

[< VOLTAR](#)

# Primeiro projeto completo com Flutter

Nessa aula vamos criar nosso primeiro aplicativo híbrido: Um pequeno assistente pessoal que nos informa se devemos abastecer o carro com álcool ou gasolina, dado o preço de cada um.

## NESTE TÓPICO

- O aplicativo a ser desenvolvido e o Material Design
- Criando o projeto
- Criando o corpo da aplicação
- Lógica da aplicação
- Pequena validação dos campos



## O aplicativo a ser desenvolvido e o Material Design

Desenvolveremos nosso primeiro aplicativo a partir desta aula. O aplicativo consistirá em um pequeno assistente que nos informa se devemos abastecer um veículo "flex" com álcool (etanol) ou gasolina, baseado na seguinte regra:

- Se o álcool for até 70% mais barato do que a gasolina, então deve-se utilizar o álcool.
- Caso contrário, gasolina.

Para encontrarmos essa proporção, basta dividirmos o valor do álcool (que deverá ser informado pelo usuário), pelo valor da gasolina (que também deverá ser informado pelo usuário). Por exemplo:

- Álcool, preço por litro: R\$ 3,05
- Gasolina, preço por litro: R\$ 4,55
- Divisão:  $3,05 / 4,55 = 0,670$
- Resultado: O álcool representa, neste caso, 67% do valor da gasolina
- Conclusão: Usar o álcool, pois  $67 < 70$  (67 é menor do que 70)

E para desenvolvermos nossos aplicativos, precisamos também conhecer o Material Design, que consiste basicamente em uma série de componente visuais e ícones desenvolvidos pelo Google com o intuito de serem simples, intuitivos e, claro, muito bonitos, como são as interfaces dos aplicativos da própria gigante. Como o próprio site do projeto define:

Material Design é um sistema de design, apoiado por código-fonte aberto e que ajuda as equipes a construir experiências digitais de alta qualidade.

(MATERIAL DESIGN, s.d.)

Uma grande vantagem de o Flutter utilizar de forma intrínseca o Material Design é que todos os componentes já possuem alguns efeitos visuais, sombreadamentos, arredondamentos, efeitos de flutuação etc., já prontos e, portanto, não precisamos escrever códigos gigantescos para implementar algo que já "já vem pronto" para usarmos, ou seja, não precisamos ficar "reinventando a roda" e isso é sensacional.

O Flutter implementa todos os seus componentes utilizando o Material Design, o que deixará nossas aplicações ainda mais elegantes, mas não se preocupe, pois, cada um dos componentes é altamente personalizável.

## SAIBA MAIS SOBRE O MATERIAL DESIGN

Para saber mais sobre o Material Design, não deixe de consultar seu site oficial ([material.io](https://material.io)) e explorar seus componentes e sua documentação. Os componentes estão, atualmente, disponíveis para desenvolvimento em Android nativo, iOS nativo, Web e Flutter.



Mas vamos ao que interessa: Começar nosso primeiro projeto! O vídeo abaixo mostra como nosso App ficará depois de pronto e um pouco mais sobre o Material Design.

O aplicativo que vamos desenvolver e o MaterialDesign



## Criando o projeto

Agora que já sabemos como nosso App se comportará, vamos começar a codificação dele e, para isso, precisamos criar o um novo projeto.

Se você for utilizar a IntelliJ ou o Android Studio, a criação do projeto é bastante simples e pode ser feita normalmente pelos próprios editores e, inclusive, como mostrado no vídeo abaixo.

Entretanto, se você for utilizar o Visual Studio Code, você deverá criar o projeto manualmente, ou seja, via linha de comando no terminal, como já aprendemos, de qualquer maneira, se você for criar o projeto manualmente, siga os passos abaixo:

1. Crie uma pasta para ser seu "container" de projetos
2. Abra o terminal (ou prompt de comando)
3. Navegue usando o comando "cd" até a pasta do ítem (1)
4. Execute o comando abaixo:  
`flutter create --org br.uninove.primeiroapp alcool_gasolina`
5. Pronto, agora basta abrir a pasta criada com o Visual Studio Code

Lembre-se que os passos descritos acima são exclusivos para os discentes que utilizam o Visual Studio Code, ok? Caso você utilize as demais IDEs, pode seguir adiante.

O vídeo abaixo mostra como criar o projeto, arrumar algumas coisas e nossas primeiras linhas de código com o Flutter.



### Criando o projeto



Fique bastante atento(a) com os detalhes da codificação, especialmente pela abertura e fechamento dos parênteses, chaves etc. É muito fácil, no começo, errar isso e se perder, mas não desanime, pois todos nós passamos por essa fase.

## Criando o corpo da aplicação

Agora que já configuramos o projeto, aprendemos sobre widgets do tipo stateless e stateful e criamos a barra superior de nosso App, podemos começar a desenvolver o corpo da aplicação.

Para isso, acompanhe a continuação do vídeo anterior no vídeo abaixo.



### Criação do corpo da aplicação



## Lógica da aplicação

Agora que já terminamos de implementar todos os componentes visuais do aplicativo, podemos começar a implementar a parte lógica, ou seja, incluir as funcionalidades nos dois botões: O de calcular e o de limpar os dados.

Para isso, nada melhor do que um pequeno vídeo mostrando como implementar estes recursos, usando as melhores práticas.

### Criação da lógica da aplicação



Como você deve ter visto no vídeo anterior, o desenvolvimento em Flutter é realmente muito rápido e esta é sua principal característica: A agilidade em que podemos criar aplicações robustas.

## Pequena validação dos campos

Agora que nosso aplicativo já funciona, podemos aprender como funciona o processo de validação de um formulário em Flutter e, neste caso, vamos se os campos foram devidamente preenchidos e, caso contrário, inclui uma informação em destaque para o usuário realizar a correção daquele campo.

E, para que isso funcione corretamente, nada melhor do que um pequeno vídeo mostrando todo este processo, conforme disponível abaixo:

Finalizando o projeto: Inclusão da validação



## Conclusão e projeto

Chegamos ao final de nosso projeto de uma aplicação híbrida com Flutter e, como você pode observar em poucos passos desenvolvemos uma aplicação completa e profissional que pode ser executada em smartphones com Android e iOS.

Não se preocupe e não desanime se você teve qualquer dificuldade durante o processo. Essa é uma linguagem que pode ser nova para você e a forma de desenvolvimento hierárquico pode ser um pouco estranha mesmo no começo.

Por conta disso, se você precisar verificar qualquer coisa no projeto que eventualmente tenha passada despercebida, você pode fazer ver a codificação completa (pois este projeto possui um único arquivo, por enquanto) abaixo ou realizar o download do projeto, logo em seguida.





```

1. import 'package:flutter/material.dart';
2.
3. void main() {
4.   runApp(MaterialApp(
5.     home: Home(),
6.     debugShowCheckedModeBanner: false,
7.   ));
8. }
9.
10. class Home extends StatefulWidget {
11.   @override
12.   _HomeState createState() => _HomeState();
13. }
14.
15. class _HomeState extends State<Home> {
16.   TextEditingController alcoolController = TextEditingController();
17.   TextEditingController gasolinaController = TextEditingController();
18.   String _resultado = "Informe os valores";
19.   GlobalKey<FormState> _formKey = GlobalKey<FormState>();
20.
21.   void _reset() {
22.     alcoolController.text = "";
23.     gasolinaController.text = "";
24.     setState(() {
25.       _resultado = "Informe os valores";
26.       _formKey = GlobalKey<FormState>();
27.     });
28.   }
29.
30.   void _calculaCombustivelIdeal() {
31.     setState(() {
32.       double vAlcool = double.parse(alcoolController.text.replaceAll(',', '.'));
33.       double vGasolina =
34.         double.parse(gasolinaController.text.replaceAll(',', '.'));
35.       double proporcao = vAlcool / vGasolina;
36.
37.       _resultado =
38.         (proporcao < 0.7) ? "Abasteça com Álcool" : "Abasteça com Gasolina!";
39.     });
40.
41.     // if(proporcao < 0.7) {
42.     //   _resultado = "Abasteça com Álcool";
43.     // } else {
44.     //   _resultado = "Abasteça com Gasolina";
45.     // }
46.   }
47.
48.   @override
49.   Widget build(BuildContext context) {
50.     return Scaffold(
51.       appBar: AppBar(
52.         title: Text(
53.           "Álcool ou Gasolina?",
54.           style: TextStyle(color: Colors.white),
55.         ),
56.         centerTitle: true,
57.         backgroundColor: Colors.lightBlue[900],
58.         actions: <Widget>[
59.           IconButton(
60.             icon: Icon(Icons.refresh),
61.             onPressed: () {
62.               _reset();
63.             })
64.         ],
65.       ),
66.       body: SingleChildScrollView(
67.         padding: EdgeInsets.fromLTRB(10.0, 0, 10.0, 0),
68.         child: Form(
69.           key: _formKey,
70.           child: Column(
71.             crossAxisAlignment: CrossAxisAlignment.stretch,

```





```

72.         children: <Widget>[
73.             Icon(
74.               Icons.local_gas_station,
75.               size: 150.0,
76.               color: Colors.lightBlue[900],
77.             ),
78.             TextFormField(
79.               controller: alcoolController,
80.               textAlign: TextAlign.center,
81.               keyboardType: TextInputType.number,
82.               validator: (value) =>
83.                 value.isEmpty ? "Informe o valor do Álcool" : null,
84.               decoration: InputDecoration(
85.                 labelText: "Valor do Álcool",
86.                 labelStyle: TextStyle(color: Colors.lightBlue[900]),
87.               ),
88.               style:
89.                 TextStyle(color: Colors.lightBlue[900], fontSize: 26.0),
90.             ),
91.             TextFormField(
92.               controller: gasolinaController,
93.               textAlign: TextAlign.center,
94.               keyboardType: TextInputType.number,
95.               validator: (value) =>
96.                 value.isEmpty ? "Informe o valor da gasolina!" : null,
97.               decoration: InputDecoration(
98.                 labelText: "Valor da Gasolina",
99.                 labelStyle: TextStyle(color: Colors.lightBlue[900])),
100.              style:
101.                TextStyle(color: Colors.lightBlue[900], fontSize: 26.0),
102.            ),
103.            Padding(
104.              padding: EdgeInsets.only(top: 20.0, bottom: 20.0),
105.              child: Container(
106.                height: 50.0,
107.                child: RaisedButton(
108.                  onPressed: () {
109.                    if (_formKey.currentState.validate())
110.                      _calculaCombustivelIdeal();
111.                  },
112.                  child: Text(
113.                    "Verificar",
114.                    style: TextStyle(color: Colors.white, fontSize: 25.0),
115.                  ),
116.                  color: Colors.lightBlue[900],
117.                ),
118.              ),
119.            ),
120.            Text(
121.              _resultado,
122.              textAlign: TextAlign.center,
123.              style:
124.                TextStyle(color: Colors.lightBlue[900], fontSize: 26.0),
125.            )
126.          ],
127.        ),
128.      ),
129.    ));
130.  }
131. }

```



**MATERIAL COMPLEMENTAR**  
 (Https://Img.Uninove.Br/Static/

0/0/0/0/0/0/9/2/5/1/3/9251383/  
Alcool-Gasolina.7z)

---

## Quiz

Exercício Final

Primeiro projeto completo com Flutter

INICIAR ➤

## Referências



ALESSANDRIA, Simone. **Flutter projects**: a practical, Project-based guide to building real-worlds cross-platform mobile applications and games. Birmingham, Reino Unido: Packt Publishing, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/flutter-projects/9781838647773/>. Acesso em: 09 dez. 2020.

DART. **Dart documentation**. [s.d.]. *Site*. Disponível em: <https://dart.dev/>. Acesso em: 08 dez. 2020.

FLUTTER. **Flutter docs**. [s.d.]. *Site*. Disponível em: <https://flutter.dev>. Acesso em: 09 dez. 2020.

MATERIAL DESIGN. **Material Design documentation**. [s.d.]. *Site*. Disponível em: <https://material.io/>. Acesso em: 08 dez. 2020.

SINHA, Sanjib . **Quick start guide to Dart programming**: create high performance applications for the web and mobile. Lompoc, CA, EUA: Apress, 2019. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/quick-start-guide/9781484255629/>. Acesso em: 09 dez. 2020.

WINDMILL, Eric. **Flutter in action**. Nova Iorque: Manning publications, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/flutter-in-action/9781617296147/>. Acesso em: 08 dez. 2020.

ZACCAGNINO, Carmine. **Programming Flutter**. [s.l.]: The Pragmatic Bookshelf, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/programming-flutter/9781680507621/>. Acesso em: 09 dez. 2020.



Avalie este tópico



ANTERIOR

Introdução ao Flutter

Biblioteca

([https://www.uninove.br/conheca-](https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Projeto de um app consumindo JSON de Cursos

© Todos os direitos reservados

Ajuda?  
PRÓXIMO  
(<https://ava.uninove.br/cursos/>)

API

