1 Definir classe para **calculadora**

* No construtor definir argumentos com nomes das constantes “digit”
* Definir propriedade (currentOperation) para zerar o valor da string

4 Criar método para mostrar **dígitos** na tela e passar uma propriedade **digito**

* Fazer uma verificação lógica para saber se o digito é um “.” e se o (currentDigit) possui “.” Incluso no texto
* Abaixo atribuir à propriedade (currentOperation) o valor do **digito** acima
* Abaixo chamar o método (this.updateScreen) para atualizar a tela.

7 Criar método para as **operações** epasse como propriedade (**operação**)

* 13 verificar se o texto inserirdo no (currentDigit) desta classe está vazio e diferente da operação “C”, se sim, fazer outra verificação lógica para caso o texto inserirdo no (currentDigit) desta classe seja diferente de vazio, se for diferente chamar o método changeOperation e passar como parâmetro a **operação**
* 9 Crie 3 variaveis, uma para o **valor da operação**, outra para o número anterior (mainDigit) e outra para o (currentDigit). No mainDigit aplicar a funcionalidade split para separar a primeira string
* 10 aplicar o switch para identificar as **operações,** caso seja “+, - , \*, / etc”, o **valor da operação** é igual o (mainDigit) e “+, - , \*, / etc”, pelo (currentDigit) **desta** classe.
  + 15 Criar um retorno no switch caso a **operação** seja “DEL, CE, C, =”, cada símbolo vai retornar um método desta classe
* 11 Logo abaixo das somas chamar o método (updateScreen) **desta** classe e dentor desta declaração definir 4 argumentos que seriam **(valor da operação, operação, (currentDigit), (mainDigit))**

5 Criar método updateScreen e passe como propriedade (**valor da operação, operação, currentDigit e mainDigit)** deixe todos os valores inicialmente como null

* Fazer uma verificação lógica para se o **valor da operação** for nulo, se sim, concatenar com += este (currentDigit) inserido no texto ao valor do (currentOperation) desta classe. Se o mainDigit for 0 atribuir ao **valor da operação** o (**currentDigit)**
* 12 para passar o currentDigit para o mainDigit adicionar ao texto inserido desta (maniDigit) os valores, em templateString, do **valor da operação** e a **operação**
* Abaixo definir o (**currentDigit)** como vazio

14 Criar método changeOperation e passar como parâmetro a operação

* Crair uma constante e passar todos os símbolos da operação
* Fazer uma verificação lógica de caso a operação enviada não esteja inclusa na constante acima apenas retornar para cancelar.
* Se estiver tudo certo seguir o código. Fazer com que o texto inserido no (mainDigit) desta classe remova com slice o ultimo digito (que será o símbolo) do texto inserido no (mainDigit) e adicionar com “+” a operação

16 Criar métodos com parâmetros vazios para cada **operação** que não seja numérica

* DEL
  + Atribuir ao texto inserido no curretnDigit desta classe o texto inserido no curretnDigit desta classe com método slice aplicado no ultimo digito
* CE
  + Zerar o valor do texto inserido no curretnDigit desta classe
* C
  + Zerar o valor do texto inserido no curretnDigit desta classe e zerar o valor do texto inserido no mainDigit desta classe
* =
  + Atribuir a uma variável operation o texto inserido no (mainDigit) e aplicar a funcionalidade split para separar a primeira string

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3 declarar uma **nova constante** para chamar a instância da **calculadora**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 Loop para cada botão:

* Adicionar evento de click;
* Selecionar o **valor** do texto de cada botão em uma constante;
* Aplicar lógica para saber se o valor é número ou “.” (também considerar ponto como número).
  + 6 Se sim, o **método digito** da **calculadora** recebe como parâmetro o **valor**
  + 8 Se não o **método das operações** da **calculadora** recebe como parâmetro o **valor**