

Aula XI - Desenvolvimento Aplicativo I

Sistemas Operacionais Mobile

O SO Android

Android Studio

A Interface e criando projeto no Android Studio

Configuração do emulador do Android Studio

Detalhes da Interface

Prática

Fontes:

<https://developer.android.com/docs?hl=pt-br> -- DOCUMENTAÇÃO

<https://developer.android.com/training/basics/firstapp?hl=pt-br>

<https://distancia.qi.edu.br/mod/book/view.php?id=54366&chapterid=39093>

<https://www.devmedia.com.br/guia/linguagem-kotlin/40739>

Sistemas Operacionais Mobile

Atualmente, os dois principais sistemas operacionais para smartphones são o Android e iOS. Estes sistemas são desenvolvidos por companhias diferentes que brigam pelo domínio dessa área.

Algumas diferenças: [A batalha Android vs iOS: quais suas principais diferenças? | RMA](#)

Contexto histórico

Apesar de ter uma vida um pouco mais longa como sistema operacional (desde 2003), o Android se tornou relevante em 2007 quando a Google anunciou seus planos de disponibilização livre de um sistema operacional para dispositivos móveis com características semelhantes ao recém lançado iPhone.

Por ser um sistema operacional livre, a intenção da empresa foi de buscar parceiros, através da Open Handset Alliance, para fabricação do hardware (telefone celular) e embarcar no seu sistema operacional livre.

Comercialmente isso aconteceu em 2008 quando a HTC lançou o seu primeiro telefone com o sistema operacional Android.

O SO Android

O sistema operacional Android é hoje líder absoluto nos tablets e smartphones disponíveis no mercado mundial. Lançado pelo Google com o intuito claro de disputar com o iPhone da Apple, a plataforma móvel do “robozinho” ganhou uma enorme fatia do mercado por várias (e merecidas) razões:

- Possui uma interface agradável e fácil de usar, que conquista tanto usuários novatos como os mais experientes;
- Sua arquitetura aberta possibilita que o mesmo seja utilizado em aparelhos de vários fabricantes;
- Utiliza a linguagem Java para criação de seus aplicativos.

Visão geral do armazenamento de dados e arquivos

O sistema operacional **Android utiliza Java (ou Kotlin)** como a linguagem de programação para seus aplicativos, e traz um conjunto rico de classes e ferramentas para o desenvolvimento de programas que faz uso dos principais recursos existentes nestes aparelhos.

Kotlin é uma Linguagem de programação multiplataforma, orientada a objetos e funcional, concisa e estaticamente tipada, desenvolvida pela JetBrains em 2011, que compila para a Máquina virtual Java e que também pode ser traduzida para a linguagem JavaScript e compilada para código nativo.

Android Studio

O Android Studio é o ambiente de desenvolvimento integrado (IDE, na sigla em inglês) oficial para o desenvolvimento de apps para Android e é baseado no IntelliJ IDEA (link em inglês).

Além do editor de código e das ferramentas de desenvolvedor avançadas do IntelliJ, o Android Studio oferece ainda mais recursos para aumentar sua produtividade na compilação de apps Android, como:

- Um sistema de compilação flexível baseado em Gradle
- Um emulador rápido com inúmeros recursos
- Um ambiente unificado que possibilita o desenvolvimento para todos os dispositivos Android
- A aplicação de alterações para enviar alterações de código e recursos ao aplicativo em execução sem reiniciar o aplicativo
- Modelos de código e integração com GitHub para ajudar a criar recursos comuns de apps e importar exemplos de código
- Frameworks e ferramentas de teste cheios de possibilidades
- Ferramentas de lint para detectar problemas de desempenho, usabilidade, compatibilidade com versões, entre outros
- Compatibilidade com C++ e NDK
- Compatibilidade integrada com o Google Cloud
- Platform, facilitando a integração do Google Cloud

- Messaging e do App Engine.
- Entrada e saída são implementados através do device drivers. Sempre que um novo dispositivo é acrescentado no sistema, o drive deve ser acoplado ao núcleo.

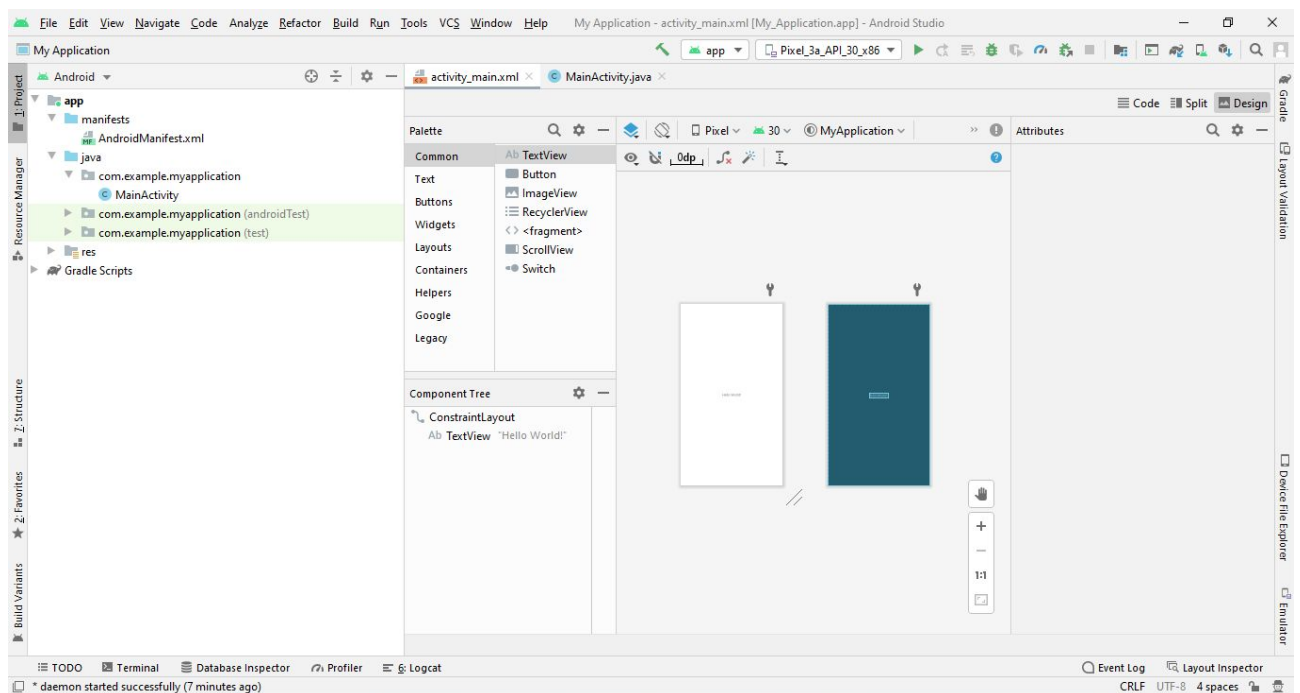
Interface e criando projeto no Android Studio

A estrutura:

Cada projeto no Android Studio contém um ou mais módulos com arquivos de código-fonte e de recursos.

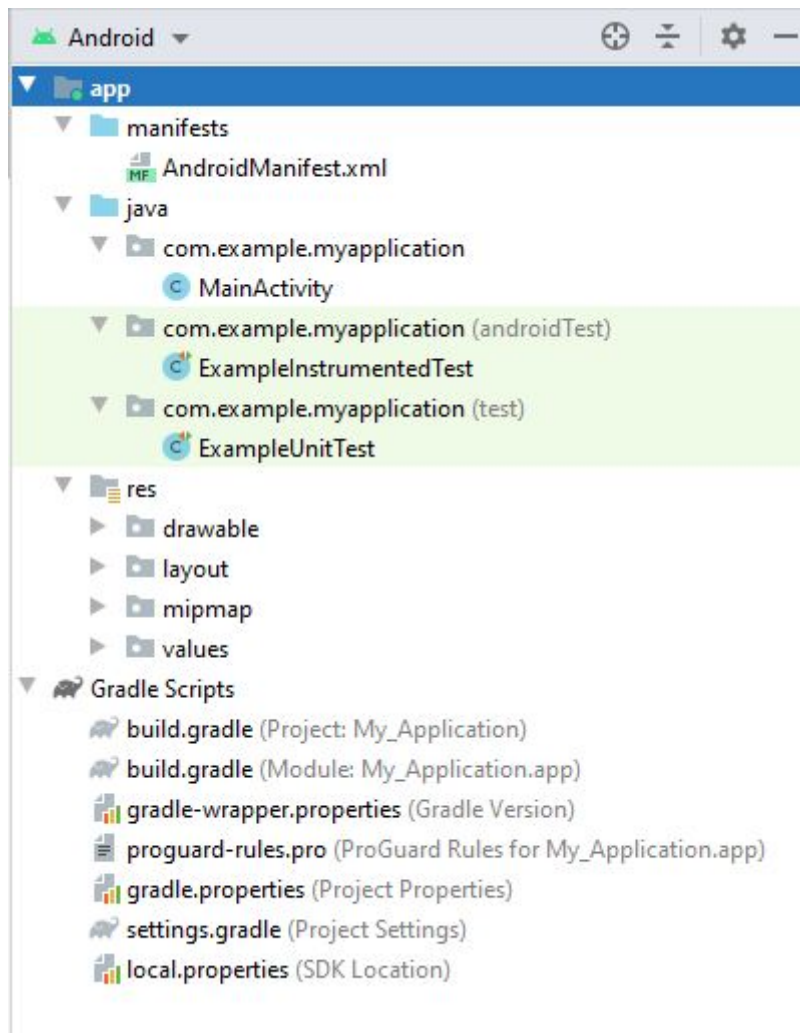
Os tipos de módulos incluem:

- Módulos de apps Android
- Módulos de biblioteca
- Módulos do Google App Engine



Por padrão, o Android Studio exibe os arquivos do projeto na visualização de projetos Android, como mostrado na figura abaixo.

Essa visualização é organizada por módulos para permitir o acesso rápido aos principais arquivos de origem do projeto.



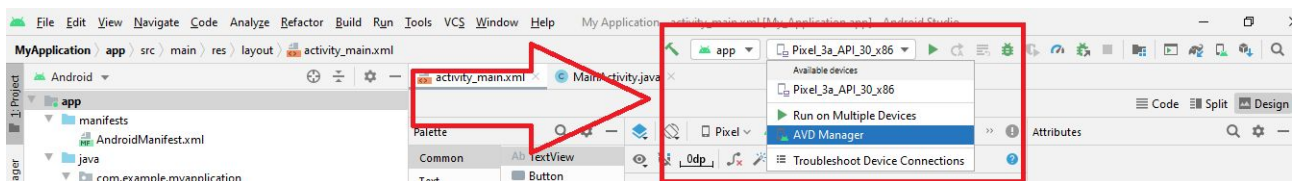
Todos os arquivos de criação podem ser vistos no nível superior em Gradle Scripts, e cada módulo de app contém as pastas a seguir:

- **manifests:** contém o arquivo **AndroidManifest.xml**. Está o arquivo com configuração do aplicativo. Por exemplo, a configuração do ícone do aplicativo.
- **Java:** Encontram-se todos os arquivos java. Dessa forma, o diretório armazena toda a lógica do aplicativo.
- **Recursos:** Estão armazenados todos os arquivos referentes à interface do usuário, podendo conter imagens, ícones, layouts etc.
- **Gradle Script:** estão armazenados todos os scripts responsáveis pela compilação do aplicativo.

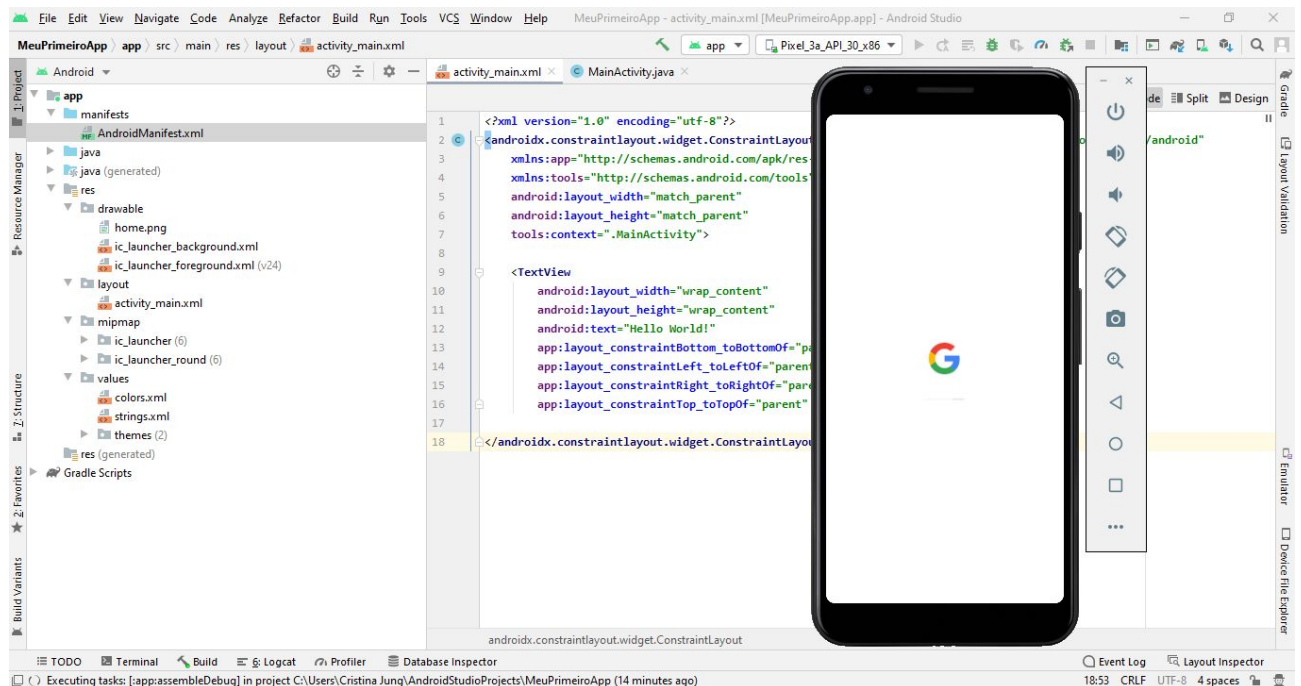
Para navegar na interface do usuário, é preciso clicar duas vezes sobre o arquivo "activity_main.xml"

A Activity é o principal componente de um aplicativo Android. As activities são os componentes associados às telas com que os usuários interagem em um aplicativo.

Configuração do emulador do Android Studio



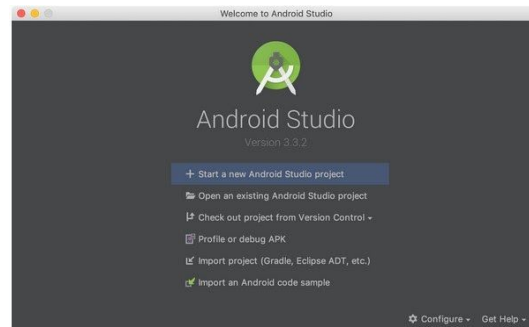
Podemos usar outros emuladores também, pois o AVD é extremamente pesado e requer muito do hardware. Abaixo vemos o emulador do Android Studio.



As opções são usar um dos emuladores que já postei na aula IX ou mesmo fazer a emulação pelo celular, para tanto, precisamos que o celular esteja conectado ao computador via USB.

Vamos fazer o tutorial pelo Genymotion, em aula podemos ver como fazer com os outros.

1. Primeiro vamos abrir um novo Projeto no Android Studio:



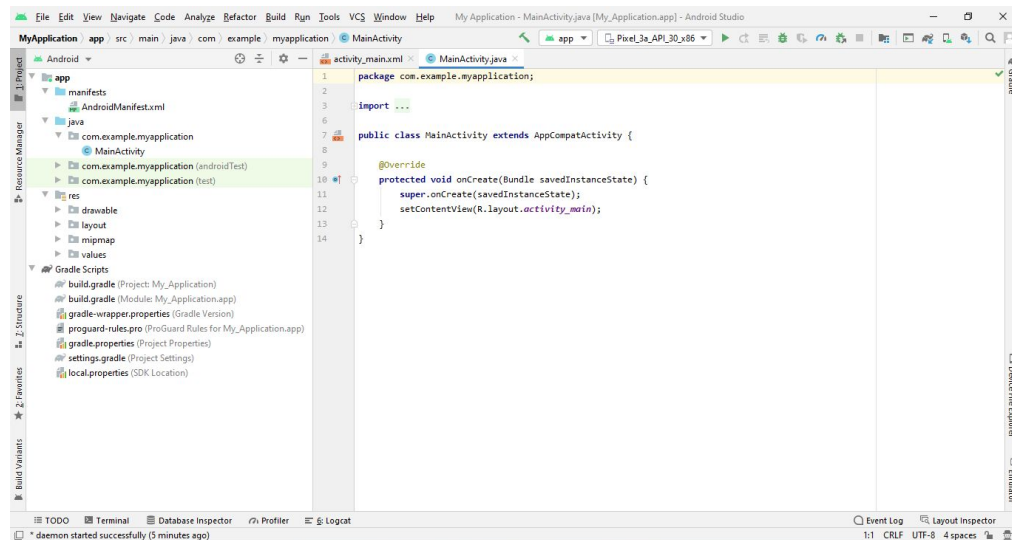
2. Se já estiver aberto:

Caso tenha um projeto aberto, selecione **File > New > New Project**.

Na janela Choose your project, selecionar **Empty Activity**.

Na janela Configure your project, insira os seguintes valores:

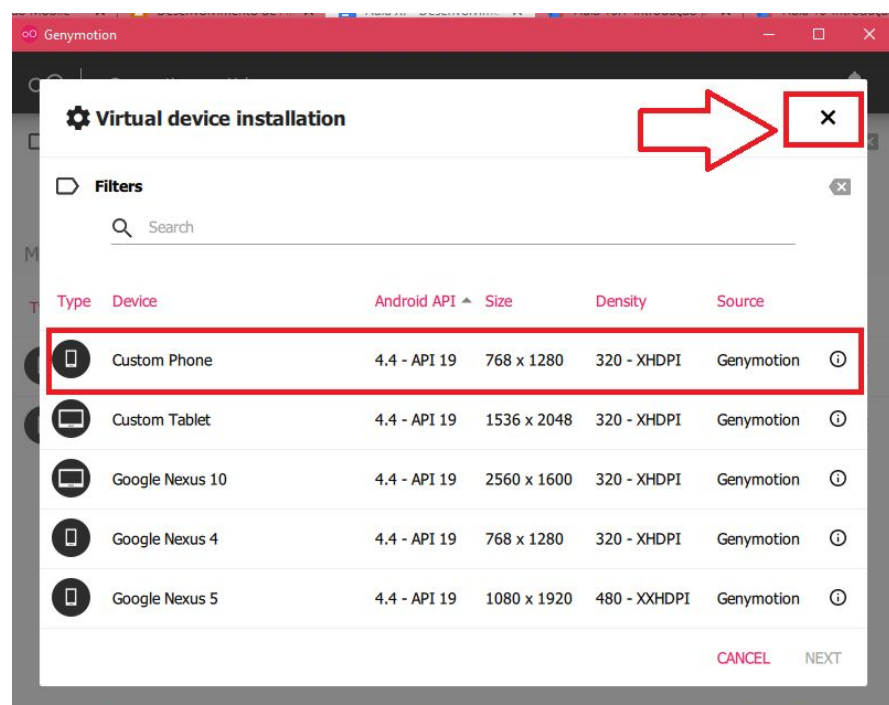
- Nome: **"MyFirstApp"**
- Nome do pacote: **"com.example.myfirstapp"**
- Marque a caixa ao lado de Use AndroidX artifacts.
- Altere o local do projeto se desejar
- Deixe as outras opções como estão. Selecione Java na linguagem.



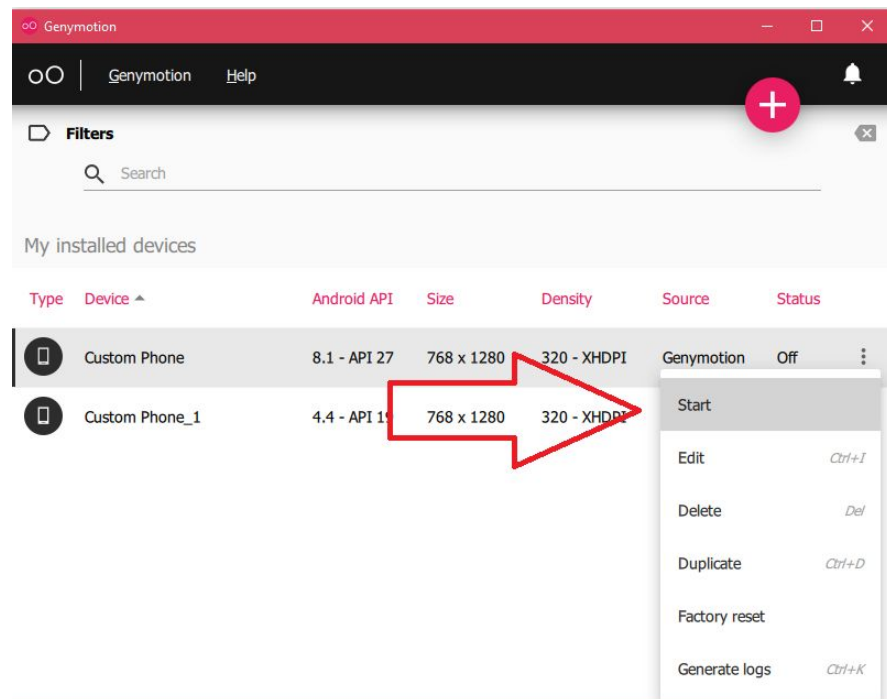
3. Agora vamos abrir o Genymotion. E aguardamos ele dar o start



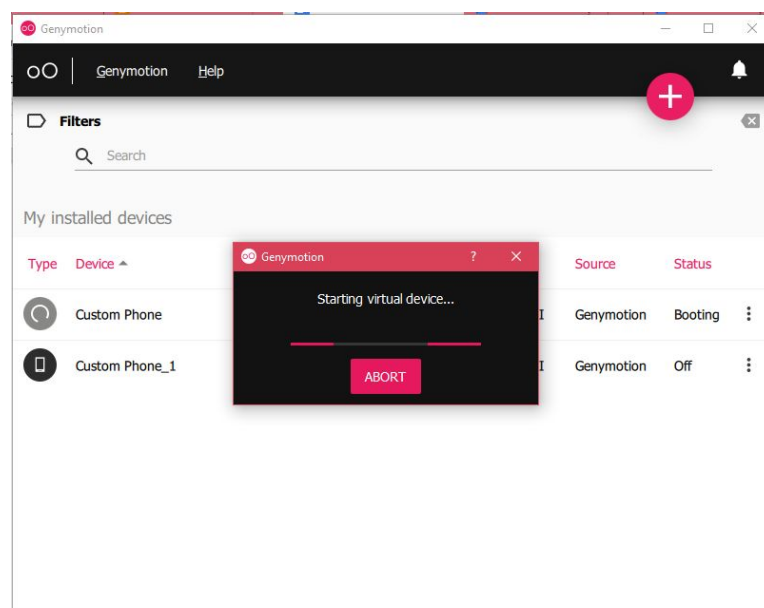
4. Após ele dar o start, vamos ativar o dispositivo, se você não tem nenhum dispositivo adicionado, basta clicar no sinal de + e adicionar o que deseja emular para o Android Studio, neste exemplo, temos um Custom Phone selecionado.



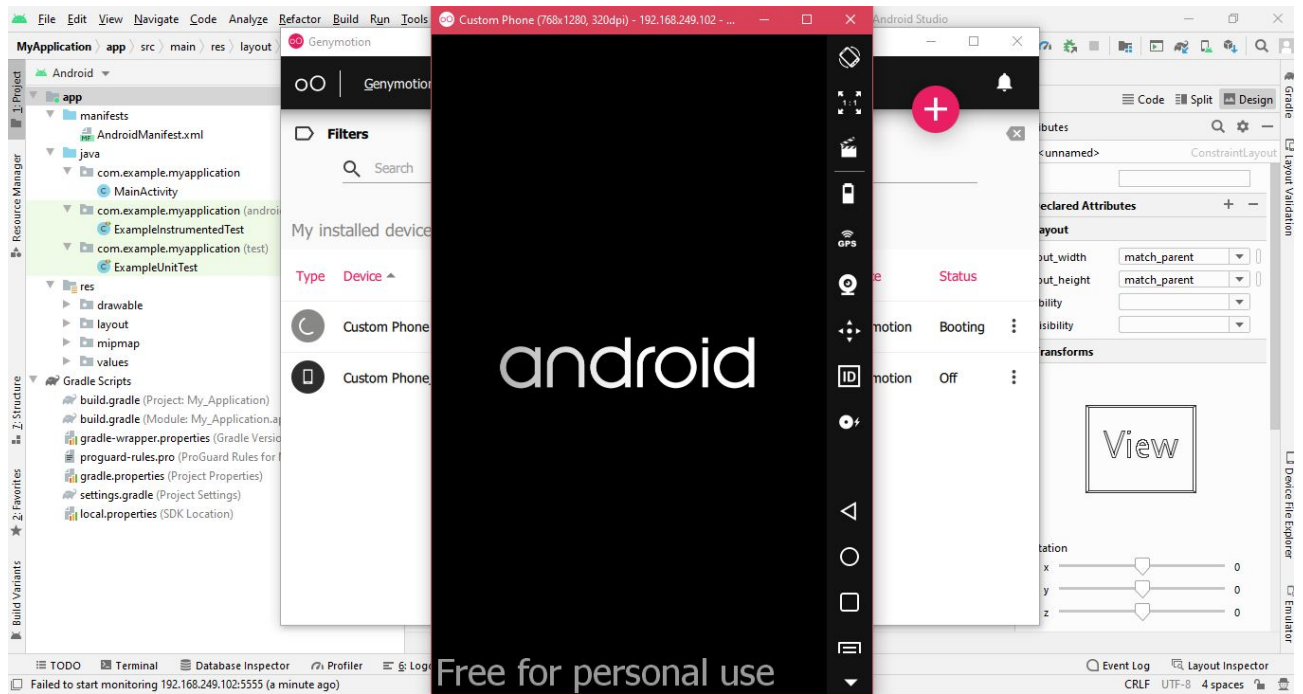
5. Após ele fazer o download, vamos dar o start no dispositivo:



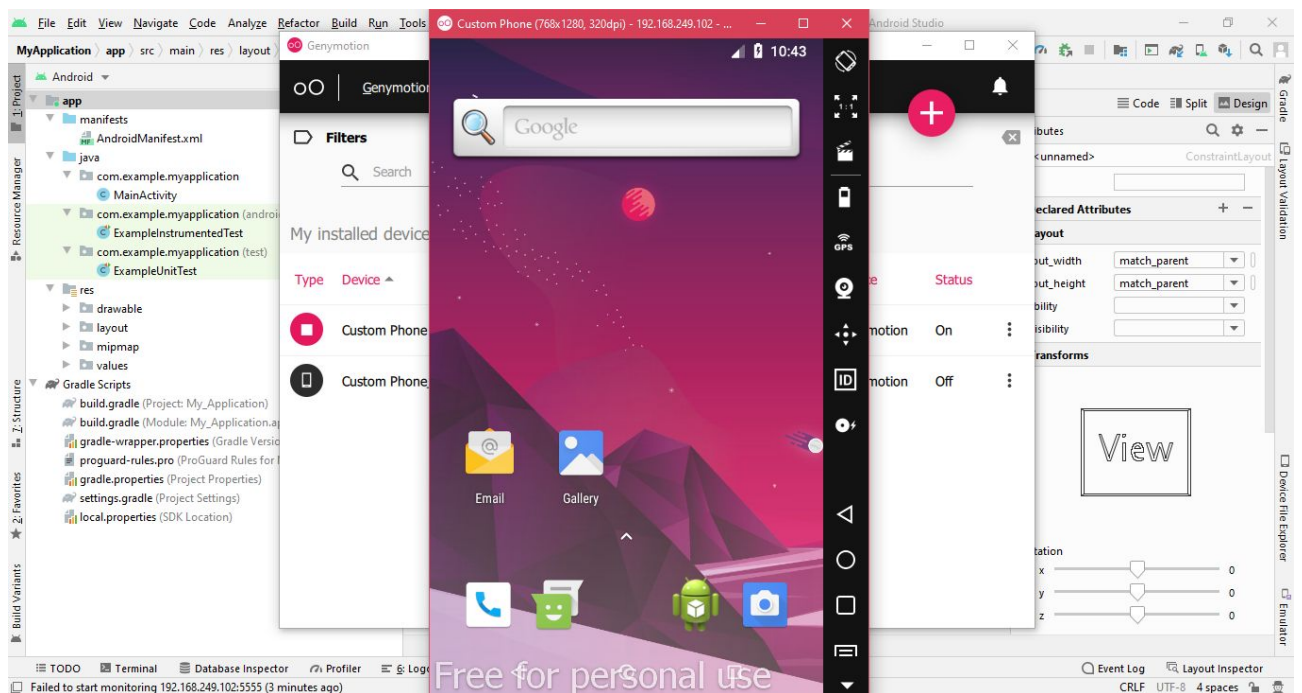
6. Aguardamos o início da virtualização;



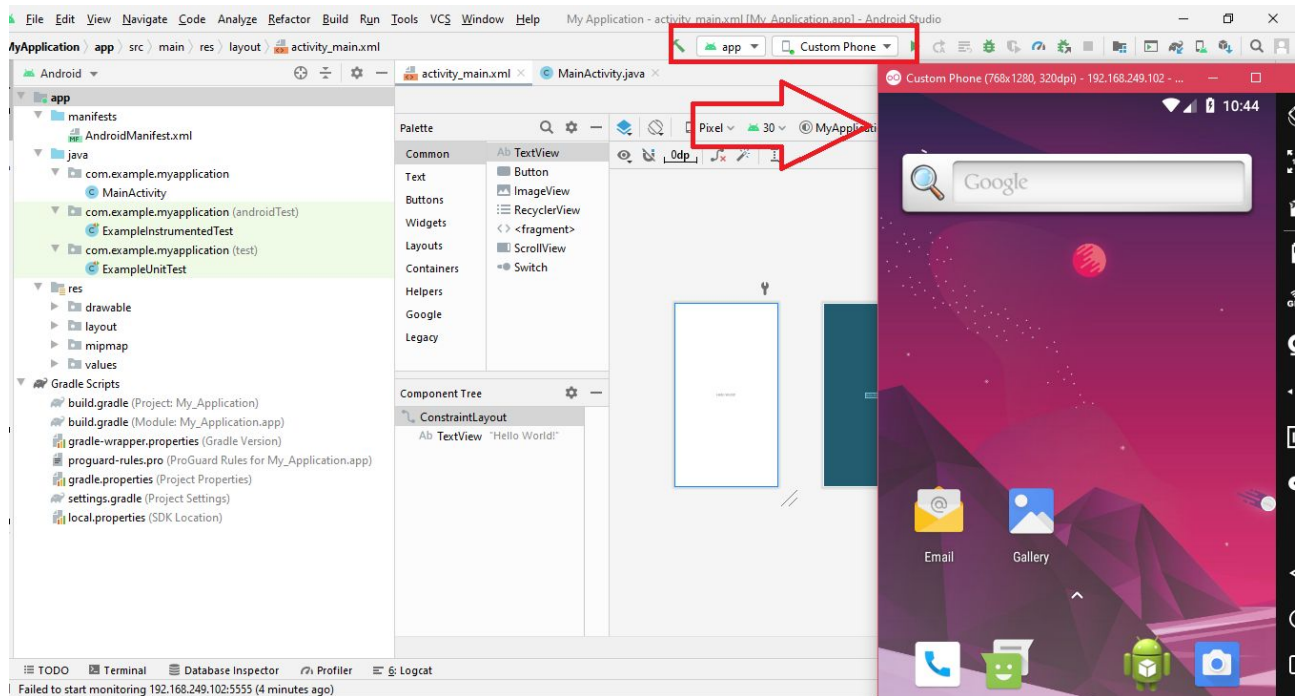
7. O dispositivo começa a ser carregado;



8. O dispositivo está pronto;



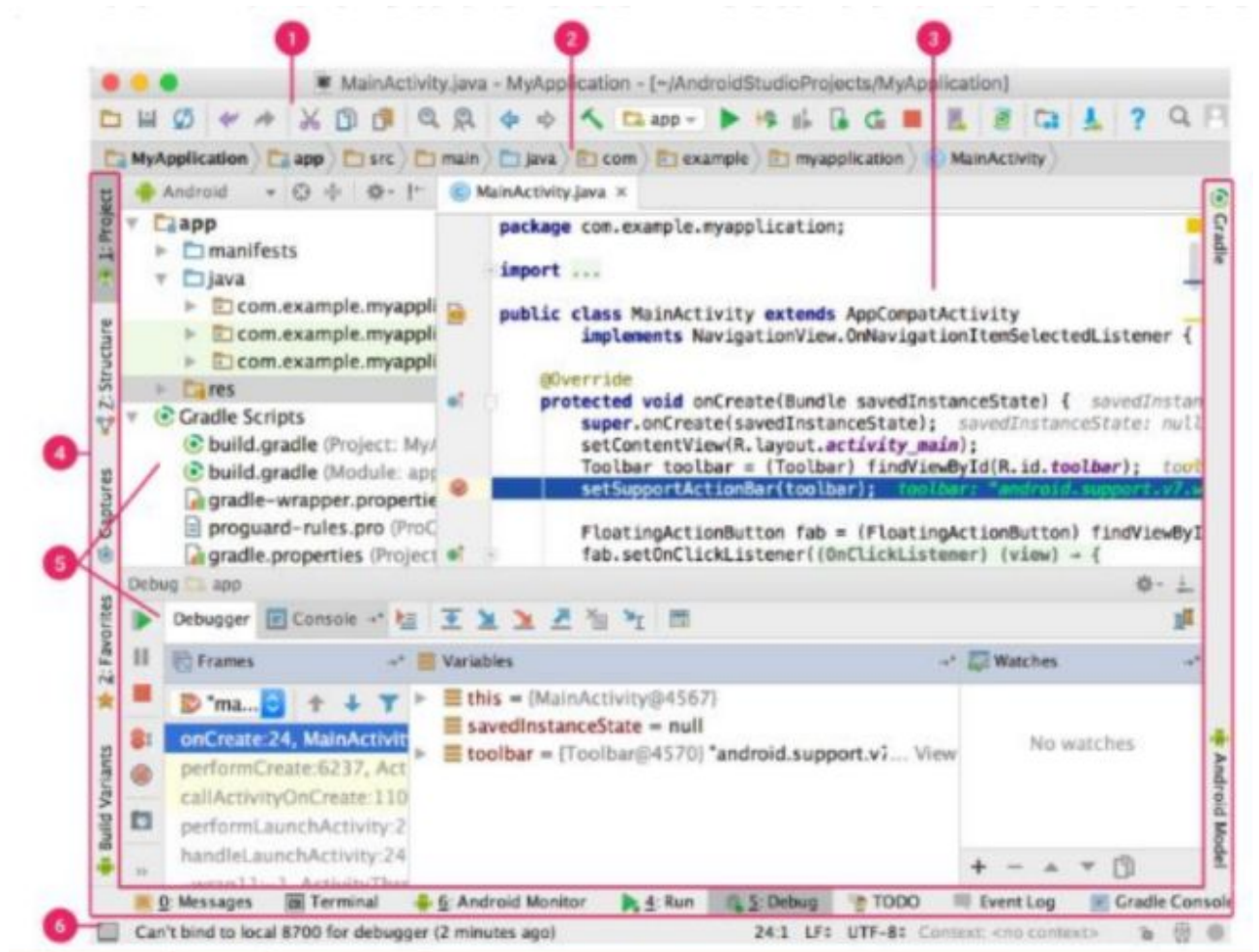
9. O dispositivo aparece no menu de configuração de emuladores;



10. O Android Studio está pronto para ser utilizado.

Detalhes da Interface:

Fonte: <https://distancia.qi.edu.br/mod/book/view.php?id=54357&chapterid=39091>



1. **A barra de ferramentas** permite realizar diversas ações, inclusive executar apps e inicializar ferramentas do Android.

2. **A barra de navegação** ajuda a navegar pelo projeto e a abrir arquivos para edição. Ela oferece uma visualização mais compacta da estrutura visível na janela Project.

3. **A janela do editor** é onde você cria e modifica o código. Dependendo do tipo de arquivo atual, o editor pode mudar. Por exemplo, ao visualizar um arquivo de layout, o editor abre o Editor de layout.

4. **A barra de janela de ferramentas** fica fora da janela do ambiente de desenvolvimento integrado e contém os botões que permitem expandir ou recolher as janelas de cada ferramenta.

5. **As janelas de ferramentas** permitem acessar tarefas específicas, como gerenciamento de projetos, pesquisa e controle de versões, entre outras. As janelas podem ser expandidas e recolhidas.

6. **A barra de status** exibe o status do projeto e do próprio ambiente de desenvolvimento integrado, bem como todos os avisos ou mensagens.