

## *Estrutura básica de um projeto*

---

Um projeto Flutter possui uma arquitetura de pastas bastante extensa, e seu conhecimento é necessário para qualquer programador que esteja criando uma aplicação em Flutter. Sendo assim, antes de iniciarmos nossos estudos sobre esse maravilhoso SDK, vamos aprender sobre sua estrutura de diretórios e arquivos.

### *Estrutura de pastas e arquivos do Flutter*

Um projeto base do Flutter possui 5 diretórios e 6 arquivos. Sendo que cada um deles possui uma importância para o projeto e para a aplicação como um todo. Vejamos agora quais são eles e suas respectivas finalidades em um projeto Flutter.

- **Diretórios**
  - **android:** Esse diretório, como o próprio nome já sugere, armazena os arquivos ao Android. Ou seja, o código-nativo Java ou Kotlin, os ícones da aplicação, arquivos de configuração, entre outros.
  - **ios:** Esse diretório, tem a mesma finalidade do diretório android, porém para para sistemas iOS.
  - **lib:** Esse diretório é único diretório (Exceto o test) em que nós devemos fazer alterações. É neste diretório que colocaremos todo o nosso código-fonte e recursos. Por mais que possa parecer estranho, o diretório lib é onde devemos desenvolver nossa aplicação Flutter.
  - **res:** Esse diretório contém alguns recursos importantes para a aplicação, como por exemplo strings de internacionalização do aplicativo.
  - **test:** Como o próprio nome já sugere, é aqui que colocamos nossos testes unitários.
- **Arquivos**
  - **.gitignore:** Se você conhece um pouco de controle de versões com Git, já deve estar familiarizado com esse arquivo. Mas caso nunca tenha ouvido falar dele, esse arquivo é responsável por descrever quais arquivos não devem ser monitorados pelo Git.
  - **.metadata:** Dados que o Android Studio usa para rastrear o projeto.

- **.packages:** O Flutter vem com seu próprio gerenciador de pacotes do flutter, o Pub. Esse arquivo mantém o rastreamento das dependências de pacotes do nosso projeto.
- **nome\_do\_projeto.iml:** Esse arquivo recebe o nome do seu projeto, e é responsável por manter as configurações de projeto do Android Studio.
- **pubspec.yaml:** Esse arquivo mantém a descrição de todos os pacotes que são utilizados em nosso projeto Flutter. Ele corresponde ao package.json do NPM ou o composer.json do PHP.
- **README.md:** Esse arquivo é um arquivo markdown e pode ser utilizado da forma que desejarmos. Normalmente é utilizado para guardarmos informações ou instruções sobre o projeto. Ele só é visível enquanto em desenvolvimento, não pode ser utilizado para passar instruções para o usuário do aplicativo.

### ***Manifesto pubspec.yaml***

Como comentamos no tópico anterior, o Flutter utiliza um gerenciador de pacotes chamado **Pub**, e seu arquivo de dependências é escrito em uma linguagem chamada yaml.

Esse formato é muito parecido com o formato JSON, se diferenciando pelo fato de utilizar espaços em branco para representar a hierarquia de objetos.

Na figura abaixo podemos ver um exemplo do arquivo pubspec.yaml padrão de uma aplicação flutter.

```
1 name: projeto_teste
2 description: Um simples exemplo do pubspec.yaml.
3 publish_to: "none"
4 version: 1.0.0+1
5
6 environment:
7   sdk: "≥2.7.0 <3.0.0"
8
9 dependencies:
10  flutter:
11    sdk: flutter
12
13  cupertino_icons: ^1.0.0
14
15 dev_dependencies:
16  flutter_test:
17    sdk: flutter
18
19 flutter:
20  uses-material-design: true
21
22  assets:
23    - images/a_dot_burr.jpeg
24    - images/a_dot_ham.jpeg
25
26  fonts:
27    - family: Schyler
28      fonts:
29        - asset: fonts/Schyler-Regular.ttf
30        - asset: fonts/Schyler-Italic.ttf
31          style: italic
32    - family: Trajan Pro
33      fonts:
34        - asset: fonts/TrajanPro.ttf
35        - asset: fonts/TrajanPro_Bold.ttf
36          weight: 700
37
```

Neste momento não será necessário que entendamos todos os aspectos desse arquivo. Basta entendermos que será nele que colocaremos as dependências de nosso projeto Flutter. Mas para frente, voltaremos a falar deste arquivo.

## ***Widgets***

---

O primeiro conceito que devemos entender em Flutter são os Widgets. Widget é o nome dado a um componente de interface no Flutter. Para ser sincero, quase tudo no Flutter é um Widget. Um botão, um texto, um menu e tudo, tudo mesmo, é um Widget.

Para resumir de forma simples e sucinta, um Widget é:

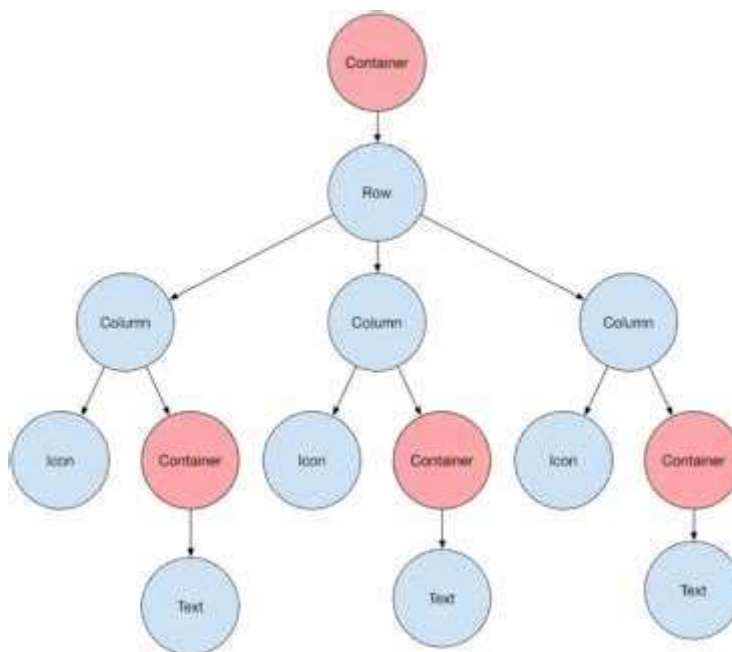
“Um bloco de UI, ou seja, um elemento gráfico apresentado na tela do usuário”.

Outro aspecto que merece atenção quando falamos de Widgets, é o fato de que todo Widget é um objeto, ou seja, todo Widget do Flutter é uma instância de uma classe do core do Flutter. E como qualquer objeto, esses widgets possuem propriedades, métodos e construtores.

## ***A árvore de Widgets***

---

O Flutter faz uso de uma hierarquia de elementos visuais, os nossos Widgets, de forma a que eles sejam empilhados um dentro do outro formando uma espécie de árvore de Widgets. Árvore essa que forma a tela visualizada pelo usuário. Na figura 26 podemos ver um exemplo desta árvore de widgets do Flutter.



**Figura 1 – Árvore de Widgets do Flutter**

**Fonte: Flutter.io**