
30								
20								
10								
15								
10								



FILAS



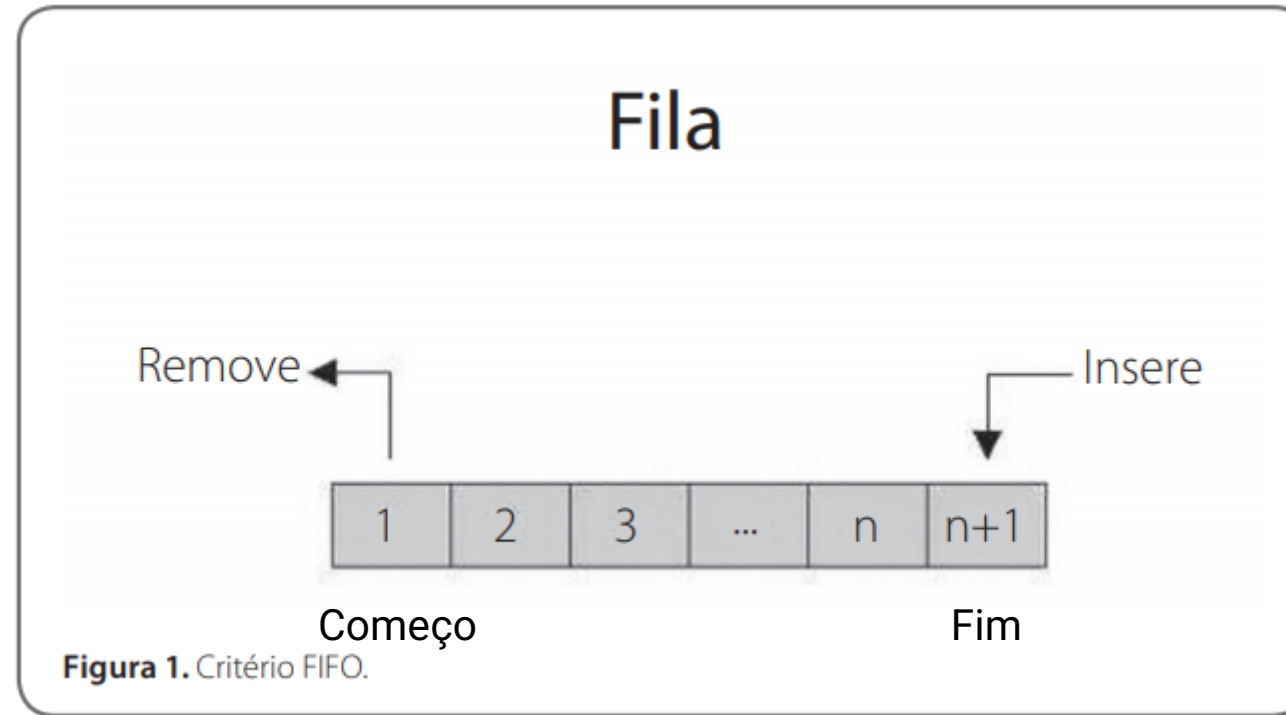
- As filas existem para estruturar, organizar, facilitar o acesso aos elementos e armazenar elementos.

Filas



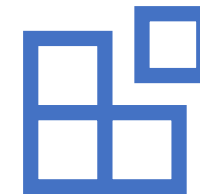
- Uma fila é uma estrutura de dados em que as inserções e retiradas dos elementos são feitas em extremidades diferentes.
- O primeiro elemento a entrar será o primeiro a sair, por essa razão, uma fila é chamada de estrutura *FIFO* (*first in / first out*).

Estrutura *FIFO* (*first in / first out*).



Operações sobre filas:

- insert – adiciona novo item na fila (enqueue);
- remove – Remove o primeiro item adicionado e o retorna (dequeue);
- isEmpty – verifica se a fila está vazia;
- size - retorna o número de elementos que existem na Fila



A palavra queue, da língua inglesa, significa fila.

As duas operações básicas que uma fila suporta são:

- enqueue: - insere um elemento no final da fila.
- dequeue: - remove um elemento do começo da fila.

Uso das filas

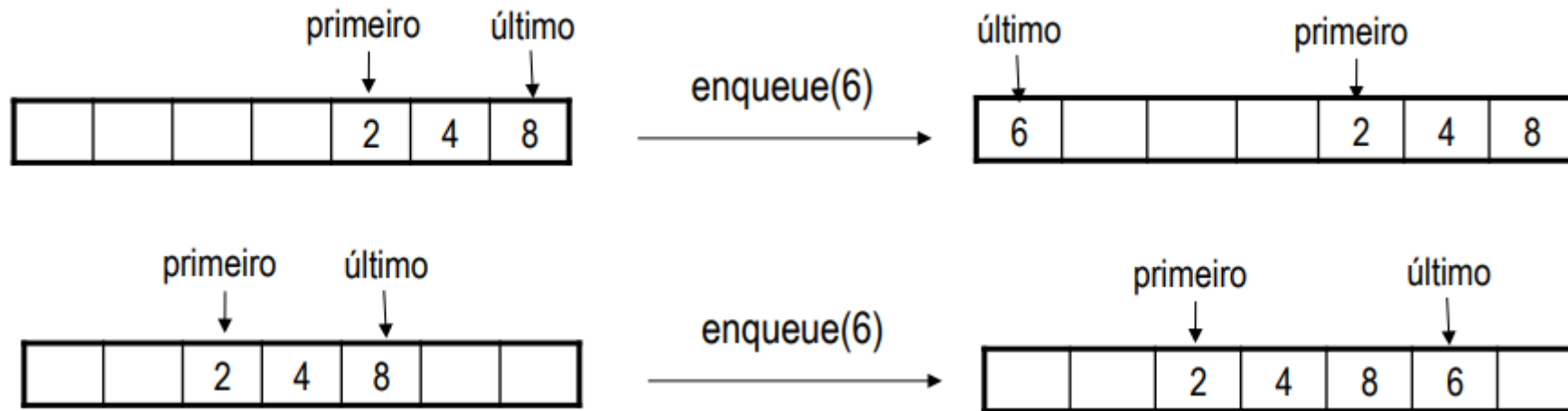
As filas são frequentemente usadas em simulações.

Ex:

- Peças em linhas de montagem;
- Caminhões esperando para descarregar;
- Atendimento de pessoas;
- Controle de documentos para impressão;
- Troca de mensagem entre computadores numa rede;
- Gerenciamento de processos em sistemas operacionais em que os processos aguardam na fila de pronto para execução;
- Processos de comunicação em redes de computadores.

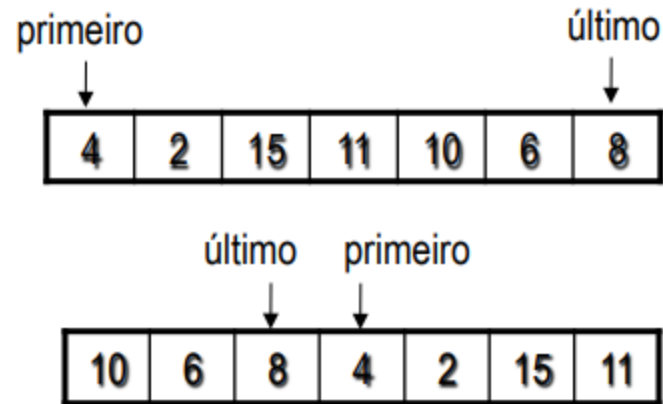
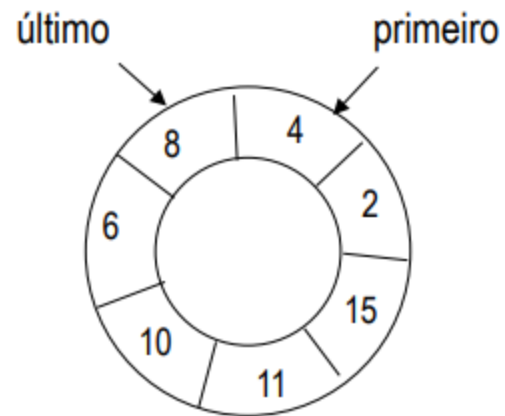
Funcionamento das filas (implementação estática)

- A implementação de filas pode ser realizada por meio de vetor.
- Se o último elemento estiver na última célula do vetor, mas houver células disponíveis antes do primeiro elemento, um novo elemento pode ser inserido.
- Se o último elemento estiver em qualquer outra posição, o novo elemento é colocado depois do último, se houver espaço.



Funcionamento das filas

Essa situação pode ser visualizada como uma matriz circular.



A fila está cheia se o primeiro elemento imediatamente precede o último na direção anti-horária.

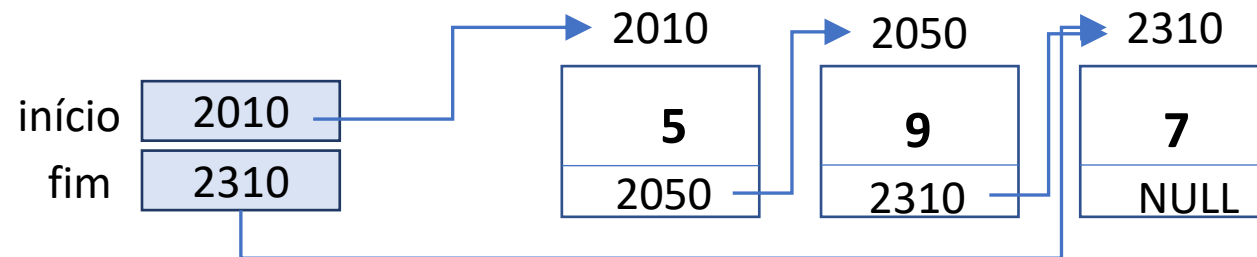
Sendo F uma Fila e x um elemento, a operação F:enqueue(x) aumenta o tamanho da Fila F, acrescentando o elemento x no seu final.

A operação F:dequeue(x), faz o inverso.

Estado inicial	Operação	Estado final
F:[]	F:enqueue(a)	F:[a]
F:[a]	F:enqueue(b)	F:[a,b]
F:[a,b]	F:enqueue(c)	F:[a,b,c]
F:[a,b,c]	F:enqueue(d)	F:[a,b,c,d]
F:[a,b,c,d]	F:dequeue()	F:[b,c,d]
F:[b,c,d]	F:enqueue(F:dequeue())	F:[c,d,b]
F:[c,d,b]	F:enqueue(e)	F:[c,d,b,e]
F:[c,d,b,e]	F:enqueue(F:dequeue())	F:[d,b,e,c]

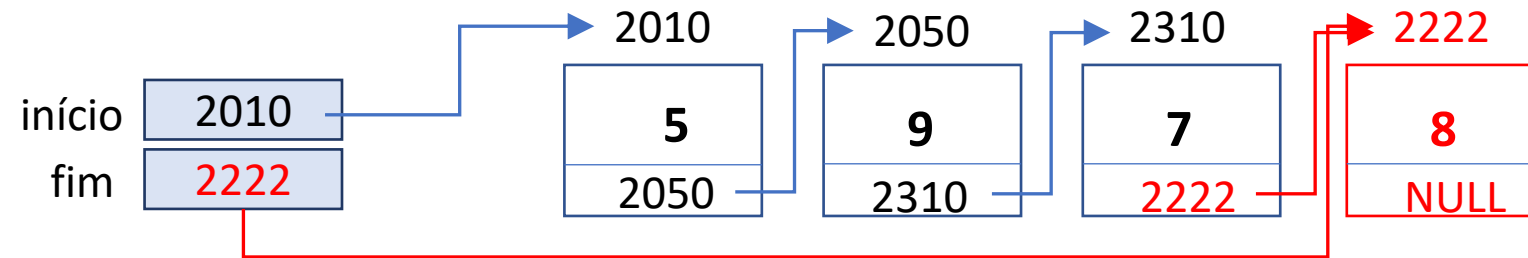
Funcionamento das filas (implementação dinâmica)

- A memória é alocada sob **demanda**.
- Cada elemento indica quem é o seu **sucessor** (quem é o próximo na fila).
- Os endereços de **início** e **fim** da fila são controlados.



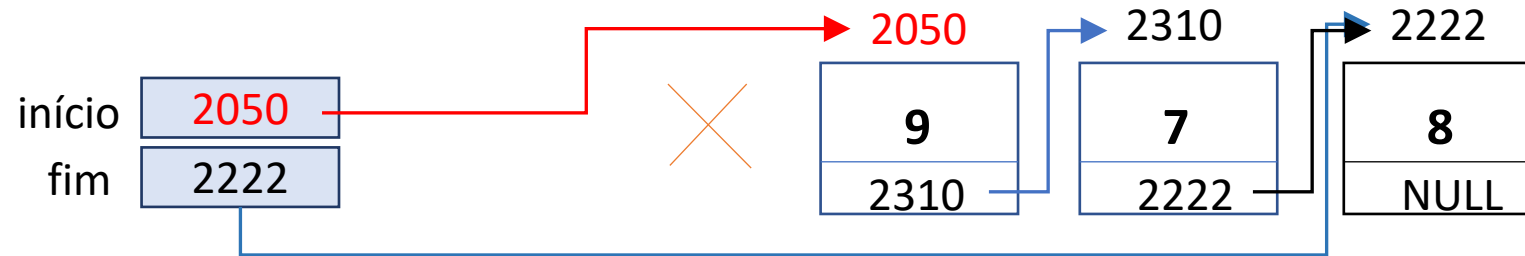
Funcionamento das filas (implementação dinâmica)

- Inserção de um novo elemento:



Funcionamento das filas (implementação dinâmica)

- Exclusão de um elemento:
(a memória ocupada pelo primeiro elemento é liberada)



Na tabela abaixo, em ordem crescente, estão as unidades de medida em tecnologia.

Unidade	Símbolo	Valor
Bit	b	dígito binário
Byte	B	8 bits
Kilobyte	kByte	1.024 bytes
Megabyte	MB	1.024 kilobytes
Gigabyte	GB	1.024 megabytes
Terabyte	TB	1.024 gigabyte
Petabyte	PB	1.024 terabytes
Exabyte	EB	1.024 petabytes
Zettabyte	ZB	1.024 exabytes
Yottabyte	YB	1.024 zettabytes

Todas as unidades

- 1 Bit = Binary Digit
- 8 Bits = 1 Byte
- 1024 Bytes = 1 Kilobyte
- 1024 Kilobytes = 1 Megabyte
- 1024 Megabytes = 1 Gigabyte
- 1024 Gigabytes = 1 Terabyte
- 1024 Terabytes = 1 Petabyte
- 1024 Petabytes = 1 Exabyte
- 1024 Exabytes = 1 Zettabyte
- 1024 Zettabytes = 1 Yottabyte
- 1024 Yottabytes = 1 Brontobyte
- 1024 Brontobytes = 1 Geopbyte
- 1024 Geopbytes = 1 Saganbyte
- 1024 Saganbytes = 1 Pijabyte
- 1024 Pijabytes = 1 Alphabyte
- 1024 Alphabytes = 1 Kryatbyte
- 1024 Kryatbytes = 1 Amosbyte
- 1024 Amosbytes = 1 Pectrolbyte
- 1024 Pectrolbytes = 1 Bolgerbyte
- 1024 Bolgerbytes = 1 Sambobyte
- 1024 Sambobytes = 1 Quesabyte
- 1024 Quesabytes = 1 Kinsabyte
- 1024 Kinsabytes = 1 Rutherbyte
- 1024 Rutherbytes = 1 Dumbnibyte
- 1024 Dumbnibytes = 1 Seaborgbytte
- 1024 Seaborgbyttes = 1 Bohrbyte
- 1024 Bohrbytes = 1 Hassiubyte
- 1024 Hassiubytes = 1 Meitnerbyte
- 1024 Meitnerbytes = 1 Dormstadbyte
- 1024 Dormstadbytes = 1 Teoentbyte

Exercícios

1) No contexto estrutura de dados, pilha é:

- a) Uma lista LILO.
- b) Uma lista FIFO
- c) Um tipo de lista linear em que as operações de inserção e remoção são realizadas aleatoriamente.
- d) Um tipo de lista linear em que o último elemento a ser inserido é o primeiro retirado.
- e) Um tipo de lista linear em que as operações de inserção são realizadas em determinada extremidade e as operações de remoção são realizadas em outra.



Exercícios

2) Avalie as assertivas abaixo e identifique a alternativa correta.

- I. Uma das possíveis utilizações de uma pilha é a implementação da sequência de desfazer (Ctrl + Z) de um editor de texto.
- II. Na estrutura pilha, o último elemento a entrar também é o último a sair.
- III. Na pilha, as operações de exclusão e inclusão são realizadas na mesma extremidade, chamada topo.
- IV. As operações de exclusão e inclusão são realizadas em qualquer parte da pilha.

Assinale a alternativa correta:

- a) Somente I e II estão corretas.
- b) Somente III e IV estão corretas.
- c) Somente I e III estão corretas.
- d) Somente II e IV estão corretas.

Exercícios

3) Indique Verdadeiro ou Falso para cada item:

a) Arrays - uma coleção de elementos do mesmo tipo que podem ser acessados por meio de um índice.

V	F
<input type="checkbox"/>	<input type="checkbox"/>

b) Listas - uma coleção de elementos que podem ser adicionados ou removidos de forma dinâmica.

V	F
<input type="checkbox"/>	<input type="checkbox"/>

c) Pilhas - uma estrutura de dados na qual o último elemento adicionado é o primeiro a ser removido (Last In, First Out - LIFO).

V	F
<input type="checkbox"/>	<input type="checkbox"/>

d) Filas - uma estrutura de dados na qual o primeiro elemento adicionado é o primeiro a ser removido (First In, First Out - FIFO).

V	F
<input type="checkbox"/>	<input type="checkbox"/>

Exercícios

4) Considere a seguinte estrutura de pilha:



Assinale a alternativa que resulta da retirada de dois elementos dessa estrutura:



Exercícios

5) Considere a seguinte estrutura de fila:

início

5	2	6	3	8	4	9
---	---	---	---	---	---	---

Assinale a alternativa que resulta da retirada de dois elementos dessa estrutura:

início

a)

6	3	8	4	9
---	---	---	---	---

início

b)

5	2	6	3	8
---	---	---	---	---

início

c)

5	2	8	4	9
---	---	---	---	---

início

d)

2	6	3	8	4
---	---	---	---	---

Exercícios

1) No contexto estrutura de dados, pilha é:

- a) Uma lista LILO.
- b) Uma lista FIFO
- c) Um tipo de lista linear em que as operações de inserção e remoção são realizadas aleatoriamente.
- d) Um tipo de lista linear em que o último elemento a ser inserido é o primeiro retirado.
- e) Um tipo de lista linear em que as operações de inserção são realizadas em determinada extremidade e as operações de remoção são realizadas em outra.

4) Considere a seguinte estrutura de pilha:



Assinale a alternativa que resulta da retira de dois elementos dessa estrutura:



5) Considere a seguinte estrutura de fila:



Assinale a alternativa que resulta da retira de dois elementos dessa estrutura:



2) Avalie as assertivas abaixo e identifique a alternativa correta.

I. Uma das possíveis utilizações de uma pilha é a implementação da sequência de desfazer (Ctrl + Z) de um editor de texto.

II. Na estrutura pilha, o último elemento a entrar também é o último a sair.

III. Na pilha, as operações de exclusão e inclusão são realizadas na mesma extremidade, chamada topo.

IV. As operações de exclusão e inclusão são realizadas em qualquer parte da pilha.

Assinale a alternativa correta:

- a) Somente I e II estão corretas.
- b) Somente III e IV estão corretas
- c) Somente I e III estão corretas.
- d) Somente II e IV estão corretas.

3) Indique Verdadeiro ou Falso para cada item:

a) Arrays - uma coleção de elementos do mesmo tipo que podem ser acessados por meio de um índice.

☐ V ☐ F

b) Listas - uma coleção de elementos que podem ser adicionados ou removidos de forma dinâmica.

☐ V ☐ F

c) Pilhas - uma estrutura de dados na qual o último elemento adicionado é o primeiro a ser removido (Last In, First Out - LIFO).

☐ V ☐ F

d) Filas - uma estrutura de dados na qual o primeiro elemento adicionado é o primeiro a ser removido (First In, First Out - FIFO).

☐ V ☐ F

As estruturas de dados mais comuns incluem:

1. Arrays - uma coleção de elementos do mesmo tipo que podem ser acessados por meio de um índice.
2. Listas - uma coleção de elementos que podem ser adicionados ou removidos de forma dinâmica.
3. Pilhas - uma estrutura de dados na qual o último elemento adicionado é o primeiro a ser removido (Last In, First Out - LIFO).
4. Filas - uma estrutura de dados na qual o primeiro elemento adicionado é o primeiro a ser removido (First In, First Out - FIFO).
5. Árvores - uma estrutura de dados hierárquica na qual cada nó tem um pai e zero ou mais filhos.
6. Grafos - uma estrutura de dados que consiste em um conjunto de vértices (ou nós) conectados por arestas.

Cada uma dessas estruturas de dados é adequada para diferentes tipos de problemas e é importante escolher a estrutura certa para cada situação, a fim de otimizar a eficiência e a performance do programa.

Links importantes:

https://www.onlinegdb.com/online_c_compiler

https://www.youtube.com/watch?v=u5P_vryX0fo

<http://calhau.dca.fee.unicamp.br/wiki/images/0/01/EstruturasDados.pdf>