



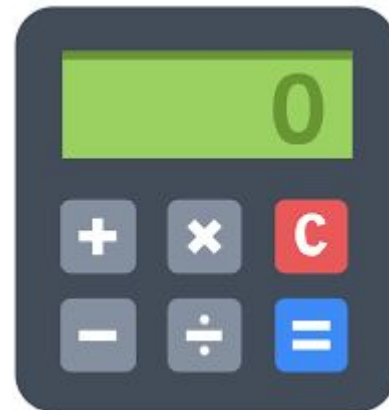
Dart Básico - Aula 1

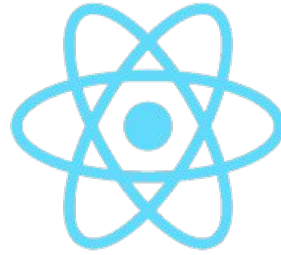


Precisamos desenvolver uma calculadora...

... qual tecnologia usaremos?...

...e em qual plataforma ela será executada?



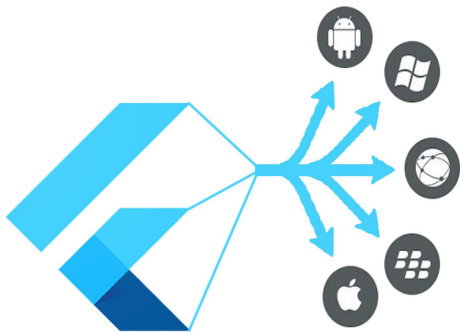


React



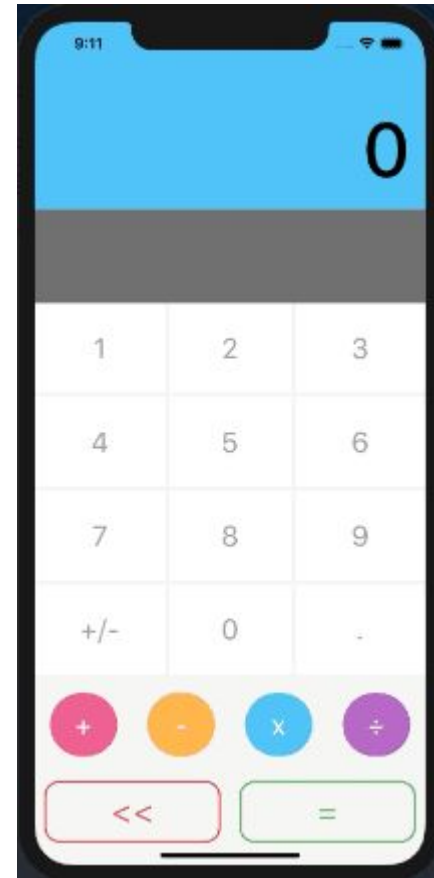


- Lançado em 2017
- Versão atual: 3.7



- Botões
- Textos
- Páginas
- Aplicação

Tudo é um Widget!



Dart

```
class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            const Text(
              'You have pushed the button this many times:',
            ),
            Text(
              '$_counter',
              style: Theme.of(context).textTheme.headlineMedium,
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: const Icon(Icons.add),
      ),
    );
  }
}
```

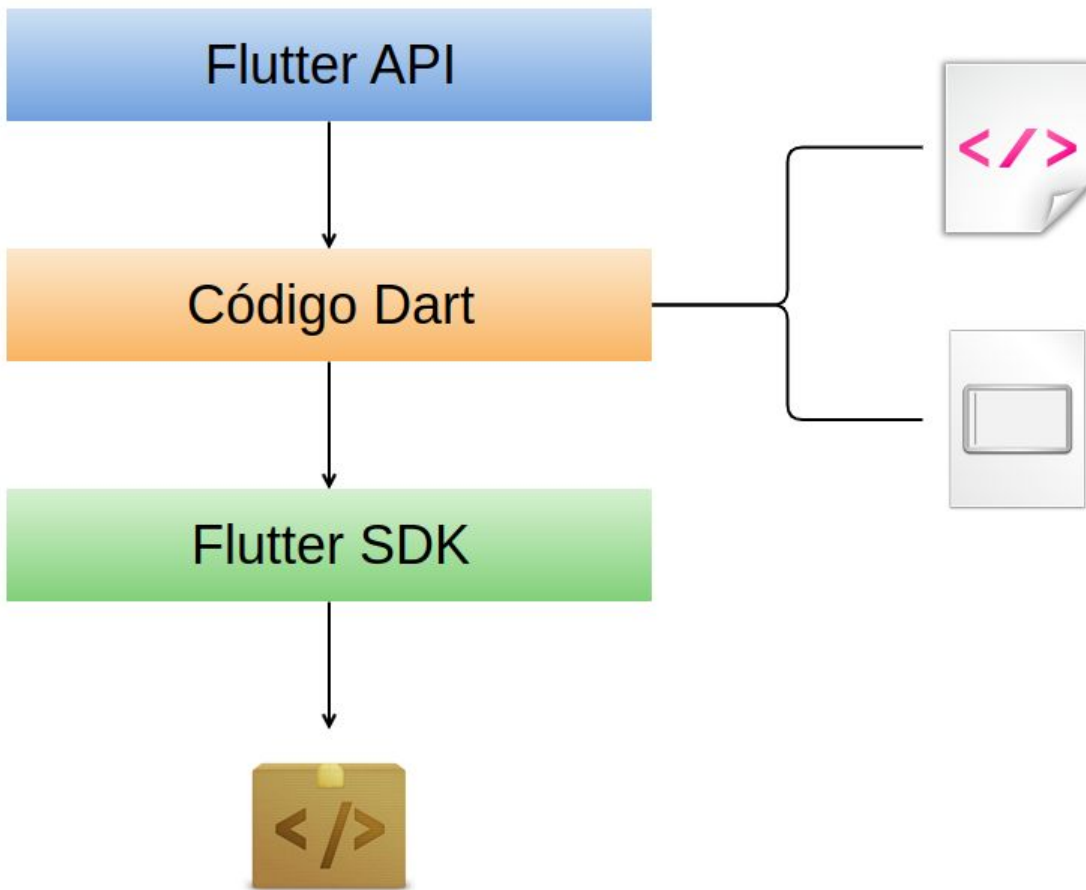
Flutter Demo Home Page

You have pushed the button this many times:

3



Como o Flutter funciona?





- Compilada
- Fortemente tipada
- Lançada em 2011
- Semelhante a C
- *Case sensitive*

```
void main() {  
    // Seu código aqui  
}  
  
void main(List<String> arguments) {  
    // Seu código aqui  
}
```




Object

- Numbers - int & double
- String
- Bool
- List
- Set
- Maps

null

```
// String para inteiro  
int numero = int.parse('1')
```

```
// Inteiro para string  
String numeroStr = 1.toString()
```



- Nome deve começar com letra ou _
- Fortemente tipada
- Tipo nomeVariavel

```
int idade = 21;

var nome = "Paulo";

String sobrenome = 'Silva';

var nomeCompleto = nome + " " + sobrenome;

print("O nome completo do usuário é
$nomeCompleto");
```



- Matemáticos: `+` `-` `*` `/`
- Lógicos: `&&` `||` `!`
- Igualdade: `==` `!=`
- Cascata: `..` `?..`
- Relacional e teste de tipo: `>=` `>` `<=` `<` `as` `is` `is!`
- Operadores condicionais: `condição ? resultado1 : resultado2` `expressao1 ?? expressao2;`



- Comentário de somente uma linha: `//`
- Comentário de várias linhas: `/**/`
- Documentação: `///` `/**`
- Link para documentação: `[]`



if...else

```
if (diaAula()) {  
    voce.arrumarMaterial();  
} else if (diaTrabalho()) {  
    voce.irTrabalhar();  
} else {  
    voce.abrirGelada();  
}
```



Vamos criar um programa que leia a idade de uma pessoa a partir de uma variável. Conforme a idade da pessoa, o programa deve exibir em tela a seguinte informação:

- Se a pessoa tiver até 12 anos: Criança
- Se a pessoa tiver entre 12 a 17 anos: Adolescente
- Se a pessoa tiver 18 anos ou mais: Adulto
- Se a pessoa tiver 60 anos ou mais: Idoso





for

```
// for
for (var i = 0; i < 5; i++) {
    print("Partiu Flutter x$i");
}

// foreach
for (final candidato in listaCandidatos) {
    candidato.entrevistar();
}
```

Exercício 2



Vamos escrever um programa que calcule o número fatorial de um número. Esse número será lido a partir de uma variável.

O fatorial é obtido somando multiplicando todos os valores menores ou iguais ao número que se deseja fatorar. Ex:

$$5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$$





while...do

```
while (!estaGanhando()) {  
    meninoNey.fazerGol();  
}
```

do...while

```
do {  
    escreverLinha();  
} while (!terminouDocumento())
```



Para gerar um número aleatório em dart, utilizo o seguinte código:

```
import 'dart:math';

void main() {
  Random random = Random();

  random.nextInt(100);
}
```

Exercício 3



Neste exercício, vamos gerar um número aleatório, entre 0 a 10. Enquanto esse número não for igual a 7, deveremos tentar novamente.

Exiba sempre o número que você tentou gerar.





- break
- continue

switch

```
string comando = "run";

switch(comando) {
    case "open": executarOpen();
                  break;

    case "run":  executarRun();
                  break;

    default: enviarMensagem();
}
```




Vamos criar uma calculadora. Teremos duas variáveis para guardar os valores a calcular, e uma terceira variável, que guardará o texto da operação a ser efetuada pela calculadora:

- soma
- subtração
- divisão
- multiplicação

Conforme a operação, a calculadora deverá efetuar o cálculo.





```
void teste(int prParam) {  
    return 4;  
}
```

```
bool teste(int prParam) => prParam == 4;
```

```
list.map((item) => item.toUpperCase());
```

Exercício 5



Vamos criar uma função que retorne o valor de Fibonacci. Devemos passar por parâmetro o número de vezes que a sequência de Fibonacci deve ser executada.

A sequência de Fibonacci é calculada somando cada número com o anterior.

Ex:

0,1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, ...

