

USO DE FUNÇÕES

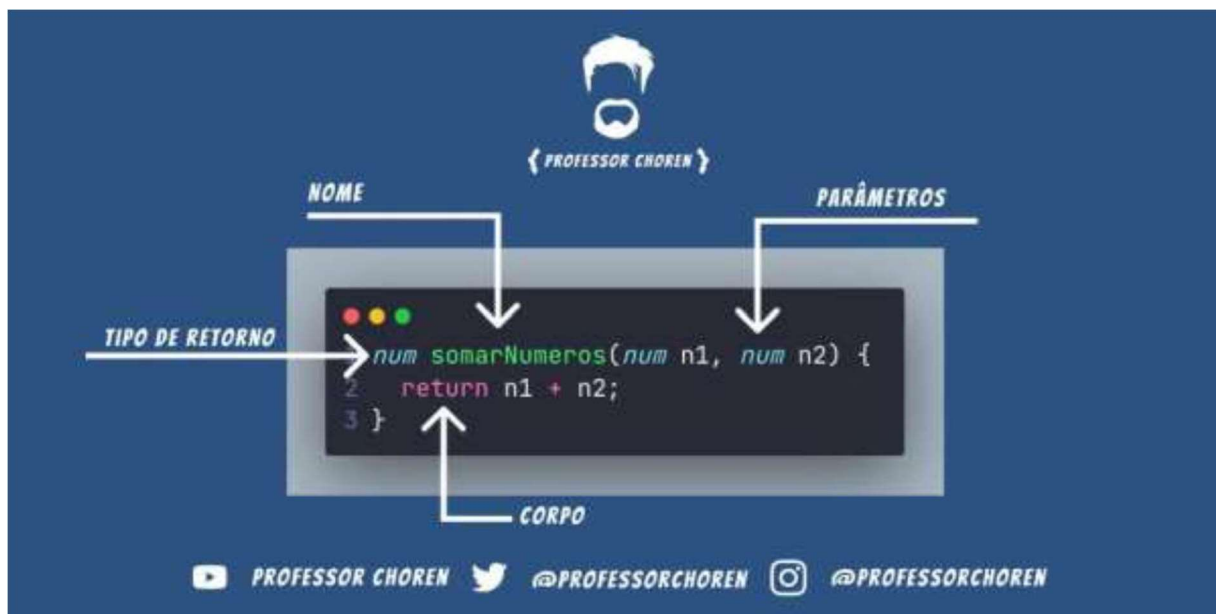
Funções são formas de declarar blocos de comandos, os quais podem ser acionados através de uma declaração de chamada. Assim determinando que desejamos executar tal bloco de comandos.

Declaração de métodos

Ao declararmos um método temos que ficar atento a sua declaração. Um método é formado por três partes fundamentais, sendo elas:

- **Nome**
- **Parâmetros**
- **Tipo de retorno**

Na figura podemos ver um diagrama apresentado as partes que compõem um método no Dart.



É importante ressaltar que um método pode ou não possuir parâmetros, tudo dependerá da finalidade do método e de sua necessidade de informações externas a seu escopo.

Declaração de parâmetros

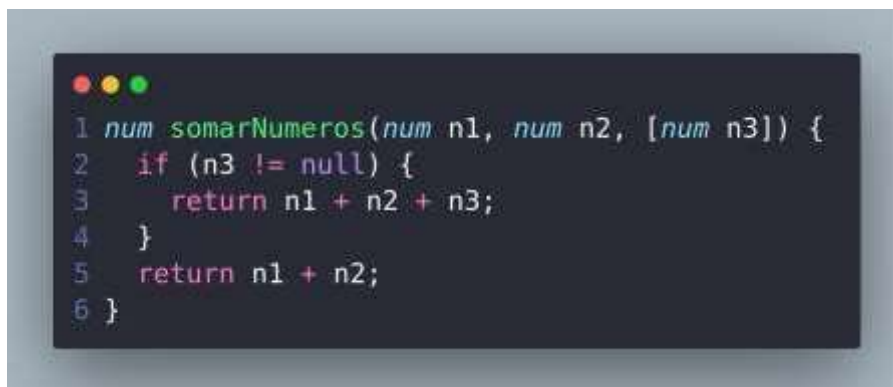
Como vimos anteriormente, um método pode ou não possuir parâmetros. Isso dependerá da razão de sua existência e da necessidade de informações externas ao seu

escopo. Porém existem alguns aspectos importantes a serem tratados sobre o uso de parâmetros, vamos a eles.

Parâmetros opcionais

No exemplo apresentado na figura acima vimos a declaração de um método para a soma de dois números, a qual recebe como parâmetro dois valores numéricos, porém ambos são obrigatórios, ou seja, ambos devem obrigatoriamente ser informados, caso contrário sua execução acarretará um erro em tempo de compilação. Mas e se necessitarmos declarar parâmetros opcionais, ou seja, que podem ou não ser informadas no momento da chamada do método.

Para esse exemplo, vamos recriar nosso método de soma, agora para que ela permita um terceiro número, mas este sendo opcional.

A screenshot of a code editor with a dark background and light-colored text. The code is written in Dart and defines a method named 'somarNumeros'. The method signature is 'num somarNumeros(num n1, num n2, [num n3])', where the third parameter '[num n3]' is enclosed in square brackets to indicate it is optional. The method body contains an 'if' statement: 'if (n3 != null) { return n1 + n2 + n3; }'. If the condition is false, it returns 'n1 + n2'. The code is numbered from 1 to 6 on the left side of the editor.

```
1 num somarNumeros(num n1, num n2, [num n3]) {  
2   if (n3 != null) {  
3     return n1 + n2 + n3;  
4   }  
5   return n1 + n2;  
6 }
```

Como podemos ver no exemplo, utilizamos um par de colchetes “[]” para declarar que o terceiro parâmetro é opcional. No corpo do método incluímos uma condicional IF para verificar se devemos utilizar o terceiro parâmetro ou não. Dessa forma evitando um erro em tempo de execução devido ao fato do parâmetro estar nulo.

Parâmetros nomeados

Outro aspecto interessante sobre o Dart, é a possibilidade de nomear os parâmetros de um método. Dessa forma podemos informar nossos parâmetros na ordem em que desejarmos, ou até mesmo não informá-los, de forma a criar o mesmo efeito do parâmetro opcional. Mas antes de entendermos como podemos fazer isso, é importante entender como o Dart interpreta os parâmetros de um método.

O Dart, por padrão, faz uso da declaração posicional de parâmetros. Ou seja, ao chamarmos um método, devemos passar seus parâmetros na mesma ordem em que eles foram declarados na criação do mesmo, pois caso contrário, o Dart irá colocar os valores

nos locais errados e isso acarretará em um erro na execução do método. Vejamos um exemplo.

```
1 void main() {
2   // Na ordem correta
3   print(imprimirSaudacao("Anderson","M"));
4   // Com parâmetros invertidos
5   print(imprimirSaudacao("F","Michele"));
6 }
7
8 String imprimirSaudacao(String nome, String genero) {
9   if (genero == "M") {
10    return "Bem-vindo Sr. $nome";
11  }
12  return "Bem-vindo Sra. $nome";
13 }
14
```

Ao fazer uso de parâmetros nomeados, podemos dar nomes a cada um dos parâmetros do método, dessa forma podemos informá-los na ordem em que desejarmos, pois o Dart irá associar o nome do parâmetro e não sua posição declarativa. Vejamos como podemos mudar nosso método para aceitar parâmetros nomeados.

```
1 void main() {
2   print(imprimirSaudacao(nome: "Anderson", genero: "M"));
3   print(imprimirSaudacao(genero: "F", nome: "Michele"));
4 }
5
6 String imprimirSaudacao({String nome, String genero}) {
7   if (genero == "M") {
8     return "Bem-vindo Sr. $nome";
9   }
10  return "Bem-vinda Sra. $nome";
11 }
12
```

Você já deve ter notado que no momento da declaração dos parâmetros de nossa função, colocamos os mesmos dentro um par de chaves "{}". Isso informa para o Dart que nossos parâmetros nome e gênero são nomeados. Assim podemos passá-los na ordem em que desejarmos.

É importante ressaltar que todos os parâmetros nomeados são considerados pelo Dart como opcionais. Dessa forma, eles podem ou não ser informados na chamados do método. Dessa forma, é necessário ter cuidado com seu uso, pois podemos ter problemas tanto de execução, quanto de lógica, em nossas aplicações.