

# Dicionários e Tabelas Hash

## Conceitos, complexidade e funcionamento

Prof. Marcelo de Souza

45RPE – Resolução de Problemas com Estruturas de Dados  
Universidade do Estado de Santa Catarina

# Dicionários

## Conceitos básicos



Um **dicionário** armazena entradas compostas por uma **chave** e um **valor**.

- ▶ A chave serve para buscar o registro.
- ▶ O valor armazena o registro associado à chave.

# Dicionários

## Conceitos básicos



Um **dicionário** armazena entradas compostas por uma **chave** e um **valor**.

- ▶ A chave serve para buscar o registro.
- ▶ O valor armazena o registro associado à chave.

Também chamado de **mapa**, **tabela** ou **array associativo**, o dicionário organiza e acessa as entradas pelas suas **chaves**, em vez das suas posições, i.e. a chave é o índice da estrutura.

# Dicionários

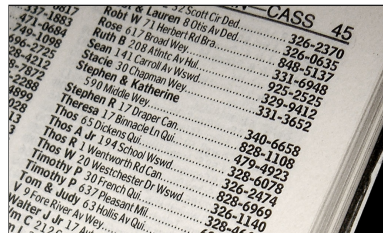
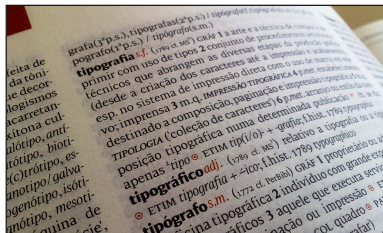
## Conceitos básicos

Um **dicionário** armazena entradas compostas por uma **chave** e um **valor**.

- ▶ A chave serve para buscar o registro.
- ▶ O valor armazena o registro associado à chave.

Também chamado de **mapa**, **tabela** ou **array associativo**, o dicionário organiza e acessa as entradas pelas suas **chaves**, em vez das suas posições, i.e. a chave é o índice da estrutura.

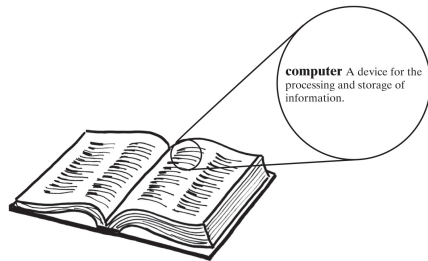
**Exemplos** no cotidiano: dicionários, listas telefônicas, cardápios, índices remissivos, ...





### Características:

- ▶ Na maioria das implementações, **a chave é única**.
  - ▶ Ao inserir uma entrada com chave existente, o valor atual é substituído pelo novo valor.
- ▶ A chave pode ser um objeto de **qualquer classe**.
  - ▶ Necessário garantir a comparação de chaves.



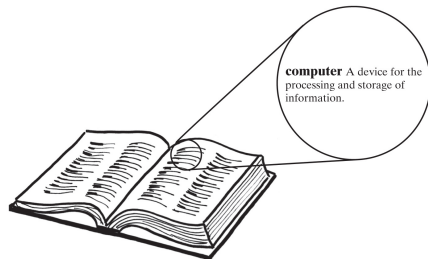


### Características:

- ▶ Na maioria das implementações, **a chave é única**.
  - ▶ Ao inserir uma entrada com chave existente, o valor atual é substituído pelo novo valor.
- ▶ A chave pode ser um objeto de **qualquer classe**.
  - ▶ Necessário garantir a comparação de chaves.

### Operações:

- ▶  $D[k]$ : retorna o valor associado à chave  $k$ .
- ▶  $D[k] = v$ : insere uma entrada com chave  $k$  e valor  $v$ .
- ▶  $D.\text{pop}(k)$ : remove a entrada com chave  $k$ .



# Dicionários

## Análise de complexidade



Podemos implementar um dicionário usando **arranjos** ou **encadeamento**. E ainda podemos manter a estrutura **não ordenada** ou **ordenada**.



Podemos implementar um dicionário usando **arranjos** ou **encadeamento**. E ainda podemos manter a estrutura **não ordenada** ou **ordenada**.

Operação	Arranjo		Encadeamento	
	Não ordenado	Ordenado	Não ordenado	Ordenado
Consulta	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Inserção	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
Remoção	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

— a inserção em um arranjo ordenado é feita em  $\mathcal{O}(\log n)$  na substituição.





Podemos implementar um dicionário usando **arranjos** ou **encadeamento**. E ainda podemos manter a estrutura **não ordenada** ou **ordenada**.

Operação	Arranjo		Encadeamento		Hashing
	Não ordenado	Ordenado	Não ordenado	Ordenado	
Consulta	$\mathcal{O}(n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$
Inserção	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$
Remoção	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$

— a inserção (put) em um arranjo ordenado é feita em  $\mathcal{O}(\log n)$  na substituição.

**Hashing** permite complexidade constante das operações de um mapa no **caso médio**.

- ▶ É a forma como um mapa é implementado na maioria das linguagens.
- ▶ Estudaremos essa estratégia mais adiante!

45RPE – Resolução de Problemas com Estruturas de Dados  
Prof. Marcelo de Souza