

Preparando o ambiente

Nesse curso, utilizaremos o projeto final do curso de [Flutter: Gerenciamento de estados com Provider](https://cursos.alura.com.br/course/flutter-gerenciamento-de-estados-com-provider) (<https://cursos.alura.com.br/course/flutter-gerenciamento-de-estados-provider>) como base. Nos links a seguir, você pode:

- [acessar o projeto que utilizaremos como base no GitHub](https://github.com/alura-cursos/flutter-gerenciamento-de-estado/tree/aula-5) (<https://github.com/alura-cursos/flutter-gerenciamento-de-estado/tree/aula-5>).
- ou [baixar o zip do arquivo](https://github.com/alura-cursos/alura_flutterforms/archive/master.zip) (https://github.com/alura-cursos/alura_flutterforms/archive/master.zip).

Também será muito útil você [baixar agora a logomarca do Bytebank](https://github.com/alura-cursos/alura_flutterforms/blob/aula_1/assets/images/bytebank_logo.png) (https://github.com/alura-cursos/alura_flutterforms/blob/aula_1/assets/images/bytebank_logo.png) que usaremos neste curso.

Para começarmos, é **importantíssimo** que você tenha o Flutter e as demais ferramentas necessárias para realizar a codificação com tudo instalado na sua máquina. O [Flutter está disponível para download em sua página oficial](https://flutter.dev/docs/get-started/install) (<https://flutter.dev/docs/get-started/install>). Escolha o sistema operacional que está utilizando e siga as instruções de instalação.

No momento em que criamos este treinamento, o Flutter está disponível para Windows, macOS, Linux e Chrome OS.

Durante o curso, utilizamos a versão 1.22. Se no momento em que estiver fazendo o curso houver uma versão mais recente, fique à vontade em utilizá-

la, mas para melhor acompanhamento deste projeto, sugerimos que utilize a mesma versão.

Após finalizar a configuração que adiciona o Flutter na variável de ambiente `PATH`, que permite usar o Flutter em qualquer diretório via prompt ou terminal, execute o comando `flutter doctor` no terminal ou prompt. Você deverá encontrar um resultado similar a este:

```
[-] Android toolchain - develop for Android devices
  • Android SDK at /Users/skywalker/Library/Android/sdk
  X Android SDK is missing command line tools; download from h
  • Try re-installing or updating your Android SDK,
    visit https://flutter.dev/setup/#android-setup for detaile
```

[COPIAR CÓDIGO](#)

Essa resposta é uma abordagem comum no comando `doctor`, que avalia se o ambiente está corretamente configurado. Caso tenha alguma divergência, é apresentado o que falta. Sempre que houver algo faltando para o bom funcionamento do Flutter, o comando exibirá o que falta e, em alguns casos, irá sugerir o que fazer para resolver a questão.

A mensagem final do `flutter doctor` varia de acordo com o que você tem instalado no seu computador.

Instalando o Android Studio e Android SDK

Após instalar o Flutter, [baixe o Android Studio](#) (<https://developer.android.com/studio>) e [instale para acessar o Android SDK](#)

(<https://developer.android.com/studio/install>) e criar um emulador Android que será utilizado durante o curso.

Criando o emulador

Ao finalizar o processo de instalação, vai aparecer a *launcher* do Android Studio. Clique na opção **Configure**, na parte inferior à direita, e escolha a opção **AVD Manager**. Siga as [instruções para criar um emulador](https://developer.android.com/studio/run/managing-avds#createavd) (<https://developer.android.com/studio/run/managing-avds#createavd>). Na documentação do Flutter, você também encontra um [tutorial de instalação do emulador](https://developer.android.com/studio/run/managing-avds#createavd) (<https://developer.android.com/studio/run/managing-avds#createavd>).

No curso, utilize a versão 4.1 do Android Studio e o emulador da API 30 Android 11. Se no momento em que estiver fazendo o curso houver uma mais recente, fique à vontade em utilizá-la.

Escolhendo o editor/IDE para desenvolvimento

Por fim, precisamos configurar a ferramenta que vai nos auxiliar na escrita de código. No momento que o curso foi gravado, a documentação do Flutter apresentou como sugestões as seguintes ferramentas:

- [Android Studio](https://developer.android.com/studio/) (<https://developer.android.com/studio/>);
- [VSCode](https://code.visualstudio.com/) (<https://code.visualstudio.com/>);
- [IntelliJ](https://www.jetbrains.com/idea/download/) (<https://www.jetbrains.com/idea/download/>).

Você pode escolher a ferramenta da sua preferência. As configurações necessárias são descritas na [página oficial do Flutter que sugere a instalação de plugins tanto para o Flutter como para o Dart](https://flutter.dev/docs/get-started/editor) (<https://flutter.dev/docs/get-started/editor>).

Durante o curso utilizaremos o IntelliJ. Sendo assim, você pode escolher entre o Android Studio ou o IntelliJ para manter um ambiente similar ao que será apresentado no curso. Optamos pelo IntelliJ por consumir menos recursos de hardware e ter uma interface mais amigável para desenvolvemos em Flutter.

O Android Studio é um IntelliJ Community personalizado com tudo o que você precisa para desenvolver com o Flutter, segundo a documentação.

Portanto, você pode baixar apenas o Android Studio e utilizá-lo como IDE também!

Após realizar todas as configurações, você pode verificar se o ambiente configurado está correto com `flutter doctor` e, se tudo der certo, podemos prosseguir com a próxima atividade.

Caso tenha algum problema, entre em contato com a gente por meio do fórum! Lembrando que nele você também pode ajudar e ser ajudado(a) por colegas de curso, incentivando o espírito de comunidade.

Bom aprendizado e bora aprender :)