

## **Atividade 4 – Lista de Mapa**

### **Luna Costa Vasconcelos**

#### **01) O que é um mapa em Dart?**

Um mapa em Dart é uma coleção de pares chave/valor, em que cada chave é distinta e se traduz em um único valor. Em outras linguagens de programação, um dicionário é comparável a ele. Qualquer tipo de objeto pode servir como chave ou valor. A classe 'Map' no Dart implementa os mapas.

#### **02) Como criar um mapa vazio em Dart?**

Podemos declarar um mapa vazio das seguintes formas em Dart:

```
Map<String, dynamic> mapa1 = {};  
var mapa2 = Map<String, dynamic>();  
Map<String, dynamic> mapa3 = new Map();
```

#### **03) Como criar um mapa com elementos em Dart?**

Podemos usar a inicialização por meio de chaves {} para declarar um mapa com elementos, da seguinte forma:

```
Map<String, int> mapa = {  
  'um': 1,  
  'dois': 2,  
  'três': 3,  
};
```

#### **04) Qual a diferença entre uma lista e um mapa em Dart?**

A lista é um agrupamento ordenado de componentes de tipos idênticos que podem ser acessados usando seus respectivos índices numéricos. O índice da lista varia de zero ao comprimento da lista menos um. Para armazenar e manipular dados sequenciais, como números inteiros, caracteres ou objetos, são utilizadas listas. Enquanto o mapa é um grupo de pares chave-valor, onde cada valor está conectado a uma determinada chave. Qualquer tipo de valor e qualquer tipo de chave pode ser usado como uma chave em um mapa. Os dados podem ser convenientemente armazenados e recuperados usando mapas e uma chave.

#### **05) Como adicionar um elemento a um mapa em Dart?**

Podemos usar o próprio operador de índice do mapa para adicionar um novo elemento, da seguinte forma:

```
Map<String, int> meuMapa = {'Maria': 10, 'Ana': 5, 'Carlos': 3};  
meuMapa['João'] = 7;
```

O resultado será um novo par chave-valor adicionado ao final do mapa com o elemento dentro do [] sendo a chave e o valor fornecido pela igualdade:

```
{'Maria': 10, 'Ana': 5, 'Carlos': 3, 'João': 7}
```

#### **06) Como remover um elemento de um mapa em Dart?**

Podemos usar o método remove() para isso, ele irá aceitar como parâmetro a chave que deve ser removida, por exemplo:

```
Map<String, int> meuMapa = {'Maria': 10, 'Ana': 5, 'Carlos': 3};  
meuMapa.remove('Carlos');
```

O resultado será o mapa sem o par chave-valor informado pelo parâmetro fornecido ao método remove():

```
{'Maria': 10, 'Ana': 5}
```

### 07) Como verificar se um mapa contém uma determinada chave em Dart?

Podemos usar o método 'containsKey' da classe Map. O método containsKey aceita um parâmetro que é a chave a ser verificada e retorna true se a chave existir no mapa, caso contrário, retorna false. Por exemplo:

```
Map<String, int> pontos = {'Alice': 10, 'Bob': 5, 'Charlie': 3};
if (pontos.containsKey('Alice')) {
  print('O mapa contém o nome "Alice".');
} else {
  print('O mapa não contém o nome "Alice".');
}
// Saída: O mapa contém o nome "Alice".
```

### 08) Como verificar se um mapa é vazio em Dart?

Podemos usar a propriedade isEmpty da classe Map, que irá funcionar de forma semelhante ao método de mesmo nome existente nas listas e conjuntos. A propriedade isEmpty retorna true se o mapa não tiver nenhum par de chave-valor e false caso contrário. Por exemplo:

```
Map<String, int> pontos = {};
if (pontos.isEmpty) {
  print('O mapa está vazio.');
```

```
} else {
```

```
  print('O mapa não está vazio.');
```

```
}
```

```
// Saída: O mapa está vazio.
```

### 09) Como acessar o valor de uma chave em um mapa em Dart?

Para acessar o valor associado a uma chave em um mapa em Dart, você pode usar a notação de colchetes [] ou o método mapName.containsKey(key). Ambos retornam o valor correspondente à chave especificada e não apresentam grandes diferenças de performance

```
void main() {

  Map<String, int> meuMapa = {'Maria': 10, 'Ana': 5, 'Carlos': 3};

  // Acessando o valor da chave 'Maria' usando a notação de colchetes
  int? pontosMaria = meuMapa['Maria'];
  print('A Maria tem $pontosMaria pontos.');
```

```
  // Verificando se a chave 'Ana' existe usando o método containsKey
```

```
  if (meuMapa.containsKey('Ana')) {
```

```
    int? pontosAna = meuMapa['Ana'];
```

```
    print('A Ana tem $pontosAna pontos.');
```

```
  } else {
```

```
    print('A chave Ana não existe no mapa.');
```

```
  }
```

```
  // Iterando sobre todos os pares chave-valor usando o método forEach
```

```
  meuMapa.forEach((chave, valor) {
```

```
    print('A chave $chave tem o valor $valor');
```

```
  }));
```

```
}
```

```
// Saída:  
A Maria tem 10 pontos.  
A Ana tem 5 pontos.  
A chave Maria tem o valor 10  
A chave Ana tem o valor 5  
A chave Carlos tem o valor 3
```

### 10) Como alterar o valor de uma chave em um mapa em Dart?

Para alterar o valor associado a uma chave em um mapa, podemos simplesmente utilizar a notação de colchetes [] e atribuir o novo valor à chave desejada. Por exemplo:

```
Map<String, int> meuMapa2 = {'Maria': 10, 'Ana': 5, 'Carlos': 3};  
// Acessando a chave e passando o novo valor  
meuMapa2['Maria'] = 2;  
print("Mapa com novo valor para Maria: $meuMapa2");  
// Saída: Mapa com novo valor para Maria: {Maria: 2, Ana: 5, Carlos: 3}
```

### 11) Como obter todas as chaves de um mapa em Dart?

Podemos usar o método keys, que retorna um objeto iterável que contém todas as chaves do mapa. Por exemplo:

```
void main() {  
  Map<String, int> meuMapa = {'Maria': 10, 'Ana': 5, 'Carlos': 3};  
  var chaves = meuMapa.keys;  
  print(chaves);  
}  
// Saída: (Maria, Ana, Carlos)
```

### 12) Como obter todos os valores de um mapa em Dart?

Podemos usar o método values, que retorna um objeto iterável que contém todos os valores do mapa. Por exemplo:

```
void main() {  
  Map<String, int> meuMapa = {'Maria': 10, 'Ana': 5, 'Carlos': 3};  
  var valores = meuMapa.values;  
  print(valores);  
}  
// Saída: (10, 5, 3)
```

### 13) Como verificar se duas chaves em um mapa são iguais em Dart?

Em um mapa duas chaves não podem ser iguais, isso causaria uma inconsistência na declaração e esse mapa nem poderia ser criado, podemos, no entanto, verificar se duas chaves possuem o mesmo valor, podemos fazer isso da seguinte forma:

Utilizando a propriedade values do mapa para obter uma lista com os valores e, em seguida, comparar as chaves dos valores nas posições desejadas da lista. Por exemplo, vamos verificar se a primeira e a segunda chaves de um mapa possuem valores iguais:

```
Map<String, int> meuMapa = {'a': 1, 'b': 2, 'c': 2};  
if (meuMapa.values.elementAt(0) == meuMapa.values.elementAt(1)) {  
  print('A primeira e a segunda chaves possuem valores iguais.');
```

```
} else {  
  print('A primeira e a segunda chaves possuem valores diferentes.');
```

```
}  
// Saída: A primeira e a segunda chaves possuem valores iguais.
```

#### 14) Como criar um mapa a partir de duas listas em Dart?

Para criar um mapa a partir de duas listas em Dart, podemos usar o construtor da classe Map que recebe uma lista de chaves e uma lista de valores como parâmetros. As listas devem ter o mesmo comprimento para que o mapa possa ser criado corretamente. Por exemplo:

```
List<String> chaves = ['a', 'b', 'c'];
List<int> valores = [1, 2, 3];
Map<String, int> meuMapa = Map.fromIterables(chaves, valores);
print(meuMapa);
// Saída: {a: 1, b: 2, c: 3}
```

#### 15) Como criar uma lista de chaves a partir de um mapa em Dart?

Podemos usar a mesma propriedade usada para pegar as chaves em Dart, a propriedade 'keys', a qual retorna um Iterable, e passar isso para uma lista, da seguinte forma:

```
Map<String, int> meuMapa = {'a': 1, 'b': 2, 'c': 3};
List<String> minhasChaves = meuMapa.keys.toList();
print(minhasChaves);
// Saída: [a, b, c]
```

#### 16) Como criar uma lista de valores a partir de um mapa em Dart?

Podemos fazer o mesmo que a questão anterior para os valores, trocando apenas a propriedade, nesse caso será a values, para que possamos pegar os valores em Iterable do mapa, segue o exemplo:

```
Map<String, int> meuMapa = {'a': 1, 'b': 2, 'c': 3};
List<int> meusValores = meuMapa.values.toList();
print(meusValores);
// Saída: [1, 2, 3]
```

#### 17) Como transformar um mapa em uma lista de pares chave-valor em Dart?

É possível transformar um mapa em uma lista de listas usando o método map() juntamente com a classe List. Podemos criar uma nova lista para cada par chave-valor do mapa, onde o primeiro elemento da lista é a chave e o segundo elemento é o valor. Por exemplo:

```
Map<String, int> meuMapa = {'a': 1, 'b': 2, 'c': 3};
List<List<dynamic>> listaDeListas = meuMapa.entries.map((e) => [e.key,
e.value]).toList();
print(listaDeListas);
// Saída: [[a, 1], [b, 2], [c, 3]]
```

#### 18) Como remover todos os elementos de um mapa em Dart?

Podemos utilizar o método clear(). O método clear() remove todos os pares chave-valor do mapa, deixando-o vazio. Por Exemplo:

```
Map<String, int> mapa = {'a': 1, 'b': 2, 'c': 3};
print(mapa); // {a: 1, b: 2, c: 3}
mapa.clear();
print(mapa);
//Saída: {}
```

#### 19) Como calcular o tamanho de um mapa em Dart?

Podemos utilizar a propriedade length. A propriedade length retorna o número de pares chave-valor presentes no mapa. Por exemplo:

```
Map<String, int> mapa = {'a': 1, 'b': 2, 'c': 3};
```

```
print(mapa.length);  
// Saída: 3
```

## 20) Como verificar se dois mapas são iguais em Dart?

Podemos unir alguns métodos e propriedades para fazer essa verificação, abaixo segue um método criado para fazer isso juntamente com um exemplo:

```
bool saoIguais(Map map1, Map map2) {  
    if (map1.length != map2.length) return false; // se os mapas têm tamanhos  
    diferentes, eles não são iguais  
    for (var key in map1.keys) {  
        if (!map2.containsKey(key)) return false; // se uma chave não está  
        presente no segundo mapa, eles não são iguais  
        if (map1[key] != map2[key]) return false; // se o valor de uma chave é  
        diferente no segundo mapa, eles não são iguais  
    }  
    return true; // se não houve diferenças, eles são iguais  
}  
  
Map<String, int> mapa1 = {'a': 1, 'b': 2, 'c': 3};  
Map<String, int> mapa2 = {'a': 1, 'b': 2, 'c': 3};  
Map<String, int> mapa3 = {'a': 1, 'd': 2, 'c': 4};  
  
print(saoIguais(mapa1, mapa2));  
// Saída: true  
print(saoIguais(mapa1, mapa3));  
// Saída: false
```