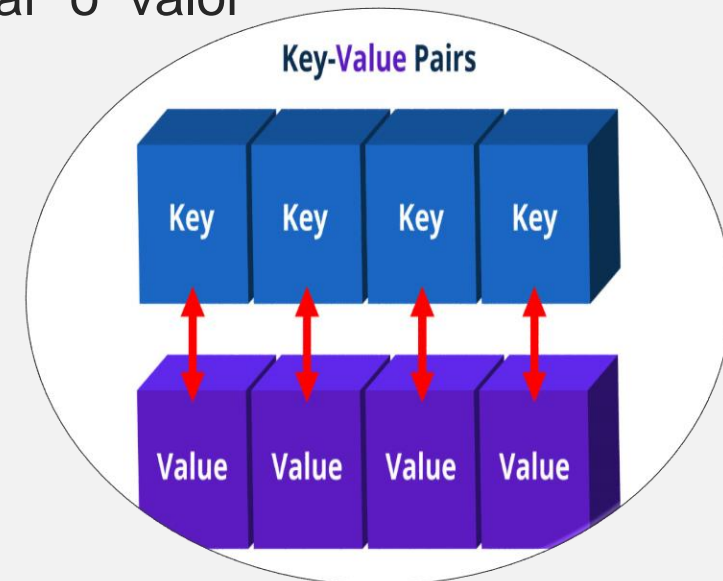


# DESENVOLVIMENTO DE SOFTWARE HASHMAP

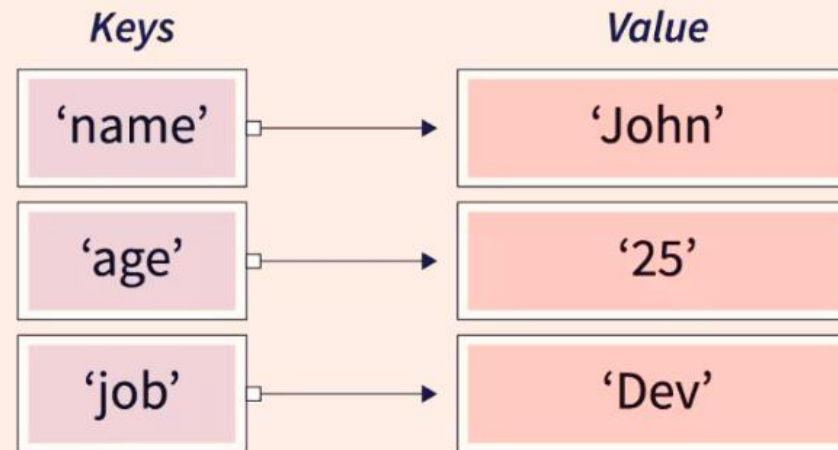
**Guilherme de Moura Xavier;  
David Patrick batista;  
Matheus Antunes.**

# MAP

- Estrutura de Dados que mapeia chaves únicas para valores correspondentes.
- Chave – Valor.
- Chave: É um identificador único usado para acessar o valor associado a ele.
- Valor: É a informação associada à chave.



# What is Java Map?





# HASHMAP

- É uma estrutura de dados em Java que implementa a interface “Map”.
- Uma das diferenças com a Lista é que o HashMap não é ordenado.
- O HashMap é útil porque permite que você armazene dados de maneira eficiente e rápida, permitindo a recuperação dos valores com base em suas chaves.
- Na maioria dos casos a Complexidade Atemporal será  $O(1)$  e em alguns casos raros poderá ser  $O(n)$  devido a colisões.





```
1 import java.util.HashMap;
2
3 public class testando {
4
5     public static void main(String[] args) {
6
7         HashMap<Integer, String> Documento = new HashMap<>();
8
9         Documento.put(19282, "Matheus");
10        Documento.put(19287, "David");
11        Documento.put(19342, "Guilherme");
12
13        System.out.println(Documento);
14    }
15 }
16
17
18
19
```

Problems @ Javadoc Declaration Console ×

<terminated> testando [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Apr 10  
{19282=Matheus, 19287=David, 19342=Guilherme})



# MÉTODOS

- Put();
- Remove();
- Get();
- containsKey(chave):
- containsValue(valor):
- Size();
- isEmpty().



# PUT

- `put(chave, valor)`: Este método adiciona um par de elementos chave-valor ao `HashMap`.
- `Exemplo.put("Nome", 1);`



```
1 import java.util.HashMap;
2
3 public class testando {
4
5     public static void main(String[] args) {
6
7         HashMap<String, Integer> metodoPut = new HashMap<>();
8
9         metodoPut.put("ADICIONANDO", 1);
10        metodoPut.put("ADICIONANDO 2", 2);
11        metodoPut.put("ADICIONANDO 3", 3);
12        metodoPut.put("ADICIONANDO 4", 4);
13
14        System.out.println(metodoPut);
15    }
16 }
17
18
19
20
```

Problems Javadoc Declaration Console ×

<terminated> testando [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Apr 10, 2024)  
{ADICIONANDO 2=2, ADICIONANDO=1, ADICIONANDO 3=3, ADICIONANDO 4=4}





## REMOVE / CLEAR

- `remove(chave)`: Remove a entrada correspondente à chave especificada do `HashMap`, se existir.
- `clear()`: Remove todas as entradas do `HashMap`, deixando-o vazio.
- `Exemplo.remove("Nome");`
- `Exemplo.remove("Nome", 1);`



```
1 import java.util.HashMap;
2
3 public class testando {
4
5     public static void main(String[] args) {
6
7         HashMap<String, Integer> metodoPut = new HashMap<>();
8
9         metodoPut.put("ADICIONANDO", 1);
10        metodoPut.put("ADICIONANDO 2", 2);
11        metodoPut.put("ADICIONANDO 3", 3);
12        metodoPut.put("ADICIONANDO 4", 4);
13        System.out.println(metodoPut);
14        System.out.println("=====");
15
16        //Método de Remoção de um par de elementos do HashMap
17        System.out.println("DEPOIS DA REMOÇÃO");
18        metodoPut.remove("ADICIONANDO", 1);
19        System.out.println(metodoPut);
20    }
21 }
22
23
24
25
```

Problems Javadoc Declaration Console ×

<terminated> testando [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Apr 10, 2024, 9:00:51 PM – 9:00:52 PM) [pi]

{ADICIONANDO 2=2, ADICIONANDO=1, ADICIONANDO 3=3, ADICIONANDO 4=4}

=====

DEPOIS DA REMOÇÃO

{ADICIONANDO 2=2, ADICIONANDO 3=3, ADICIONANDO 4=4}



## GET

- `get(chave)`: Retorna o valor associado à chave especificada, ou null se a chave não estiver presente no HashMap
- `Exemplo.get("Nome")`;



```
3 import java.util.HashMap;
4
5 public class Get {
6     public static void main(String[] args) {
7         // Criando um HashMap
8         HashMap<Integer, String> documento = new HashMap<>();
9
10        // Adicionando informações corretamente com vírgula entre a chave e o valor
11        documento.put(19282, "Matheus");
12        documento.put(19287, "David");
13        documento.put(19342, "Guilherme");
14
15        // Obtendo o valor associado à chave "19287"
16        String nome = documento.get(19287);
17        System.out.println("O número desse passaporte pertence a: " + nome);
18
19        // Obtendo o valor associado à chave "19111" (que não existe no HashMap)
20        String passport = documento.get(19111);
21        if (passport != null) {
22            System.out.println("O número desse passaporte pertence a: " + passport);
23        } else {
24            System.out.println("Não há informações para esse passaporte no HashMap.");
25        }
26    }
27 }
28
```

Problems Javadoc Declaration Console X

<terminated> Get [Java Application] C:\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.9.v20231028-0858\jre\bin\java

O número desse passaporte pertence a: David

Não há informações para esse passaporte no HashMap.



## CONTAINSKEY

- `containsKey(chave)`: retorna true se o HashMap contiver uma entrada para a chave especificada, caso contrário retorna false.
- `Exemplo.containsKey("Nome");`



```
1 import java.util.HashMap;
2
3 public class testando {
4
5     public static void main(String[] args) {
6
7         HashMap<String, Integer> metodoPut = new HashMap<>();
8
9         metodoPut.put("ADICIONANDO", 1);
10        metodoPut.put("ADICIONANDO 2", 2);
11        metodoPut.put("ADICIONANDO 3", 3);
12        metodoPut.put("ADICIONANDO 4", 4);
13        System.out.println(metodoPut);
14        System.out.println("=====
15
16
17        String keyToSearch = "ADICIONANDO";
18        metodoPut.containsKey(keyToSearch);
19        if (metodoPut.containsKey(keyToSearch)) {
20            System.out.println("Chave Existe");
21        } else {
22            System.err.println("Chave não existe");
23        }
24
25    }
26 }
```

Problems Javadoc Declaration Console ×

<terminated> testando [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Apr 10, 20

{ADICIONANDO 2=2, ADICIONANDO=1, ADICIONANDO 3=3, ADICIONANDO 4=4}

=====

Chave Existe



## CONTAINSVALUE

- `containsValue(valor)`: Retorna true se o HashMap contiver uma ou mais chaves associadas ao valor especificado, caso contrário retorna false.
- `Exemplo.containsValue(1);`



```
1 import java.util.HashMap;
2 import java.util.Scanner;
3
4 public class testando {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         HashMap<String, Integer> metodoPut = new HashMap<>();
9
10        metodoPut.put("Arroz", 1);
11        metodoPut.put("Feijao", 2);
12        metodoPut.put("Maionese", 3);
13        metodoPut.put("Alface", 4);
14        metodoPut.put("Tomate", 5);
15
16
17        System.out.println(metodoPut);
18
19        System.out.println("-----");
20
21        Integer value = 10;
22        if(metodoPut.containsValue(value)) {
23            System.out.println("Este valor está no HashMap");
24        }
25        else {
26            System.out.println("Este valor não está no HashMap");
27        }
28    }
29 }
30
```

Problems Javadoc Declaration Console ×

<terminated> testando [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Apr 11, 2024, 4:31:33 PM -

{Maionese=3, Alface=4, Arroz=1, Tomate=5, Feijao=2}

-----

Este valor não está no HashMap





## SIZE

- `size()`: Retorna o número de pares chave-valor presentes no `HashMap`.
- `Exemplo.size()`;



```
1 import java.util.HashMap;
2 import java.util.Scanner;
3
4 public class testando {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         HashMap<String, Integer> metodoPut = new HashMap<>();
9
10        metodoPut.put("Arroz", 1);
11        metodoPut.put("Feijao", 2);
12        metodoPut.put("Maionese", 3);
13        metodoPut.put("Alface", 4);
14        metodoPut.put("Tomate", 5);
15
16        System.out.println(metodoPut);
17
18        System.out.println("-----");
19
20        int tamanho = metodoPut.size();
21
22        System.out.println("Tamanho do HashMap: " + tamanho);
23
24    }
25 }
26
```

Problems Javadoc Declaration Console ×

```
<terminated> testando [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Apr 11, 2024, 4:33:35 PM)
{Maionese=3, Alface=4, Arroz=1, Tomate=5, Feijao=2}
-----
Tamanho do HashMap: 5
```



## ISEMPTY

- isEmpty(): Retorna true se o HashMap estiver vazio, caso contrário retorna false.



```
1 import java.util.HashMap;
2
3 public class testando {
4
5     public static void main(String[] args) {
6
7         HashMap<String, Integer> metodoPut = new HashMap<>();
8
9         metodoPut.put("ADICIONANDO", 1);
10        metodoPut.put("ADICIONANDO 2", 2);
11        metodoPut.put("ADICIONANDO 3", 3);
12        metodoPut.put("ADICIONANDO 4", 4);
13        System.out.println(metodoPut);
14        System.out.println("=====");
15
16
17
18        if (metodoPut.isEmpty()) {
19            System.out.println("HashMap está Vazio");
20        } else {
21            System.err.println("HashMap não está vazio");
22        }
23    }
24 }
25
26
```

Problems Javadoc Declaration Console X

<terminated> Testando [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Apr 10, 2024)

```
{ADICIONANDO 2=2, ADICIONANDO=1, ADICIONANDO 3=3, ADICIONANDO 4=4}
=====
HashMap não está vazio
```

# HASHMAP

- Para percorrer todos os elementos de um HashMap podemos utilizar o foreach.
- Como no exemplo a seguir:

```
for (String key : metodoPut.keySet()) {  
    Integer value = metodoPut.get(key);  
    System.out.println(key + " - " + value);  
}
```



```
1 import java.util.HashMap;
2 import java.util.Scanner;
3
4 public class testando {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         HashMap<String, Integer> metodoPut = new HashMap<>();
9
10        metodoPut.put("Arroz", 1);
11        metodoPut.put("Feijao", 2);
12        metodoPut.put("Maionese", 3);
13        metodoPut.put("Alface", 4);
14        metodoPut.put("Tomate", 5);
15
16        System.out.println(metodoPut);
17
18        System.out.println("-----");
19
20        for (String key : metodoPut.keySet()) {
21            Integer value = metodoPut.get(key);
22            System.out.println(key + " - " + value);
23        }
24
25    }
26 }
27
```

Problems Javadoc Declaration Console ×

<terminated> testando [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Apr 11, 2024, 4:38:55 PM -

{Maionese=3, Alface=4, Arroz=1, Tomate=5, Feijao=2}

-----

Maionese - 3

Alface - 4

Arroz - 1

Tomate - 5

Feijao - 2



## EXERCÍCIOS

- 1 - Escreva um programa que receba uma string como entrada e conte quantas vezes cada palavra aparece. Utilize um HashMap para armazenar as palavras como chaves e as contagens como valores.
- 2 - Crie um programa de agenda telefônica que permita adicionar, remover e pesquisar contatos pelo nome. Utilize um HashMap onde as chaves sejam os nomes e os valores sejam os números de telefone.
- 3 - Escreva um programa que registre a pontuação de diferentes jogadores em um jogo. Utilize um HashMap onde as chaves sejam os nomes dos jogadores e os valores sejam suas pontuações.



FIM

- David Patrick Batista;
- Guilherme de Moura Xavier;
- Matheus Antunes.