

PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS

Prof. Wilson Lourenço



INTRODUÇÃO AO FLUTTER

Flutter

- Flutter é um framework para desenvolvimento mobile multiplataforma (iOS e Android) da Google que foi lançado em 2018 que usa o DART como principal linguagem de desenvolvimento.



<https://flutter.dev/>

- Com o rápido aumento de frameworks mobile, como Ionic, React Native e Xamarin, o pessoal do Google decidiu entrar no jogo e desenvolver seu próprio framework com suporte para Android e iOS usando a mesma base de código.
- Ele utiliza uma abordagem única para lidar com os componentes nativos de cada plataforma

- O Flutter é um SDK de desenvolvimento de aplicativos móveis de código aberto desenvolvido e patrocinado pelo Google, usado para o desenvolvimento de aplicativos para Android e iOS - além de ser o principal método de criação de aplicativos para o sistema operacional Google Fuchsia.



- Cada componente é implementado pelo próprio framework e apresentado ao usuário por um motor de renderização próprio.
- O Flutter, de maneira similar à engines utilizadas em jogos como a OpenGL, renderiza os widgets do aplicativo diretamente em um Canvas.

- Assim, é possível executar o mesmo aplicativo tanto em Android quanto em iOS.
- Além disso, por conta dessa solução, também é capaz de executar animações em até 120 fps.

- Flutter é escrito em C, C ++ e Dart. Ele oferece um rico conjunto de widgets totalmente personalizáveis para a criação de interfaces nativas, incluindo a bela biblioteca Material Design e os widgets Cupertino (iOS) e hot reload - o que ajuda a criar UIs rapidamente sem perder o estado em emuladores, simuladores e qualquer hardware para iOS e Android.

- Todos esses ótimos recursos ajudaram o Flutter a decolar muito rapidamente, e os desenvolvedores estão migrando para o framework. É também um dos projetos mais utilizados no GitHub, o que o ajudou a ganhar ainda mais popularidade.

Visão Geral do Projeto

- O aplicativo que criaremos ao longo dessa disciplina é um aplicativo que podemos usar para rastrear documentos pessoais importantes com data de validade, como passaportes, carteiras de motorista ou cartões de crédito.



- Esse tipo de aplicativo é útil, portanto, sabemos quando precisamos renovar esses documentos importantes antes que eles expirem.
- Vamos usar a linguagem de programação Dart com o Flutter, com um banco de dados SQLite local.
- Vamos configurar nosso ambiente de desenvolvimento.

Instalação

- A configuração do Flutter é incrivelmente fácil, com todas as etapas de instalação estão bem documentadas no site oficial do Flutter.
- Como utilizo o Windows 10, descreverei as etapas de instalação e as informações relacionadas a este sistema operacional.




- No Windows, existem alguns requisitos de sistema essenciais que precisam estar em vigor, incluindo o PowerShell 5.0 (ou posterior) e o Git for Windows 2.X (ou posterior) instalado.
- Embora você possa escrever aplicativos Flutter em qualquer editor de sua escolha, eu recomendo o Visual Studio Code ou o Android Studio.

- O Flutter depende de uma instalação completa do Android Studio para fornecer suas dependências da plataforma Android.
- Você também precisará configurar um emulador de dispositivo Android (Eu recomendo o GenyMotion).
- Essas etapas são descritas na documentação oficial.

- Com os pré-requisitos estabelecidos para o Windows, tudo o que precisamos fazer é baixar o pacote de instalação do SDK do Flutter.
- Depois de baixar o arquivo zip, extraia-o para uma pasta desejada na sua unidade, como C:\src\Flutter.
 - Não extraia os arquivos Flutter em C:\Arquivos de Programas ou C:\Arquivos de Programas(x86), que exigem permissões elevadas ou de administrador.

- Quando os arquivos estiverem na pasta desejada, localize o arquivo `Flutter_console.bat`



Name	Date modified	Type	Size
.pub-cache	05-Sep-18 4:53 AM	File folder	
bin	08-Dec-18 9:29 PM	File folder	
dev	08-Dec-18 9:29 PM	File folder	
examples	08-Dec-18 9:29 PM	File folder	
packages	22-Sep-18 8:45 PM	File folder	
.cirrus.yml	08-Dec-18 9:29 PM	Yaml Source File	8 KB
.gitattributes	05-Sep-18 4:53 AM	Git Attributes Sour...	1 KB
.gitignore	08-Dec-18 9:29 PM	Git Ignore Source ...	2 KB
analysis_options.yaml	08-Dec-18 9:29 PM	Yaml Source File	8 KB
AUTHORS	08-Dec-18 9:29 PM	File	2 KB
CODE_OF_CONDUCT.md	08-Dec-18 9:29 PM	Markdown Source ...	3 KB
CONTRIBUTING.md	08-Dec-18 9:29 PM	Markdown Source ...	4 KB
dartdoc_options.yaml	08-Dec-18 9:29 PM	Yaml Source File	1 KB
flutter_console.bat	08-Dec-18 9:29 PM	Windows Batch File	2 KB
flutter_root.iml	05-Sep-18 4:53 AM	IML File	1 KB
LICENSE	05-Sep-18 4:53 AM	File	2 KB
PATENTS	05-Sep-18 4:53 AM	File	2 KB
README.md	08-Dec-18 9:29 PM	Markdown Source ...	7 KB
version	24-Jan-19 2:40 PM	File	1 KB

- Adicione a pasta src\flutter\Bin na variável path das variáveis de ambiente do Windows e execute o arquivo Flutter_console.bat

```
Flutter Console

##### ##      ##      ##### ##### #####
##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##
##### ##      ##      ##      ##      #####
##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##
##      ##### #####      ##      ##### ##

WELCOME to the Flutter Console.
=====

Use the console below this message to interact with the "flutter" command.
Run "flutter doctor" to check if your system is ready to run Flutter apps.
Run "flutter create <app_name>" to create a new Flutter project.

Run "flutter help" to see all available commands.

Want to use an IDE to interact with Flutter? https://flutter.io/ide-setup/

Want to run the "flutter" command from any Command Prompt or PowerShell window?
Add Flutter to your PATH: https://flutter.io/setup-windows/#update-your-path
```

- No prompt, digite o comando abaixo para verificar se o Flutter está totalmente operacional e pressione enter.

flutter doctor

- Depois de executar este comando, você obterá um resultado com todos os problemas encontrados - no meu caso, porque eu havia instalado o Android Studio e o Visual Studio Code anteriormente, recebo as seguintes informações.

- Ao executar o instalador do Android Studio, siga a documentação oficial para concluir uma instalação bem-sucedida do Android Studio e SDK.

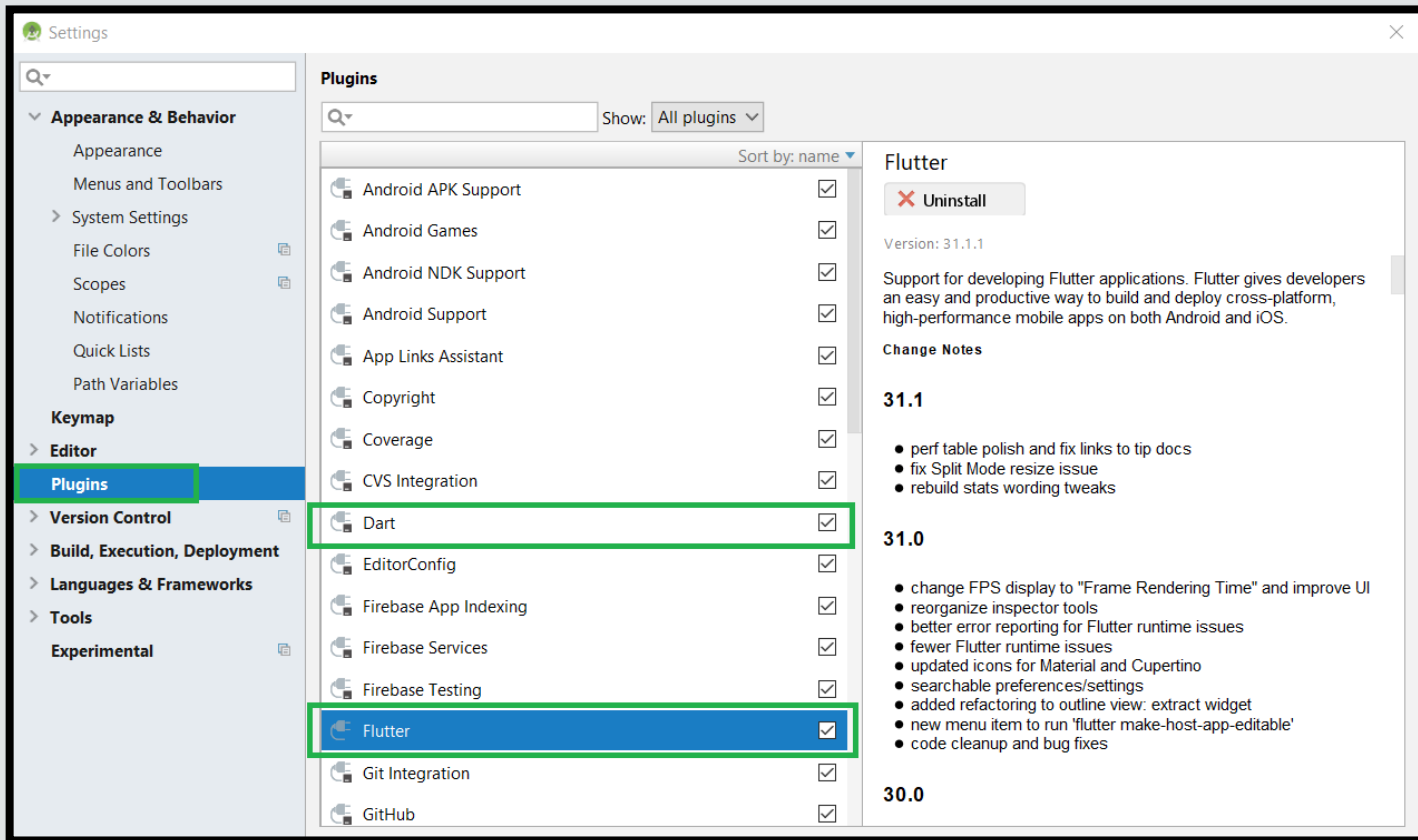
```
Doctor summary (to see all details, run flutter doctor -v):  
[✓] Flutter (Channel beta, v1.0.0, on Microsoft Windows [Version 10.0.17763.253], locale en-US)  
[!] Android toolchain - develop for Android devices (Android SDK 28.0.2)  
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses  
[✓] Android Studio (version 3.2)  
[✓] VS Code, 64-bit edition (version 1.30.2)  
[!] Connected device  
    ! No devices available  
  
! Doctor found issues in 2 categories.
```

- No meu caso, o Flutter está me dizendo que preciso executar o comando **flutter doctor --android-licenses** para resolver um problema com algumas licenças do Android que não foram aceitas.
- Ele também destacou que não tenho um dispositivo físico conectado, o que é bom por enquanto.
- Certifique-se de resolver todos os conflitos destacados pelo comando flutter doctor antes de continuar.

Configurando um Editor

- Depois de concluir todas as etapas da instalação, é necessário configurar o Flutter para trabalhar com o editor de sua escolha. Neste material, usei o Android Studio para codificar esse projeto.
- A documentação oficial do Flutter descreve como configurar o Android Studio (IntelliJ) e o Código do Visual Studio para funcionar com o Flutter.

- Se você usar o Android Studio, depois de seguir as etapas descritas, verá os plugins Dart e Flutter instalados.



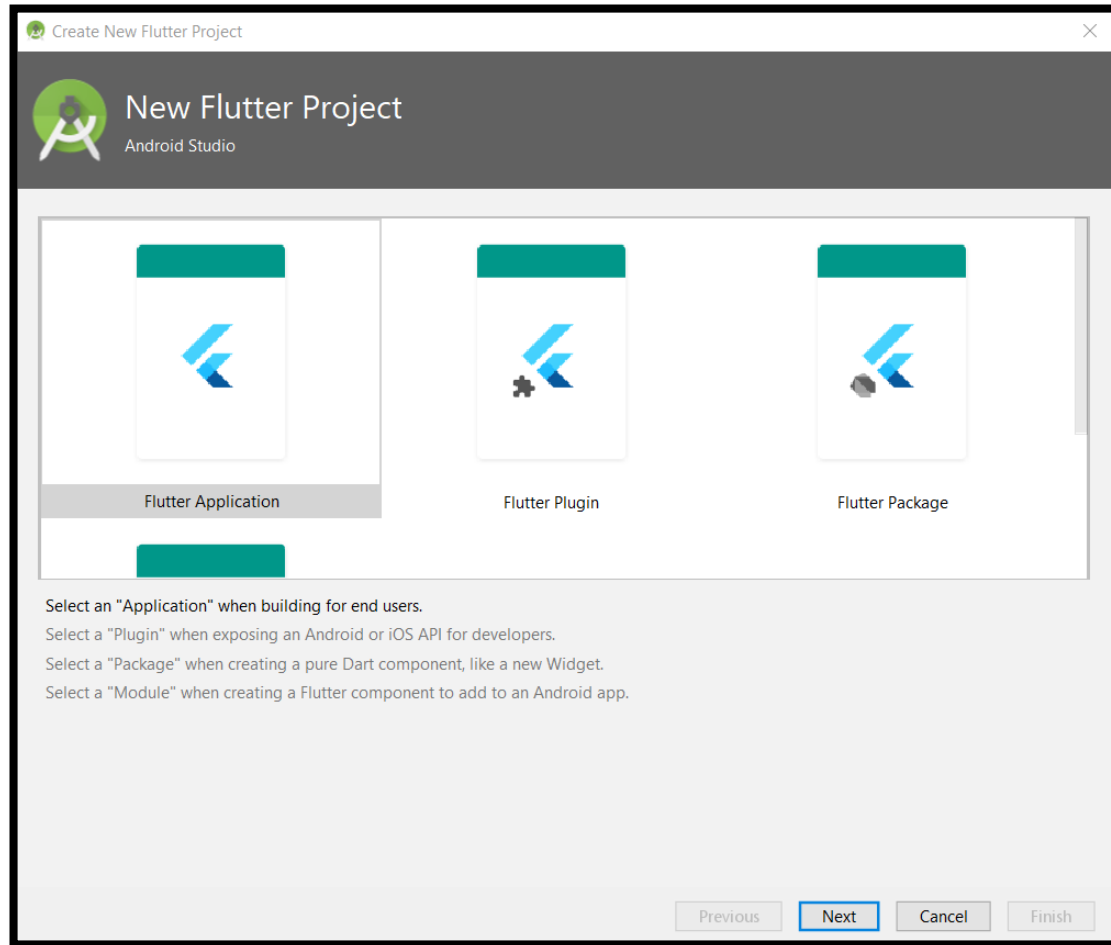
Criando o Aplicativo



Depois que seu editor de escolha for configurado corretamente, seguindo as diretrizes da documentação oficial e minhas sugestões anteriores, é hora de realizar um teste rápido.

Vamos criar um aplicativo de demonstração a partir de um dos modelos predefinidos e experimentaremos o mecanismo de "hot reload" depois de fazer uma alteração no aplicativo.

- Abra o Android Studio e navegue para Arquivo> Novo> Novo Projeto Flutter. A tela seguinte será mostrada.



- Escolha a opção Flutter Application e clique em Next.
- Em seguida, será apresentada uma tela na qual podemos inserir o nome do projeto, o caminho do SDK do Flutter, a localização do projeto e uma descrição para o aplicativo.



New Flutter Application

Android Studio

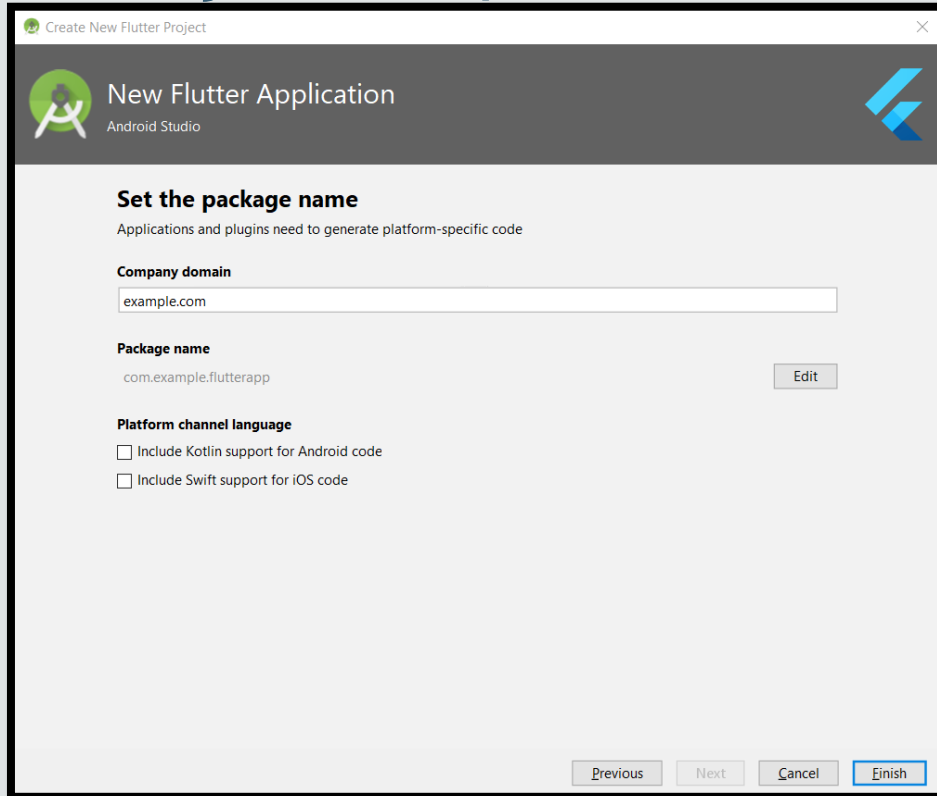


Configure the new Flutter application

Project name**Flutter SDK path**[Install SDK...](#)**Project location****Description**☐ Create project offline[Previous](#)[Next](#)[Cancel](#)[Finish](#)

- Verifique se o campo de texto do caminho do Flutter SDK especifica o local correto da pasta do SDK, conforme explicado anteriormente.
- Com essas opções inseridas, clique em Avançar.
- Na etapa final do processo de criação do aplicativo, somos solicitados a entrar no domínio da empresa e incluir (se aplicável) o suporte Kotlin para código Android e Swift para código iOS - no nosso caso, não é necessário incluir essas opções.

- A figura a seguir mostra a etapa final da tela de criação do aplicativo.



Create New Flutter Project

New Flutter Application
Android Studio

Set the package name
Applications and plugins need to generate platform-specific code

Company domain
example.com

Package name
com.example.flutterapp Edit

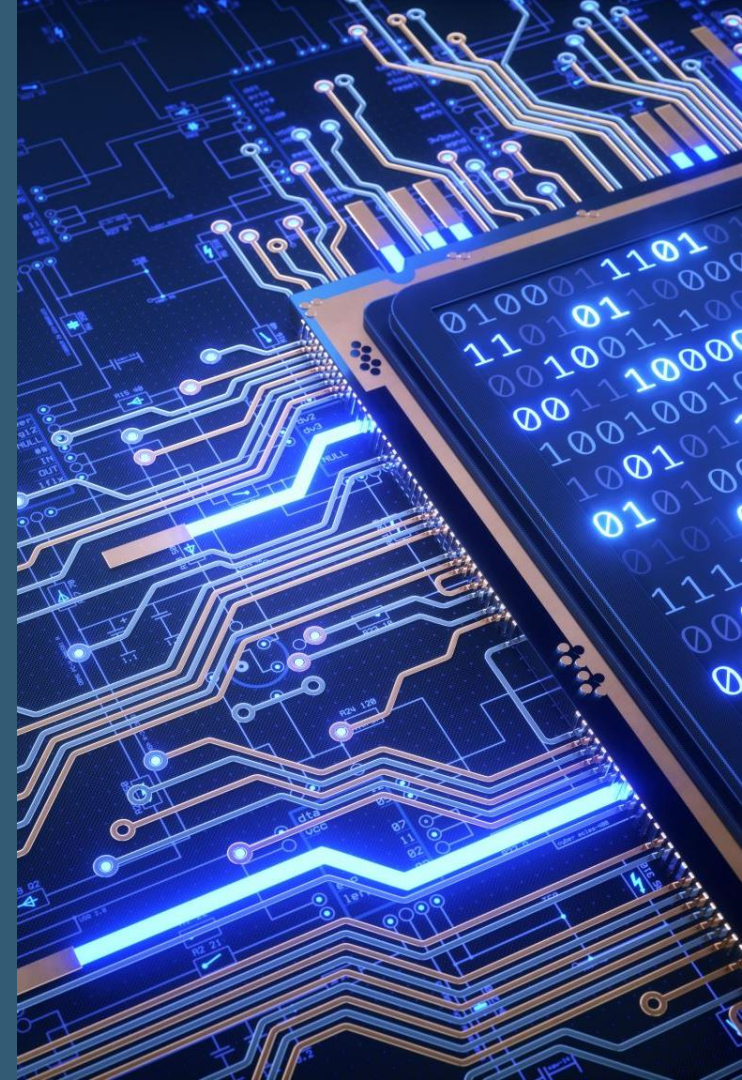
Platform channel language
☐ Include Kotlin support for Android code
☐ Include Swift support for iOS code

Previous Next Cancel Finish

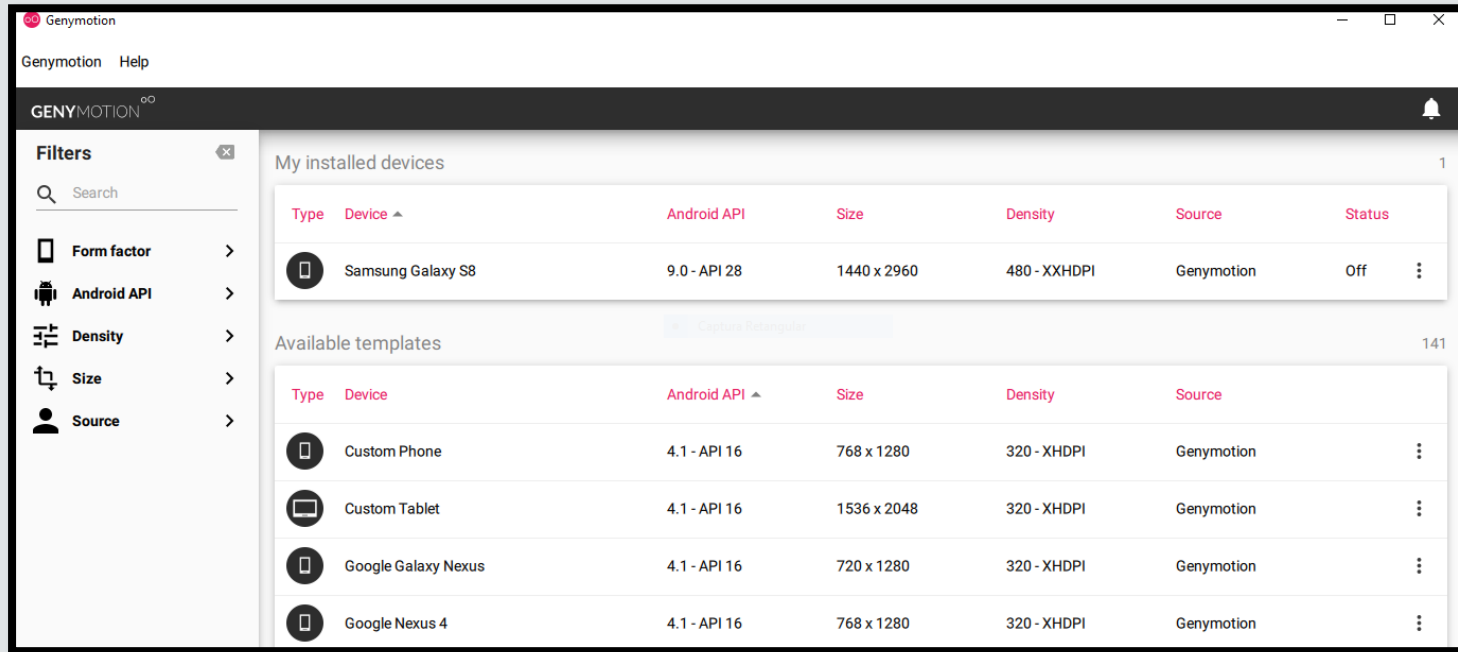
- Para finalizar a criação do aplicativo de demonstração, clique em Concluir.
- Com o aplicativo de demonstração criado, verifique se você tem um dispositivo virtual criado para testar rapidamente o aplicativo.

Criando um Dispositivo Virtual

- Vamos analisar rapidamente as etapas necessárias para criar um dispositivo virtual.



- Eu uso o GenyMotion por ser mais leve.



- Porém, não sei se isso é possível em nossos laboratórios, sendo assim, nesse tutorial vou ensinar como configurar o AVD Manager do Android Studio.
- Com o Android Studio aberto, vá para o menu Ferramentas e clique na opção AVD Manager, que exibirá a seguinte tela.



Your Virtual Devices

Android Studio




Virtual devices allow you to test your application without having to own the physical devices.

+ Create Virtual Device...

To prioritize which devices to test your application on, visit the [Android Dashboards](#), where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.

- Em seguida, clique em Create Virtual Device, que exibirá a seguinte janela com todos os dispositivos virtuais disponíveis que podem ser criados para diferentes categorias, como: telefones, TV, tablets e dispositivos vestíveis.
- Primeiro selecione o tipo e tamanho do dispositivo:




Virtual Device Configuration




Select Hardware


Android Studio

Choose a device definition

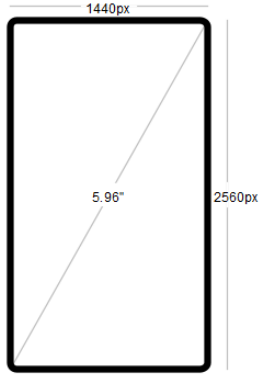
Category	Name ▾	Play Store	Size	Resolution	Density
TV	Pixel		5.0"	1080x1920	420dpi
Phone	Nexus S		4.0"	480x800	hdpi
Wear OS	Nexus One		3.7"	480x800	hdpi
Tablet	Nexus 6P		5.7"	1440x2560	560dpi
	Nexus 6		5.96"	1440x2560	560dpi
	Nexus 5X		5.2"	1080x1920	420dpi
	Nexus 5		4.95"	1080x1920	xxhdpi
	Nexus 4		4.7"	768x1280	xhdpi
	Galaxy Nexus		4.65"	720x1280	xhdpi

New Hardware Profile
Import Hardware Profiles





Nexus 6



Size: large
Ratio: long
Density: 560dpi

Clone Device...

Previous

Next

Cancel

Finish

Help

- Vou selecionar o Samsung Galaxy 8 na categoria Telefone, mas fique à vontade para escolher outro.
- É importante escolher uma imagem que funcione bem com o sistema operacional host do seu computador.
- Em essência, por razões de desempenho do emulador, não é recomendável escolher uma imagem baseada em ARM se o sistema operacional host do seu computador for baseado em uma arquitetura x86.
 - Pode ser necessário fazer o download da imagem, usando o link Download ao lado do Nome da API.



System Image

Android Studio

Select a system image

Recommended x86 Images Other Images

Release Name	API Level ▾	ABI	Target
Pie	28	x86	Android 9.0 (Google APIs)
Oreo	27	x86	Android 8.1 (Google APIs)
<i>Oreo</i> Download	26	x86	Android 8.0 (Google APIs)
<i>Nougat</i> Download	25	x86	Android 7.1.1 (Google APIs)
<i>Nougat</i> Download	24	x86	Android 7.0 (Google APIs)
<i>Marshmallow</i> Download	23	x86	Android 6.0 (Google APIs)
<i>Lollipop</i> Download	22	x86	Android 5.1 (Google APIs)



Pie



API Level

28

Android

9.0**Google Inc.**

System Image

x86

We recommend these images because they run the fastest and support Google APIs.

Questions on API level?

See the [API level distribution chart](#)

Previous

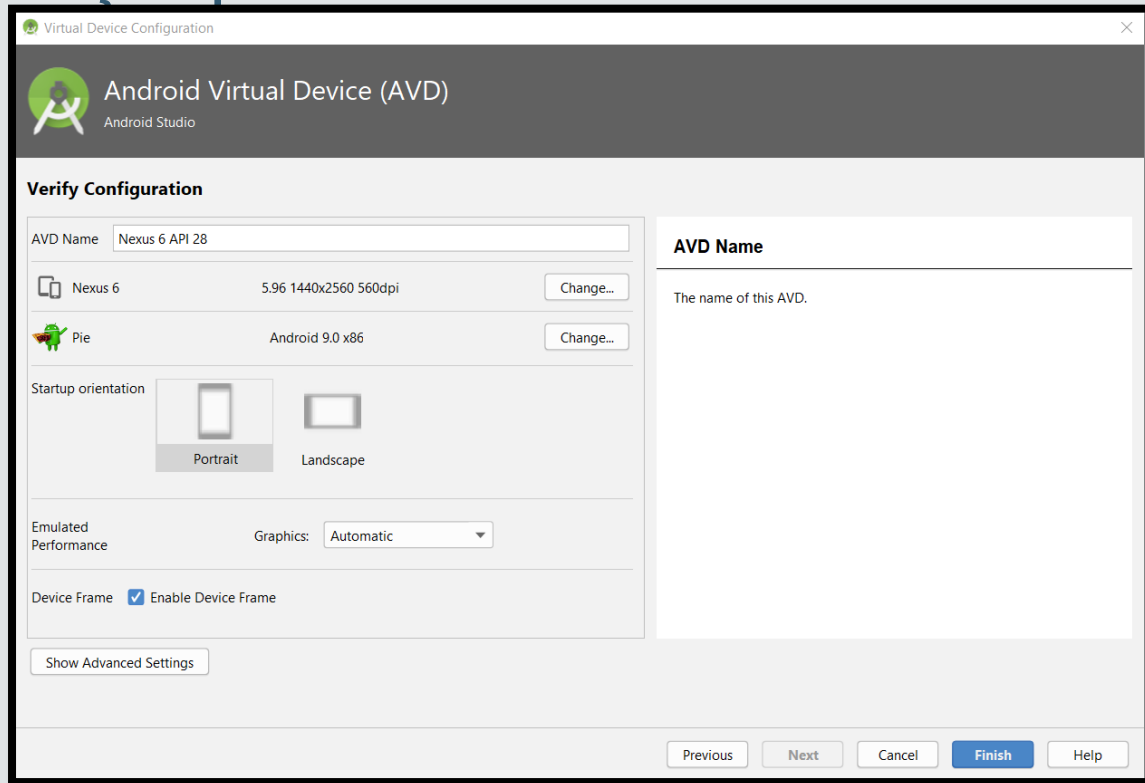
Next

Cancel

Finish

Help

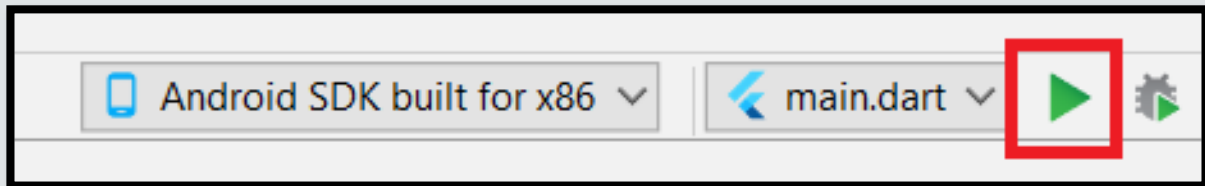
- Depois que a imagem for selecionada (e baixada, se necessário), clique em **Avançar** para continuar.



- A última etapa contém os detalhes de configuração da imagem do dispositivo virtual, que você pode deixar normalmente nas opções padrão e, em seguida, simplesmente clicar em Concluir.
- Você pode criar mais de um, se desejar, pois isso pode ajudá-lo a testar seu aplicativo com vários dispositivos.

Testando a Configuração

- Com o nosso dispositivo virtual instalado, agora é hora de executar o aplicativo que criamos e ver o que ele faz.
- Para fazer isso, verifique se o dispositivo está selecionado e clique em executar.



- Tente executar o aplicativo para ver o que acontece - no meu caso, recebo a seguinte saída do console do Android Studio.

```

Launching lib\main.dart on Android SDK built for x86 in debug mode...
Initializing gradle...
Resolving dependencies...

* Error running Gradle:
ProcessException: Process
"C:\Projects\test\flutter_app\android\gradlew.bat" exited abnormally:

> Configure project :app
Checking the license for package Android SDK Build-Tools 28.0.3 in
C:\Users\EdFreitas\AppData\Local\Android\sdk\licenses
Warning: License for package Android SDK Build-Tools 28.0.3 not accepted.

FAILURE: Build failed with an exception.

* Where:
Build file 'C:\Projects\test\flutter_app\android\build.gradle' line: 24

* What went wrong:
A problem occurred evaluating root project 'android'.
> A problem occurred configuring project ':app'.
    > Failed to install the following Android SDK packages as some
    licenses have not been accepted.
        build-tools;28.0.3 Android SDK Build-Tools 28.0.3
    To build this project, accept the SDK license agreements and install
    the missing components using the Android Studio SDK Manager.
    Alternatively, to transfer the license agreements from one
    workstation to another, see http://d.android.com/r/studio-ui/export-
    licenses.html

    Using Android SDK: C:\Users\EdFreitas\AppData\Local\Android\sdk

* Try:

```

Run with --stacktrace option to get the stack trace. Run with --info or -
-debug option to get more log output. Run with --scan to get full
insights.

* Get more help at <https://help.gradle.org>

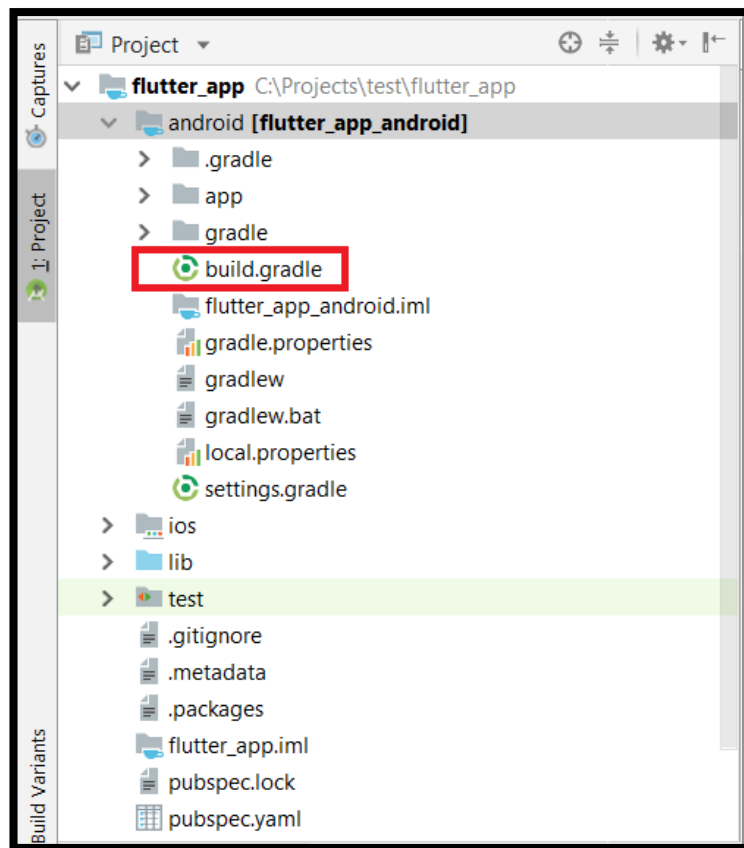
BUILD FAILED in 1s

Command: C:\Projects\test\flutter_app\android\gradlew.bat
app:properties

Finished with error: Please review your Gradle project setup in the
android/ folder.

- Se você não recebeu esta mensagem de saída do console depois de executar seu aplicativo, incrível - você pode ir direto para a parte "Hot Reload".
- Se você fez isso, analisando cuidadosamente essas informações de saída, encontramos a referência <http://d.android.com/r/studio-ui/export-licenses.html> URL que redireciona para aqui.

- Este material explica como o Gradle pode baixar automaticamente pacotes que podem estar ausentes e que são necessários ao executar um aplicativo.
- Também podemos ver que a última linha da mensagem indica a revisão da configuração do projeto Gradle na pasta do projeto Android - que se refere ao arquivo Build.gradle encontrado na pasta Android do nosso aplicativo. Podemos ver isso na captura de tela a seguir.



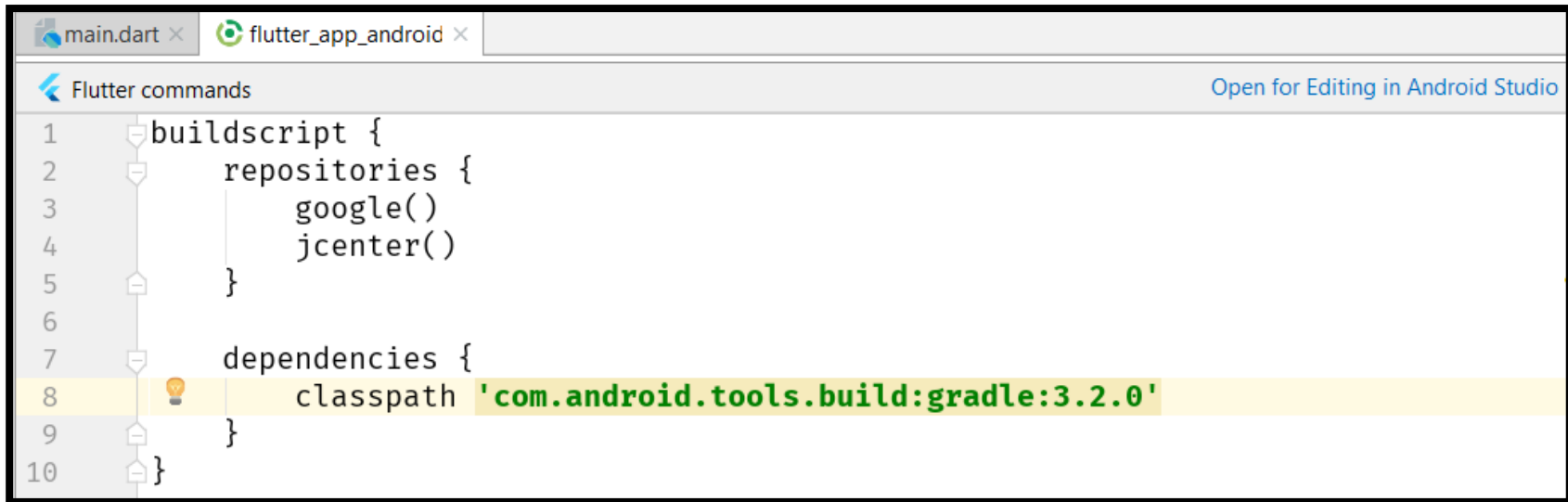
- Vamos abrir o arquivo Build.gradle e inspecionar seu conteúdo. Observe no classpath a dependência com.android.tools.build:gradle:3.2.1.

```
1 buildscript {
2     repositories {
3         google()
4         jcenter()
5     }
6
7     dependencies {
8         classpath 'com.android.tools.build:gradle:3.2.1'
9     }
10 }
11
12 allprojects {
13     repositories {
14         google()
15         jcenter()
16     }
17 }
```


- A saída anterior do console fornece uma dica do que pode ser o problema: Licença para pacote Android SDK Build-Tools 28.0.3 não aceita.
- O que isso está nos dizendo é que não foi possível executar o aplicativo com essa dependência
`com.android.tools.build:gradle:3.2.1`, porque não tenho licença para isso.

- Em outras palavras, essa versão específica `com.android.tools.build:gradle` provavelmente não foi instalada quando eu passei pelo processo de instalação do Android Studio.

- Nesse caso, a solução é usar uma versão da dependência `com.android.tools.build:gradle` que foi instalada durante o processo de instalação do Android Studio - que pode ser uma versão inferior à mencionada no arquivo `Build.gradle`.
- Para resolver o problema, tudo o que preciso fazer é alterar essa linha no arquivo `Build.gradle`
`com.android.tools.build:gradle:3.2.1` para
`com.android.tools.build:gradle:3.2.0`.



The screenshot shows an IDE window with two tabs: 'main.dart' and 'flutter_app_android'. The 'flutter_app_android' tab is active and displays a 'Flutter commands' panel. The panel contains a buildscript configuration for a Flutter Android app. The configuration is as follows:

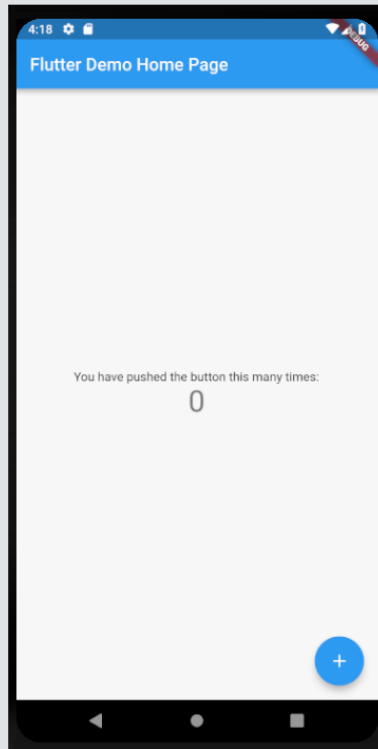
```
1 buildscript {  
2     repositories {  
3         google()  
4         jcenter()  
5     }  
6  
7     dependencies {  
8         classpath 'com.android.tools.build:gradle:3.2.0'  
9     }  
10 }
```

The line containing the classpath dependency is highlighted in yellow. A lightbulb icon is visible next to the classpath dependency, indicating a suggestion or warning. A link 'Open for Editing in Android Studio' is visible in the top right corner of the panel.

- Depois de salvar a alteração no arquivo Build.gradle, se agora clicar em Executar, obterá a seguinte saída do console de compilação no Android Studio.

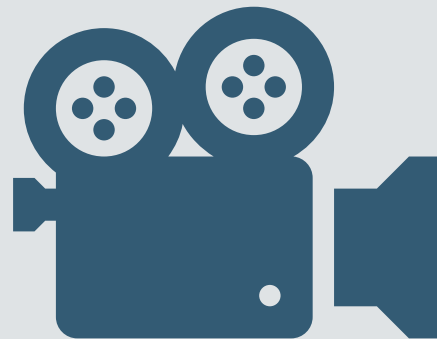
```
Launching lib\main.dart on Android SDK built for x86 in debug mode...  
Initializing gradle...  
Resolving dependencies...  
Gradle task 'assembleDebug'...  
Built build\app\outputs\apk\debug\app-debug.apk.
```

- O aplicativo foi criado e está em execução. Podemos ver isso no emulador do Android da seguinte maneira:



Vídeos

- Flutter em 6 minutos - https://www.youtube.com/watch?v=jbAh5R8CH_o



Dicas para Estudo



Seja “CURIOSO”:

Procure revisar o que foi estudado.

Pesquise as referências bibliográficas.



Seja “ANTENADO”:

Leia a próxima aula.



Seja
“COLABORATIVO”:

Traga assuntos relevantes para a sala de aula.

Participe da aula.

Proponha discussões relevantes sobre o conteúdo.



Prof. Wilson Lourenço



**Dúvidas?
Não mais..**