

Responda às seguintes questões:

### **01) O que é um conjunto em Dart?**

Em Dart, um conjunto é uma coleção de elementos únicos e não ordenados. Cada elemento em um conjunto é único e aparece apenas uma vez na coleção.

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

Para criar um conjunto vazio em Dart, pode-se usar a seguinte sintaxe: `var conjuntoVazio = <tipo>{};`. Substitua `<tipo>` pelo tipo de dados dos elementos que o conjunto irá armazenar. Por exemplo, um conjunto vazio que armazena números inteiros, use `var conjuntoVazio = <int>{};`.

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

Para criar um conjunto com elementos em Dart, pode-se usar a seguinte sintaxe: `var conjunto = <tipo>{elemento1, elemento2, elemento3};`. Substitua `<tipo>` pelo tipo de dados dos elementos que o conjunto irá armazenar e inclua os elementos que deseja adicionar, separados por vírgulas.

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

A diferença entre um conjunto e uma lista em Dart é que um conjunto é uma coleção de elementos únicos e não ordenados, enquanto uma lista é uma coleção ordenada de elementos que podem ser duplicados.

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

Para adicionar um elemento a um conjunto em Dart, pode-se usar o método `add()`. Por exemplo, se tiver um conjunto vazio e quiser adicionar o número 5 a ele, use-se a seguinte sintaxe: `conjunto.add(5);`.

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

Para remover um elemento de um conjunto em Dart, pode-se usar o método `remove()`. Por exemplo, se tiver um conjunto com o número 5 e quiser removê-lo, use a seguinte sintaxe: `conjunto.remove(5);`.

### **07) Como verificar se um conjunto contém um determinado elemento em Dart?**

Para verificar se um conjunto contém um determinado elemento em Dart, pode-se usar o método `contains()`. Por exemplo, se tiver um conjunto com o número 5 e quiser verificar se ele

contém o número 5, usa-se a seguinte sintaxe: `conjunto.contains(5);`. Este método retorna `true` se o conjunto contém o elemento e `false` caso contrário.

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

Para verificar se um conjunto é vazio em Dart, pode-se usar a propriedade `isEmpty`. Por exemplo, se tiver um conjunto e quiser verificar se ele está vazio, usa-se a seguinte sintaxe: `conjunto.isEmpty;`. Esta propriedade retorna `true` se o conjunto estiver vazio e `false` caso contrário.

#### **09) Como unir dois conjuntos em Dart?**

Para unir dois conjuntos em Dart, pode-se usar o operador `union()`. Por exemplo, se tiver dois conjuntos `conjunto1` e `conjunto2`, usa-se a seguinte sintaxe: `var conjuntoUnido = conjunto1.union(conjunto2);`. Isso irá criar um novo conjunto que contém todos os elementos únicos de `conjunto1` e `conjunto2`.

#### **10) Como obter a interseção de dois conjuntos em Dart?**

Para obter a interseção de dois conjuntos em Dart, pode-se usar o método `intersection()`. Por exemplo, se tiver dois conjuntos `conjunto1` e `conjunto2`, use a seguinte sintaxe: `var intersecao = conjunto1.intersection(conjunto2);`. Isso irá criar um novo conjunto que contém apenas os elementos que estão presentes em ambos os conjuntos.

#### **11) Como obter a diferença entre dois conjuntos em Dart?**

Para obter a diferença entre dois conjuntos em Dart, pode-se usar o método `difference()`. Por exemplo, se tiver dois conjuntos `conjunto1` e `conjunto2`, use a seguinte sintaxe: `var diferenca = conjunto1.difference(conjunto2);`. Isso irá criar um novo conjunto que contém apenas os elementos presentes em `conjunto1` e não em `conjunto2`.

#### **12) Como verificar se um conjunto é subconjunto de outro conjunto em Dart?**

Para verificar se um conjunto é subconjunto de outro conjunto em Dart, pode-se usar o método `containsAll()`. Por exemplo, se tiver dois conjuntos `conjunto1` e `conjunto2` e quiser verificar se `conjunto2` é um subconjunto de `conjunto1`, use a seguinte sintaxe: `conjunto1.containsAll(conjunto2);`. Este método retorna `true` se todos os elementos de `conjunto2` estiverem presentes em `conjunto1`.

#### **13) Como verificar se dois conjuntos são iguais em Dart?**

Para verificar se dois conjuntos são iguais em Dart, pode-se usar o método

`containsAll()`. Por exemplo, se você tiver dois conjuntos `conjunto1` e `conjunto2` e quiser verificar se eles são iguais, use a seguinte sintaxe: `conjunto1.containsAll(conjunto2) && conjunto2.containsAll(conjunto1);`. Este método verifica se todos os elementos de `conjunto1` estão presentes em `conjunto2` e se todos os elementos de `conjunto2` estão presentes em `conjunto1`.

#### **14) Como criar um conjunto a partir de uma lista em Dart?**

Para criar um conjunto a partir de uma lista em Dart, pode-se usar o construtor `Set.from()`. Por exemplo, se tiver uma lista de números inteiros chamada `listaNumeros`, usa-se a seguinte sintaxe: `var conjuntoNumeros = Set.from(listaNumeros);`. Isso irá criar um novo conjunto que contém todos os elementos únicos da lista.

#### **15) Como criar uma lista a partir de um conjunto em Dart?**

Para criar uma lista a partir de um conjunto em Dart, pode-se usar o construtor `List.from()`. Por exemplo, se tiver um conjunto de números inteiros chamado `conjuntoNumeros`, usa-se a seguinte sintaxe: `var listaNumeros = List.from(conjuntoNumeros);`. Isso irá criar uma nova lista que contém todos os elementos do conjunto, preservando a ordem em que foram adicionados.

#### **16) Como transformar um conjunto em uma lista de strings em Dart?**

Para transformar um conjunto em uma lista de strings em Dart, pode-se usar o método `map()`. Por exemplo, se tiver um conjunto de números inteiros chamado `conjuntoNumeros`, usa-se a seguinte sintaxe: `var listaStrings = conjuntoNumeros.map((numero) => numero.toString()).toList();`. Isso irá criar uma nova lista de strings que contém uma representação em string de cada elemento do conjunto.

#### **17) Como calcular a união de vários conjuntos em Dart?**

Para calcular a união de vários conjuntos em Dart, pode-se usar o método `union()` em conjunto com o operador `reduce()`. Por exemplo, se tiver uma lista de conjuntos chamada `conjuntos` e quiser calcular a união de todos eles, use a seguinte sintaxe: `var uniao = conjuntos.reduce((conjunto1, conjunto2) => conjunto1.union(conjunto2));`. Isso irá criar um novo conjunto que contém todos os elementos únicos presentes em qualquer um dos conjuntos.

#### **18) Como calcular a interseção de vários conjuntos em Dart?**

Para calcular a interseção de vários conjuntos em Dart, pode-se usar o método `reduce()` em conjunto com o método `intersection()`. Por exemplo, se tiver uma lista de conjuntos chamada `conjuntos` e quiser calcular a interseção de todos eles, usa-se a seguinte sintaxe: `var intersecao = conjuntos.reduce((conjunto1, conjunto2) => conjunto1.intersection(conjunto2));`.

Isso irá criar um novo conjunto que contém apenas os elementos que estão presentes em todos os conjuntos.

### **19) Como verificar se dois conjuntos são disjuntos em Dart?**

Para verificar se dois conjuntos são disjuntos em Dart, basta verificar se a interseção entre eles é vazia. Por exemplo, se tiver dois conjuntos `conjunto1` e `conjunto2`, use a seguinte sintaxe: `var disjuntos = conjunto1.intersection(conjunto2).isEmpty;`. Isso irá retornar `true` se os conjuntos são disjuntos, ou seja, se não tiverem nenhum elemento em comum.

### **20) Como remover todos os elementos duplicados de uma lista usando um conjunto em Dart?**

Para remover todos os elementos duplicados de uma lista usando um conjunto em Dart, pode-se criar um conjunto a partir da lista para obter apenas os elementos únicos e, em seguida, criar uma nova lista a partir do conjunto. Por exemplo, se tiver uma lista de números inteiros com elementos duplicados chamada `listaNumeros`, usa-se a seguinte sintaxe: `var listaSemDuplicatas = List.from(Set.from(listaNumeros));`. Isso irá criar uma nova lista que contém apenas os elementos únicos da lista original, preservando a ordem em que foram adicionados.