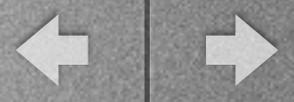




*Universidade Federal de Ouro Preto*  
*Departamento de Computação e Sistemas - DECSI*

**Computação Móvel**  
**Introdução **Android****  
**Ref. Capítulos 1 e 2**

*Vicente Amorim*  
*vicente.amorim.ufop@gmail.com*  
*www.decom.ufop.br/vicente*



## Sumário

---

- \* Introdução **Android**
- \* Primeira Aplicação **Android**
- \* Criação de um Emulador **Android (AVD)**
- \* Execução de Aplicativo no **AVD**
- \* Principais Janelas do **Android Studio**.
- \* **Android Debug Bridge (ADB)**.



# Introdução **Android**

---



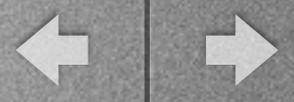
# Introdução **Android**

---

- O que é??

✓ “**Android** is the world’s most popular mobile platform...**Android** devices are already smart (and will only get smarter)... blah, blah, blah...” (Google)

✓ “Sistema operacional baseado em Linux desenvolvido especialmente para dispositivos móveis” (Wikipedia).



# Introdução **Android**

---

- O que é??



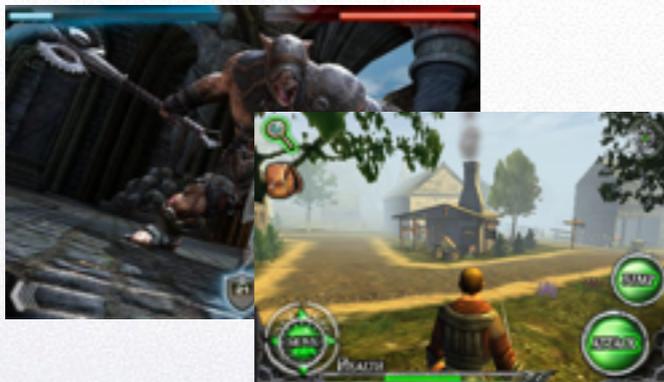
✓ **Framework** para construção de aplicações!!



# Introdução Android

- **Framework** para construção de aplicações:
  - ✓ APIs para: Construções gráficas, notificações, navegação, acesso a web, comunicação interprocessos, gerência de memória, rede (wi-fi, 3G, GPRS, BT, GPS, NFC...), etc.

✓ OpenGL -> Jogos!!! :-D



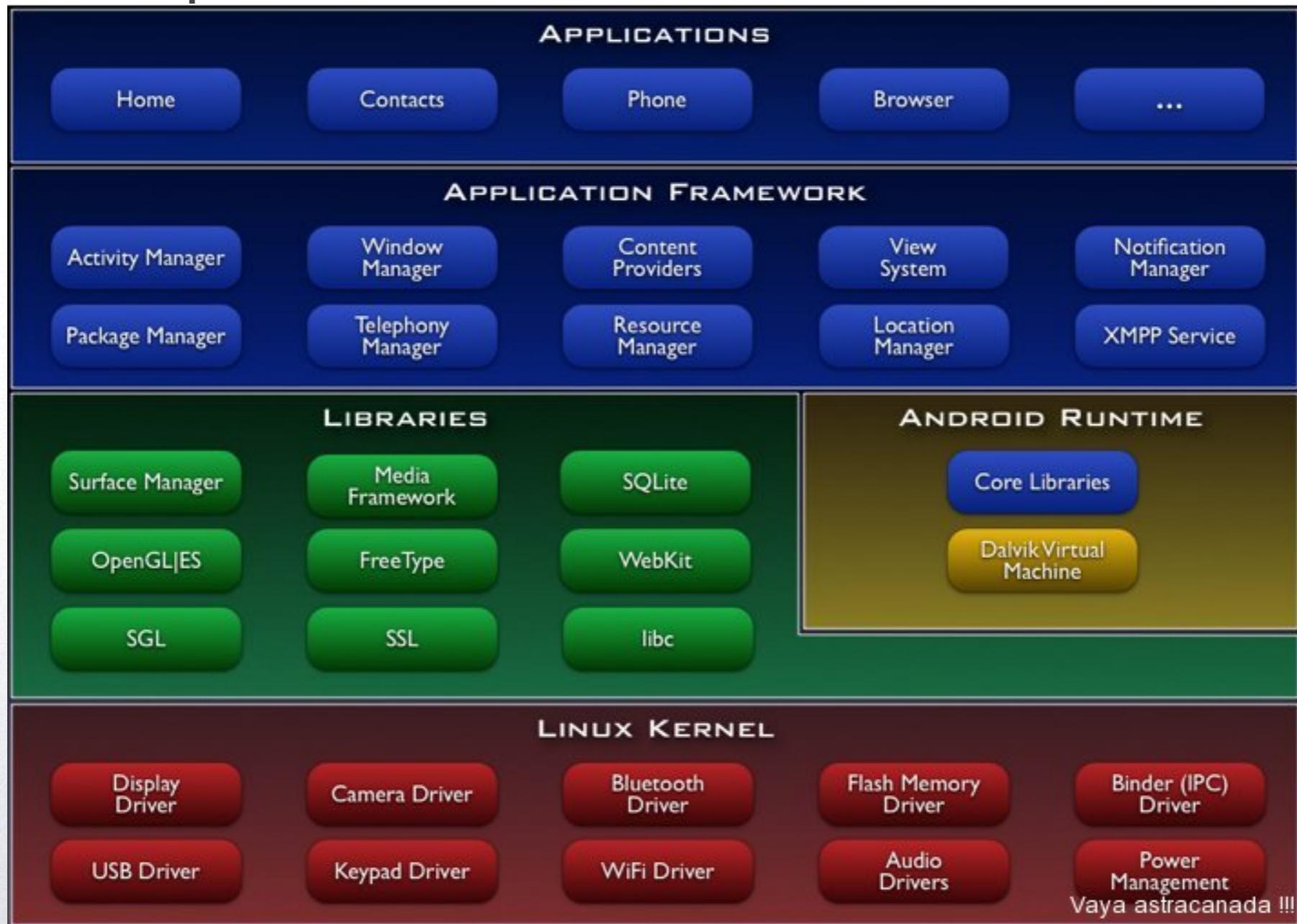
✓ Google Play Store!!





# Introdução **Android**

## - Arquitetura e Aplicabilidade





# Introdução **Android**

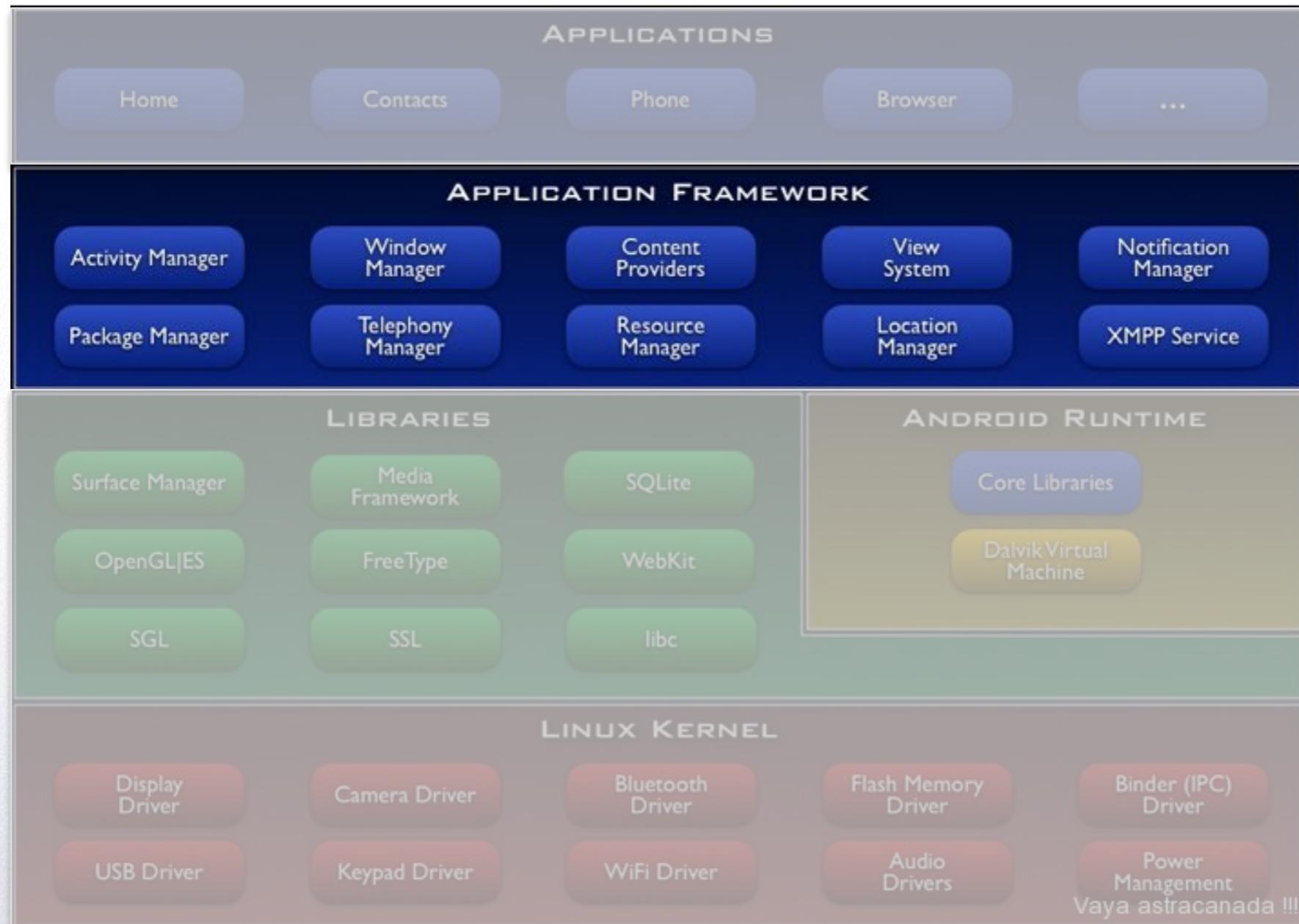
## - Arquitetura e Aplicabilidade





# Introdução **Android**

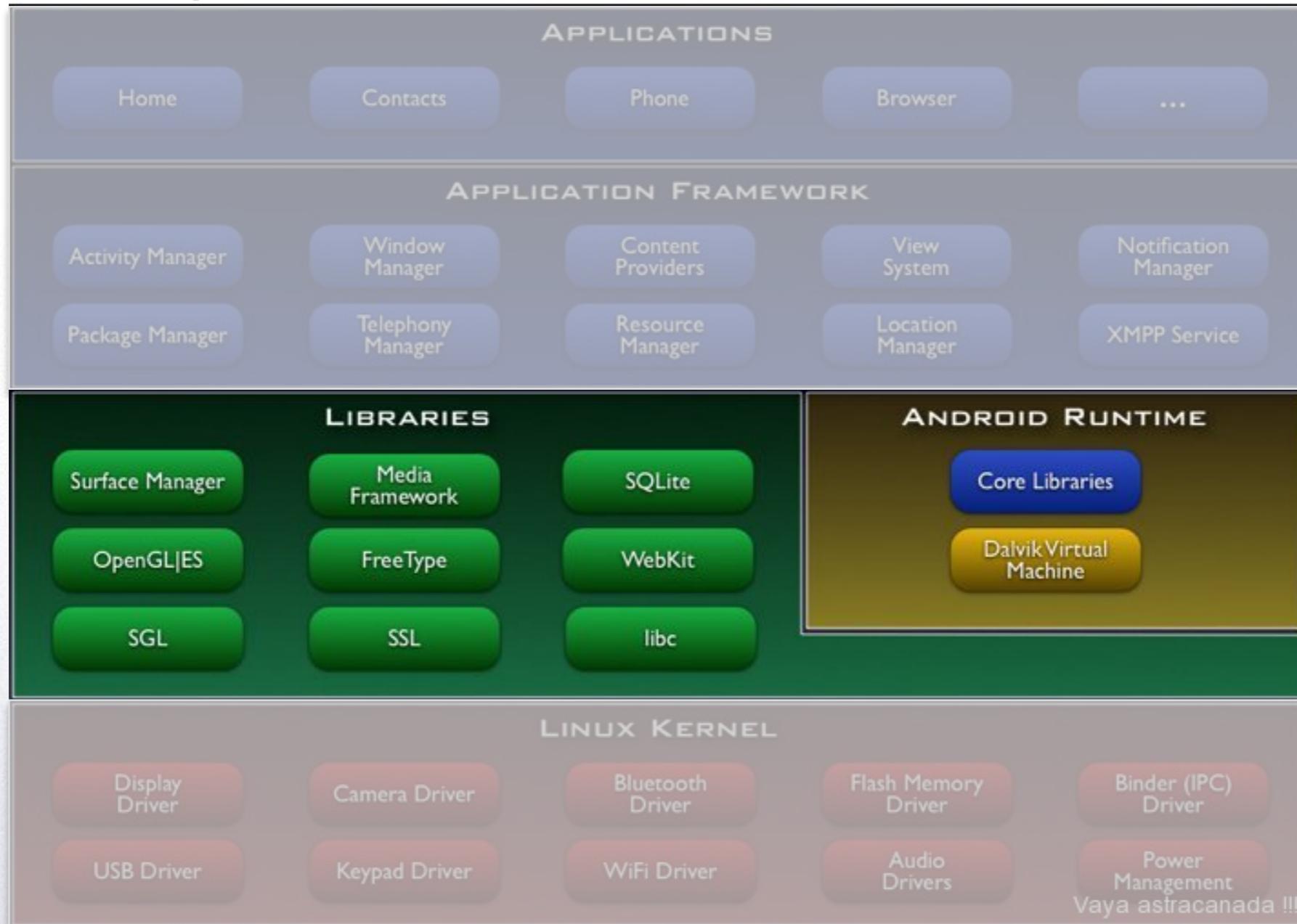
## - Arquitetura e Aplicabilidade





# Introdução **Android**

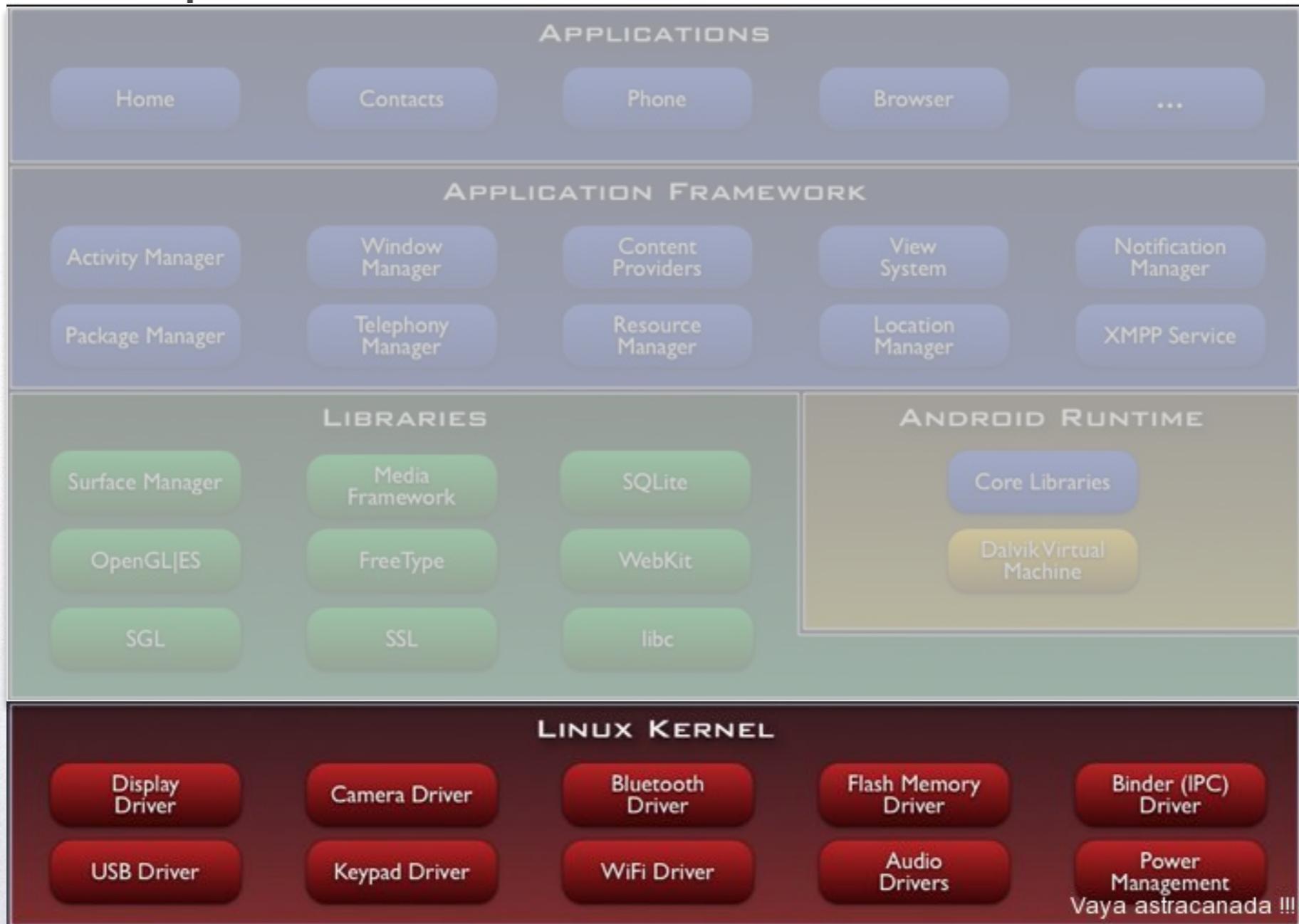
## - Arquitetura e Aplicabilidade





# Introdução **Android**

## - Arquitetura e Aplicabilidade





# Introdução **Android**

- Arquitetura e Aplicabilidade

✓ É código-aberto!! 

✓ Posso modificar o código e criar um **Android** só pra mim???

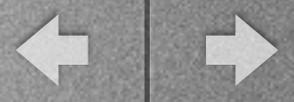


✓ Posso estender e criar aplicações específicas para as áreas em que trabalho???



✓ Posso criar aplicações para áreas específicas???  
(automotiva, médica, prospecção mineral, etc) ??





# Introdução **Android**

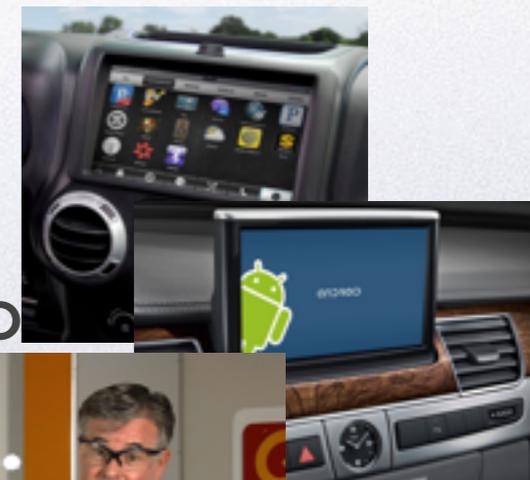
## - Arquitetura e Aplicabilidade

✓ Mas... E se eu precisar de alguma funcionalidade que não está presente no “bendito” *framework* **Android**? Como faço??

✓ “Comé” que é??



✓ Exemplo: Conectar minha aplicação **Android** a uma rede CAN de um automóvel para obter dados em RT do mesmo... É possível???

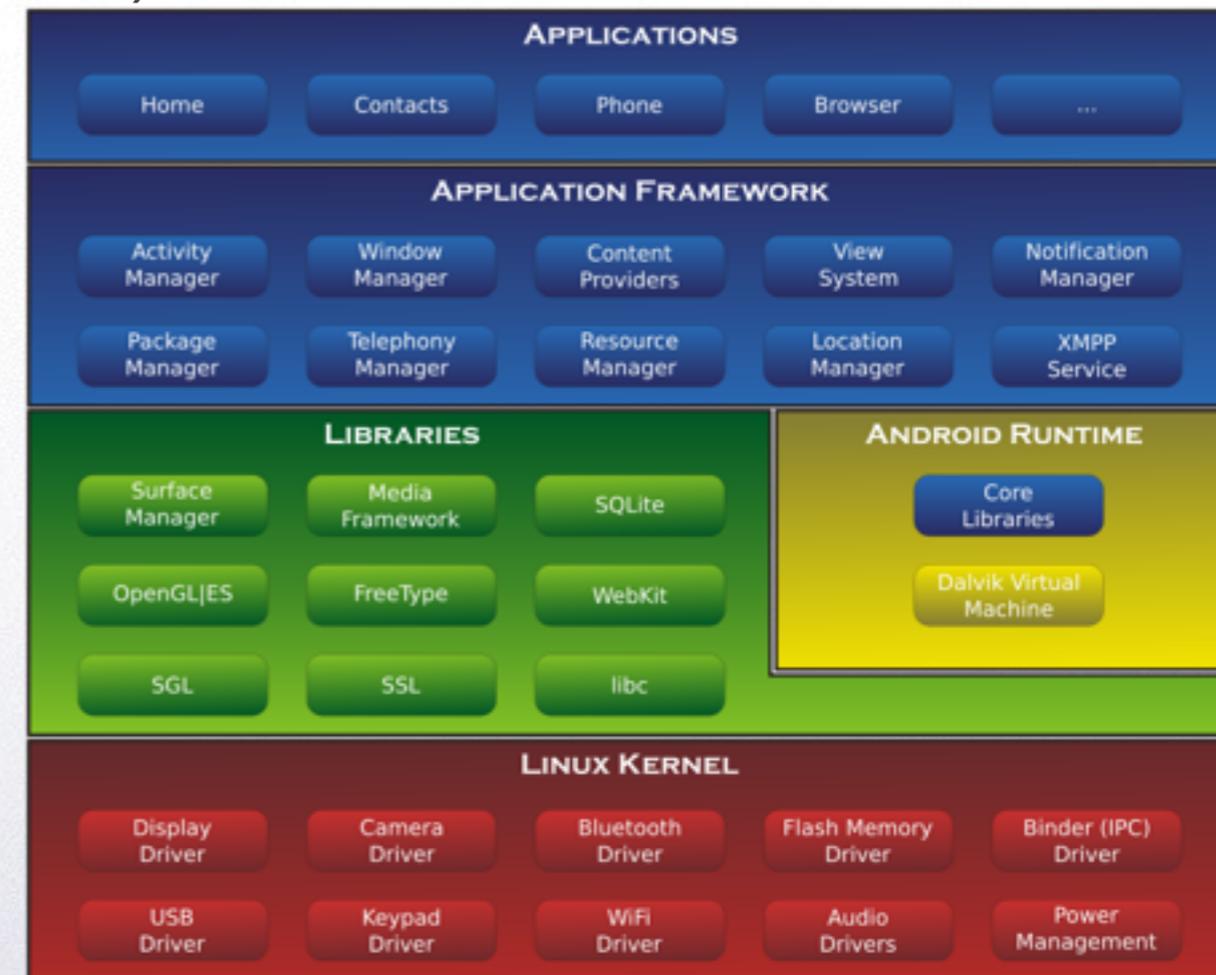




# Introdução **Android**

- **Android** Native Development Kit (NDK)
  - ✓ Conjunto de ferramentas que permite ao desenvolvedor incluir em suas aplicações componentes em código-nativo (C/C++).

- ✓ Principais vantagens:
  - ▶ Reuso de código; e
  - ▶ Em alguns casos, aumento na performance.





# Primeira Aplicação **Android**

---



# Primeira Aplicação **Android**

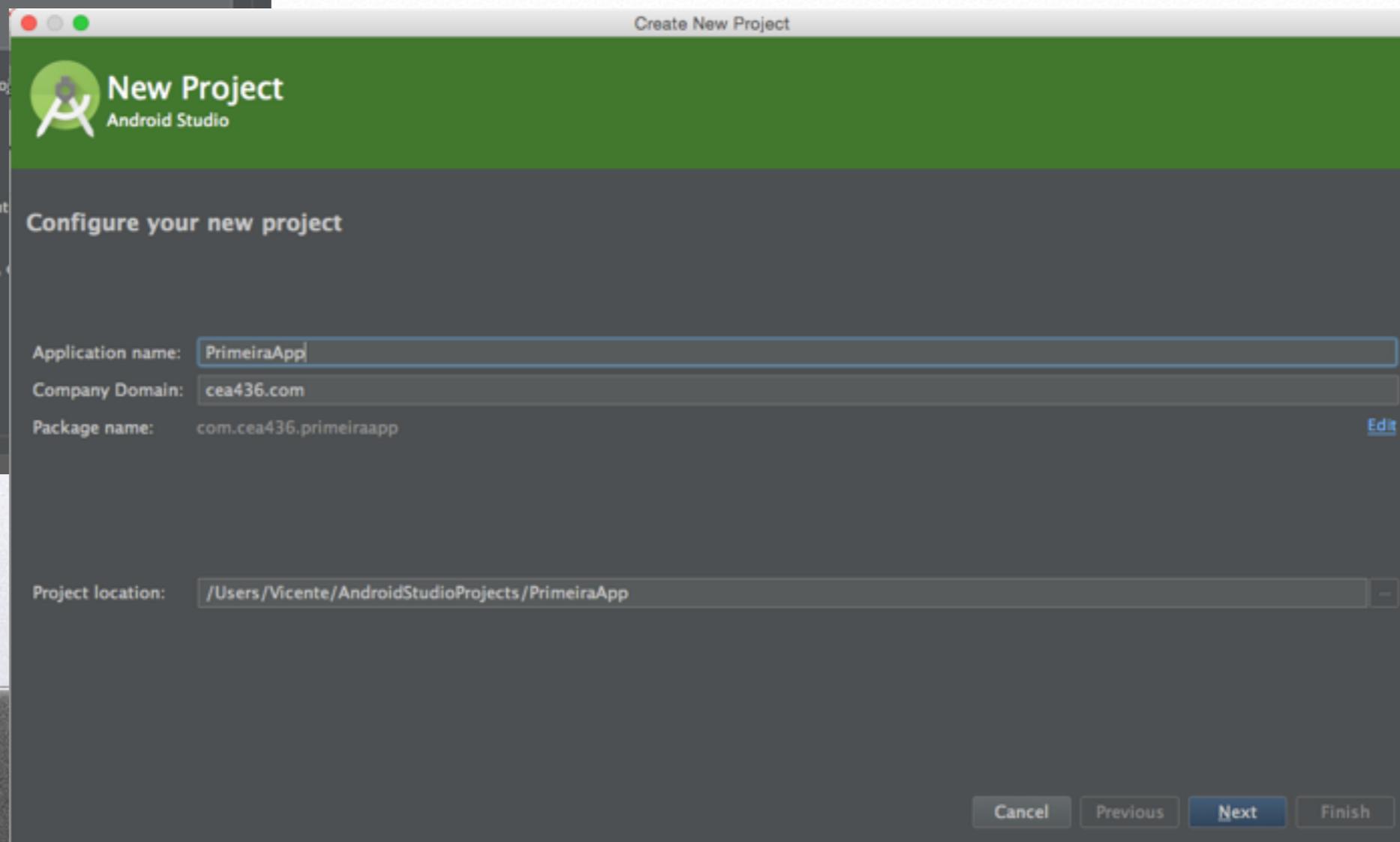
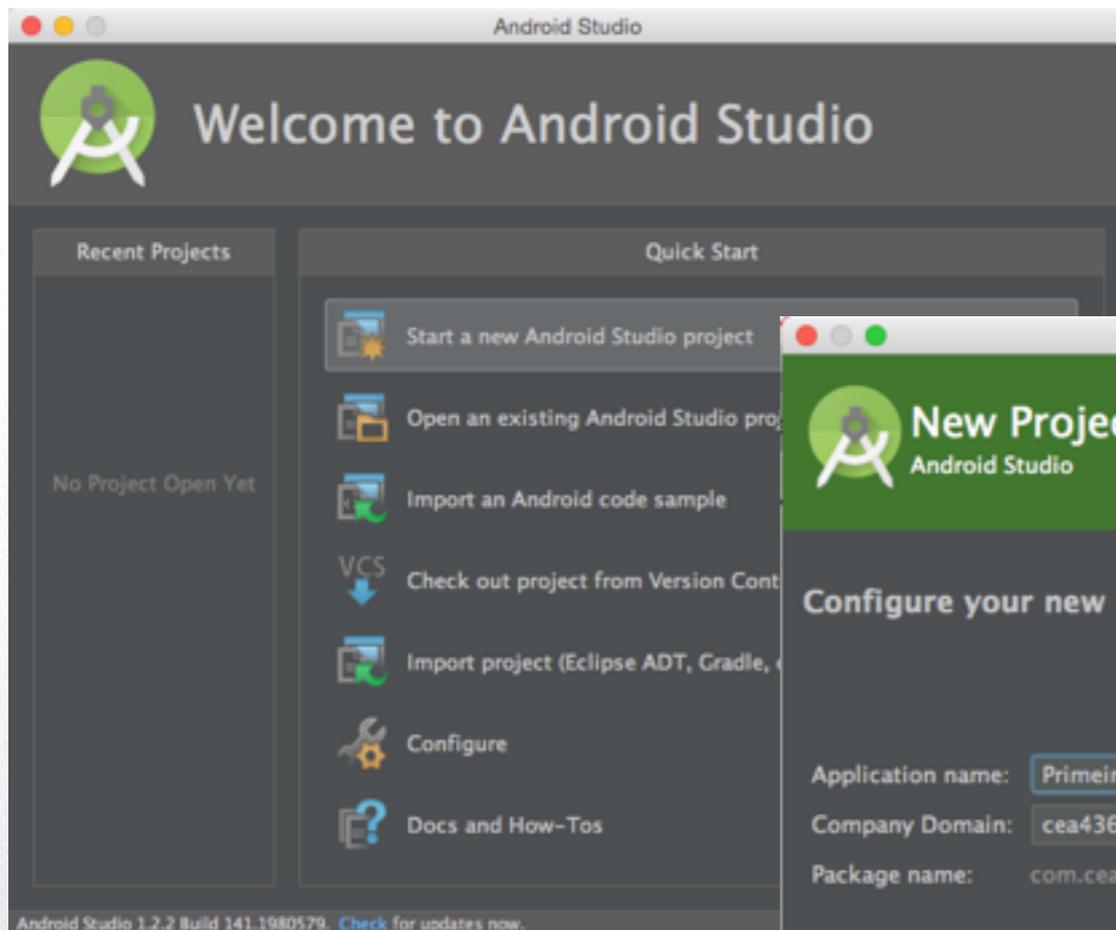
---

- Criar um novo projeto/app
  - ✓ “Start a New Android Studio project”
  - ✓ Preencher:
    - .Application Name: PrimeiraApp
    - . Company Name: cea436.com
    - .Ajustar local de armazenamento do projeto.
  - ✓ Clicar em “Next”



# Primeira Aplicação **Android**

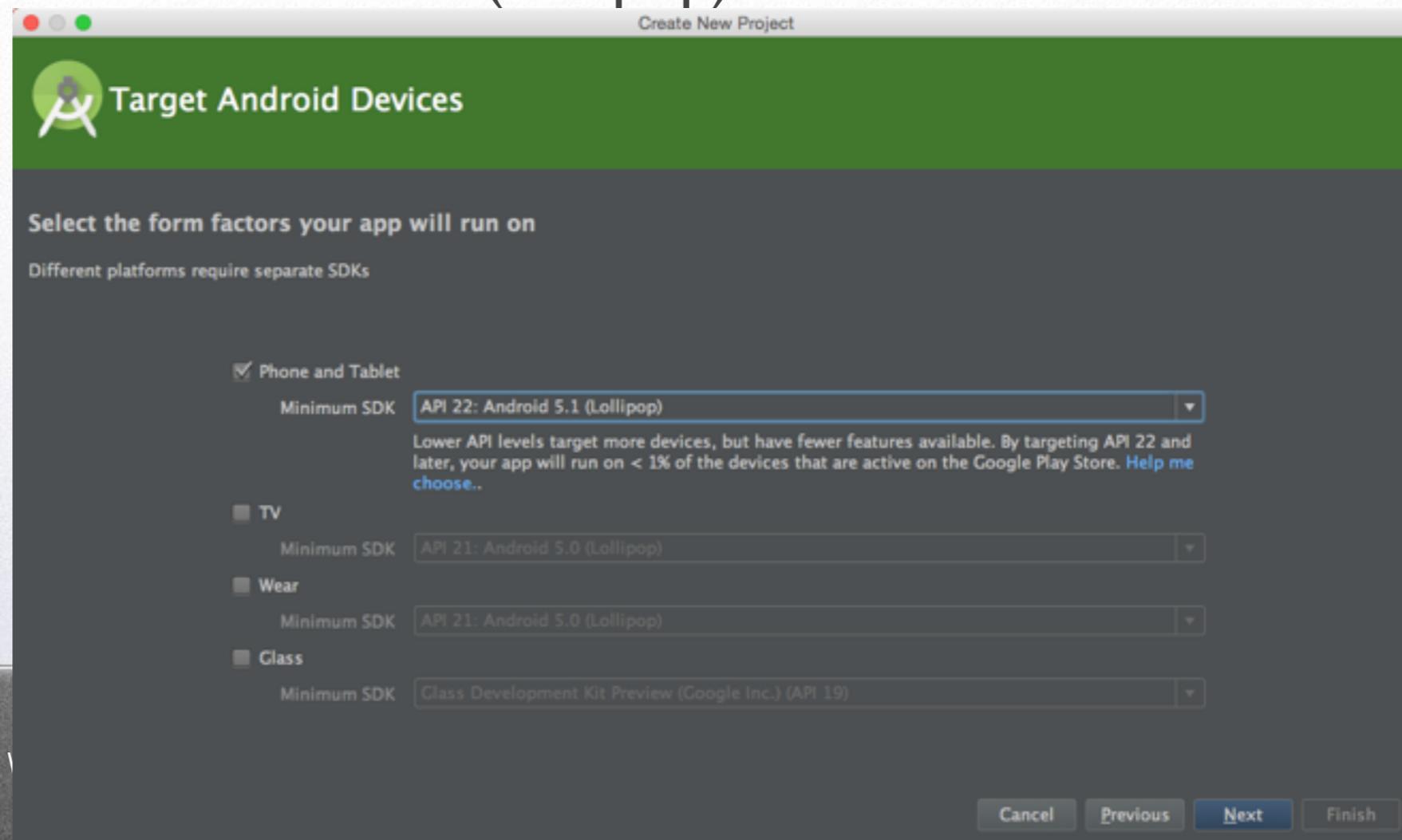
- Criar um novo projeto/app





# Primeira Aplicação **Android**

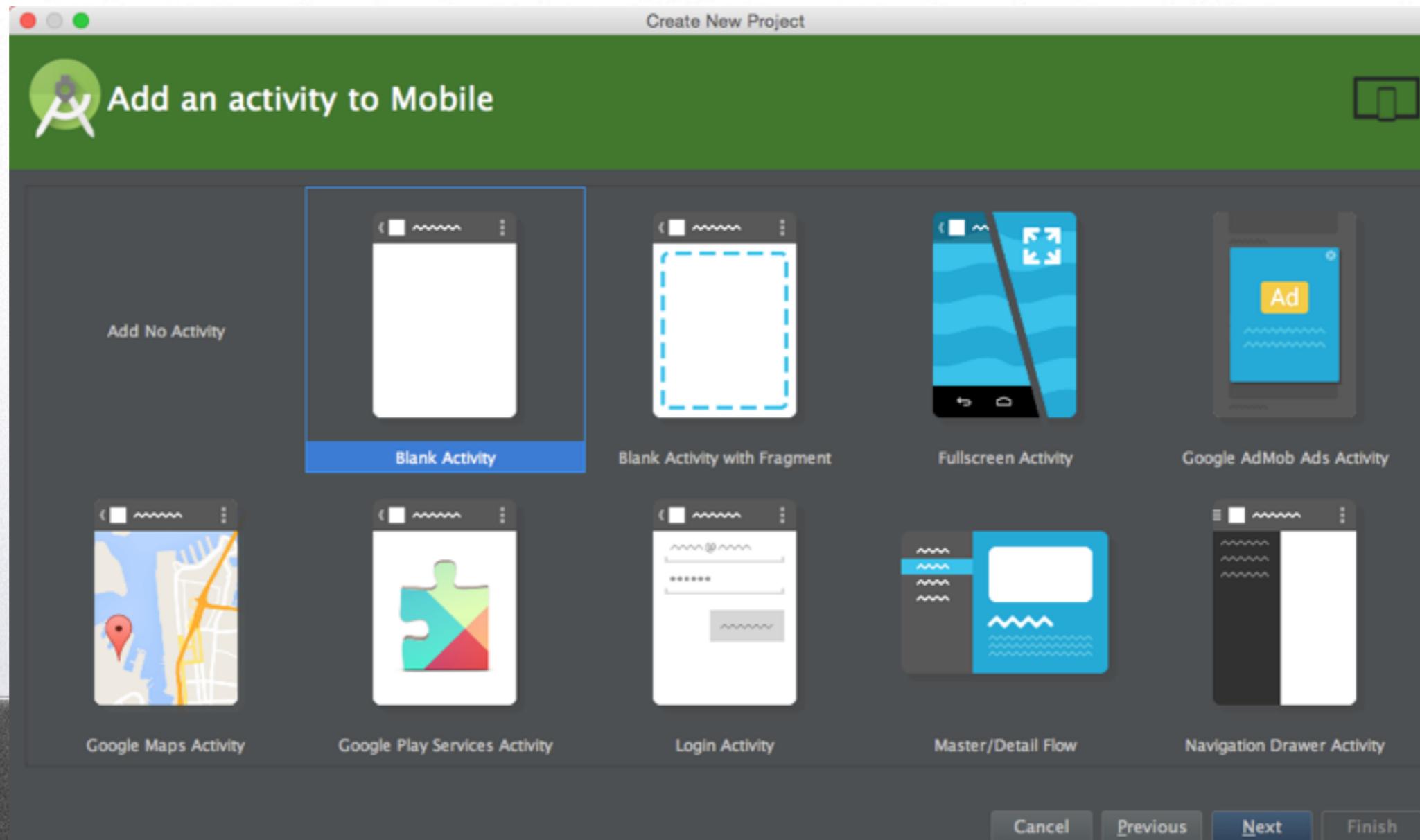
- Criar um novo projeto/app
  - ✓ Ajustar:
    - . Marcar somente opção “Phone and Tablet”
    - . Minimum SDK: API 22: Android 5.1 (Lollipop)
  - ✓ Clique em “Next”





# Primeira Aplicação **Android**

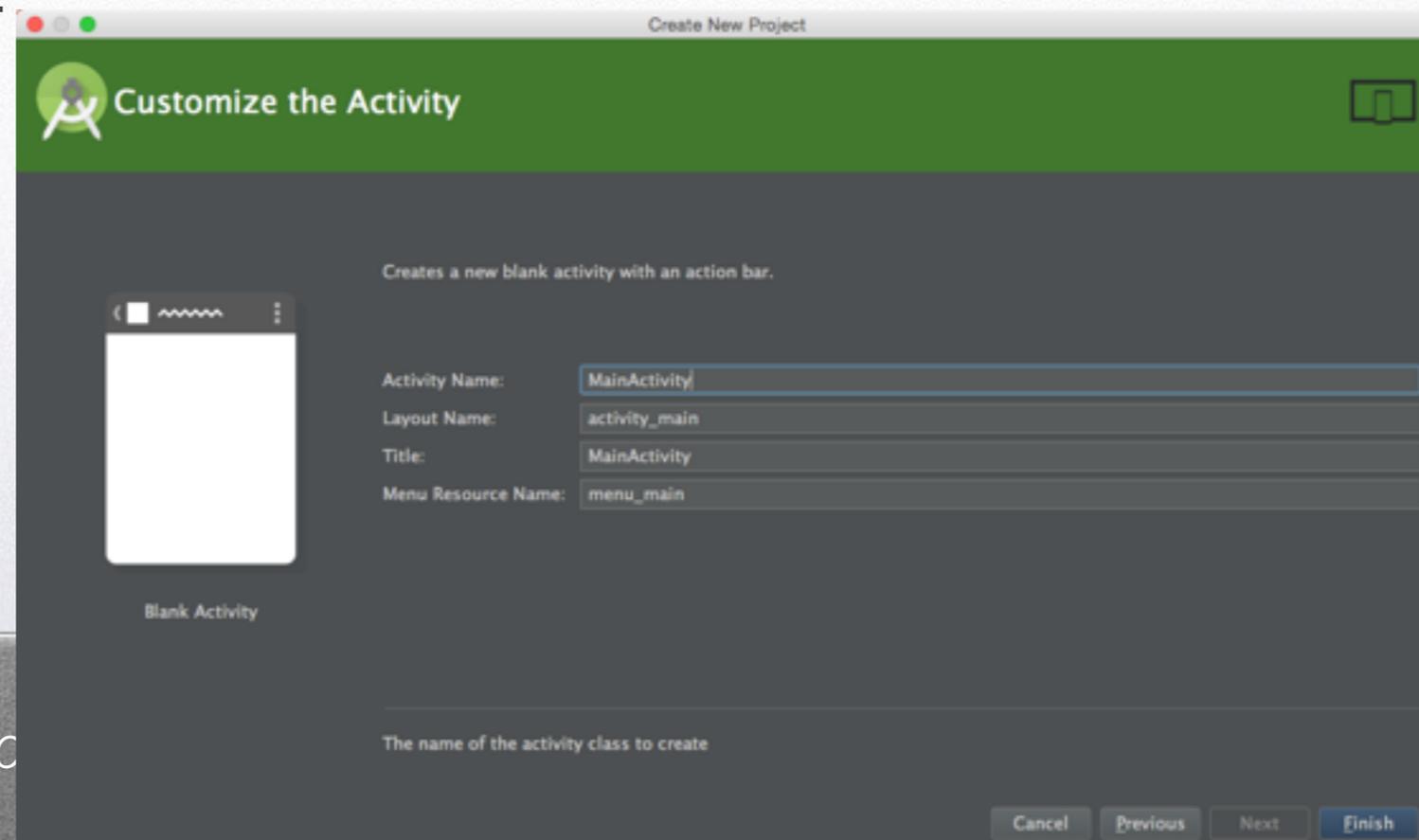
- Criar um novo projeto/app
  - ✓ Selecione “Blank Activity”.
  - ✓ Clique em “Next”





# Primeira Aplicação **Android**

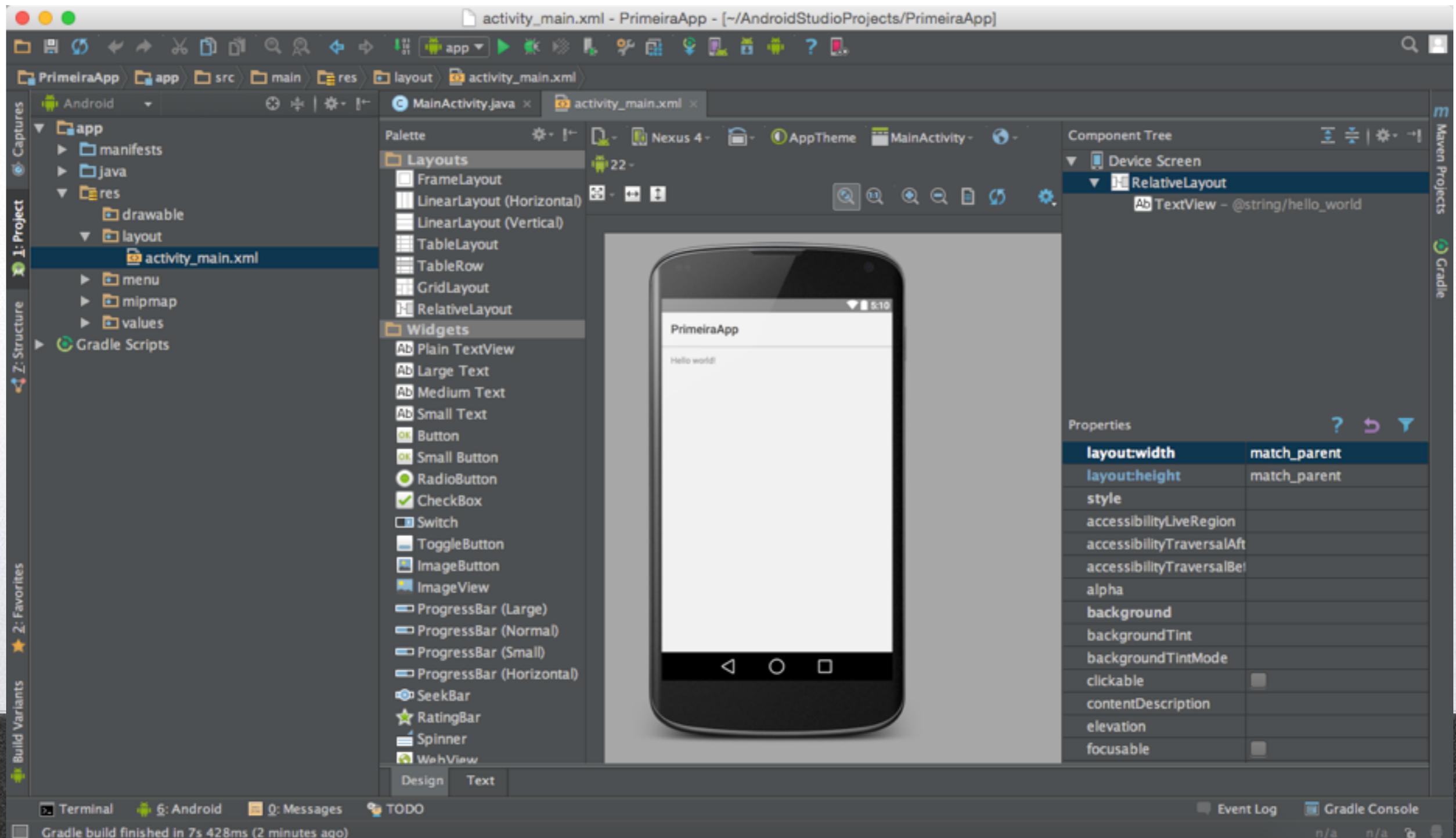
- Criar um novo projeto/app
  - ✓ Definir:
    - .Activity Name: MainActivity
    - .Layout Name: activity\_main
    - .Title: MainActivity
    - .Menu Res. Name: menu\_main
  - ✓ Clique em “Finish”
  - ✓ Aguardar até que projeto seja criado no Android Studio.





# Primeira Aplicação **Android**

- Criar um novo projeto/app





## Primeira Aplicação **Android**

---

- Criar um novo projeto/app
  - ✓ Aplicação foi criada.
  - ✓ Execução ainda não irá funcionar pois não existem emuladores criados.
  - ✓ Aplicativo criado é equivalente ao “Hello World!” em outras linguagens de programação.
  - ✓ Veremos agora a estrutura inicial do projeto de um aplicativo **Android**.



# Criação de um Emulador **Android (AVD)**

---



## Criação de um Emulador **Android** (AVD)

---

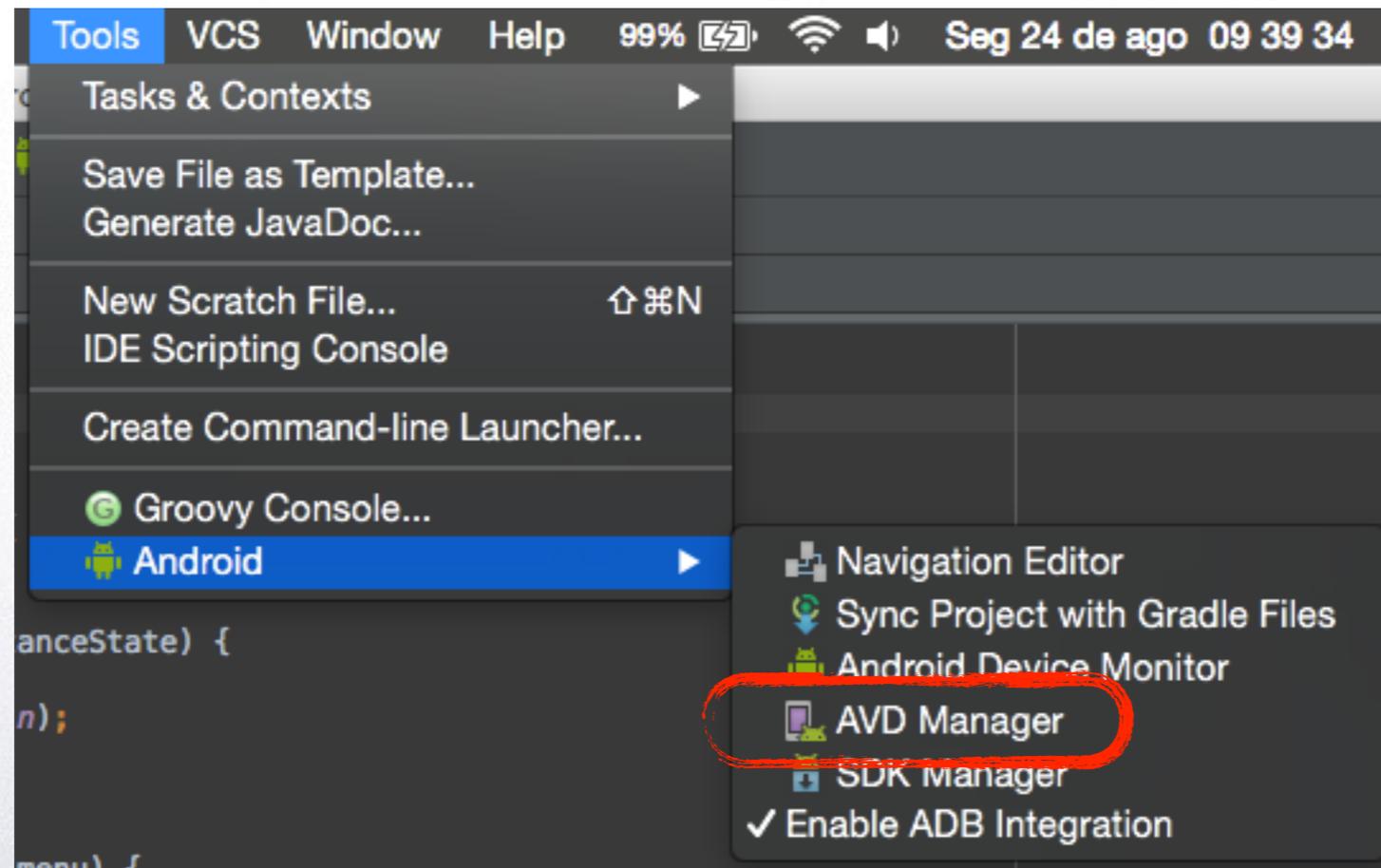
- Emulador **Android** é conhecido como AVD (*Android Virtual Device*).
- Nada mais é que uma representação virtual de um dispositivo **Android** físico (*smartphone, tablet, relógio, etc*).
- Simula o dispositivo físico com mesmas configurações de *hardware* e *software*.
- Vejamos como pode ocorrer a criação de um novo emulador.



# Criação de um Emulador **Android (AVD)**

## - Criar novo AVD

✓ Abrir o AVD Manager: Tools -> Android -> AVD Manager.

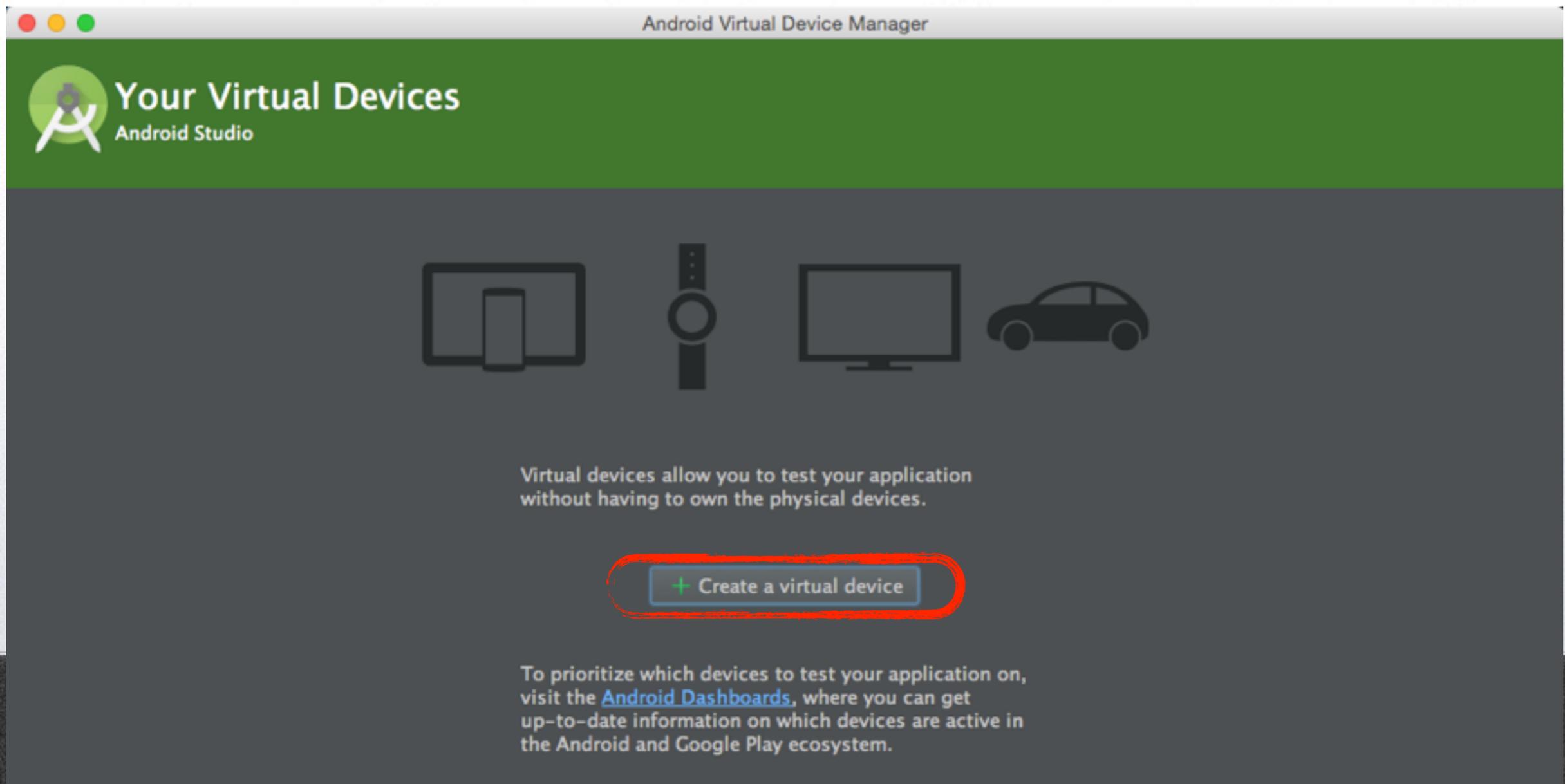




# Criação de um Emulador **Android (AVD)**

## - Criar novo AVD

✓ Como não existem AVDs criados, a listagem estará vazia.  
Clique no botão “*Create a virtual device*”.





## Criação de um Emulador **Android (AVD)**

---

### - Criar novo AVD

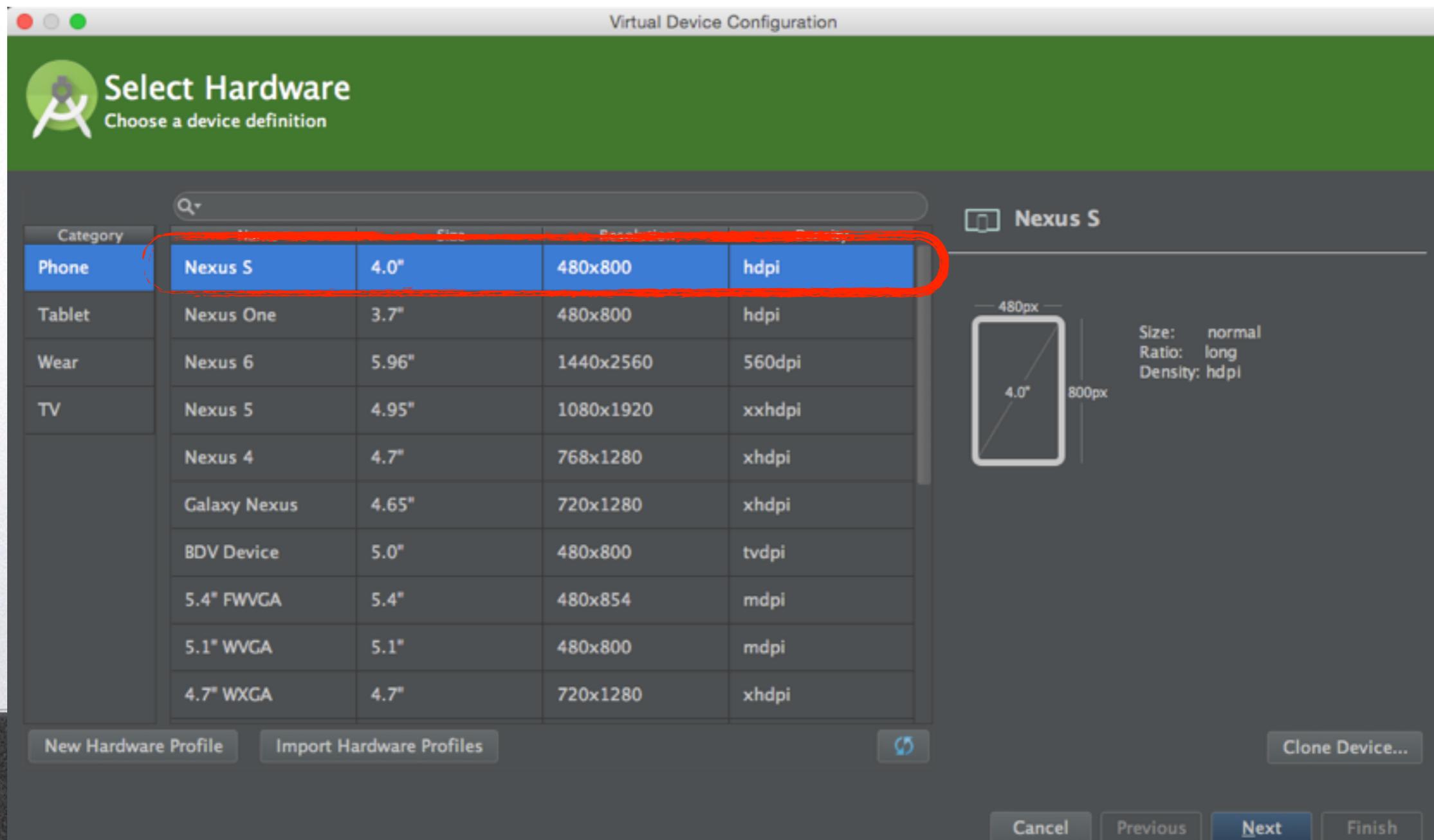
- ✓ Inicialmente é necessário escolher o tipo de dispositivo (*phone, tablet, wear ou TV*).
- ✓ Dentro das categorias já existem diversos dispositivos criados - com diversos tamanhos de telas.
- ✓ Existe uma opção de criar um *hardware* totalmente customizado por você através do botão “**New Hardware Profile**”.
- ✓ **Lembre-se:** Quanto melhor a configuração, mais processamento o AVD irá requerer.



# Criação de um Emulador **Android (AVD)**

- Criar novo AVD

✓ Aqui selecionei “Nexus S” (4” - 480 x 800).





# Criação de um Emulador **Android (AVD)**

- Criar novo AVD

✓ Selecione agora a versão desejada do **Android**:

Virtual Device Configuration

**System Image**  
Select a system image

Release Name	API Level	ABI	Target
Lollipop	22	armeabi-v7a	Android 5.1.1
<b>Lollipop</b>	<b>22</b>	<b>x86</b>	<b>Android 5.1.1</b>
Lollipop	21	x86	Google APIs (Google Inc.) - google_ap
KitKat	19	armeabi-v7a	Android 4.4.2
KitKat	19	armeabi-v7a	Google APIs (Google Inc.)
Jelly Bean	18	armeabi-v7a	Android 4.3.1
Jelly Bean	18	armeabi-v7a	Google APIs (Google Inc.)

**Lollipop**



API Level  
**22**

Android  
**5.1**

**Android Open Source Project**

System Image  
**x86**

Questions on API level?  
See the [API level distribution chart](#)

Show downloadable system images

Refreshing...

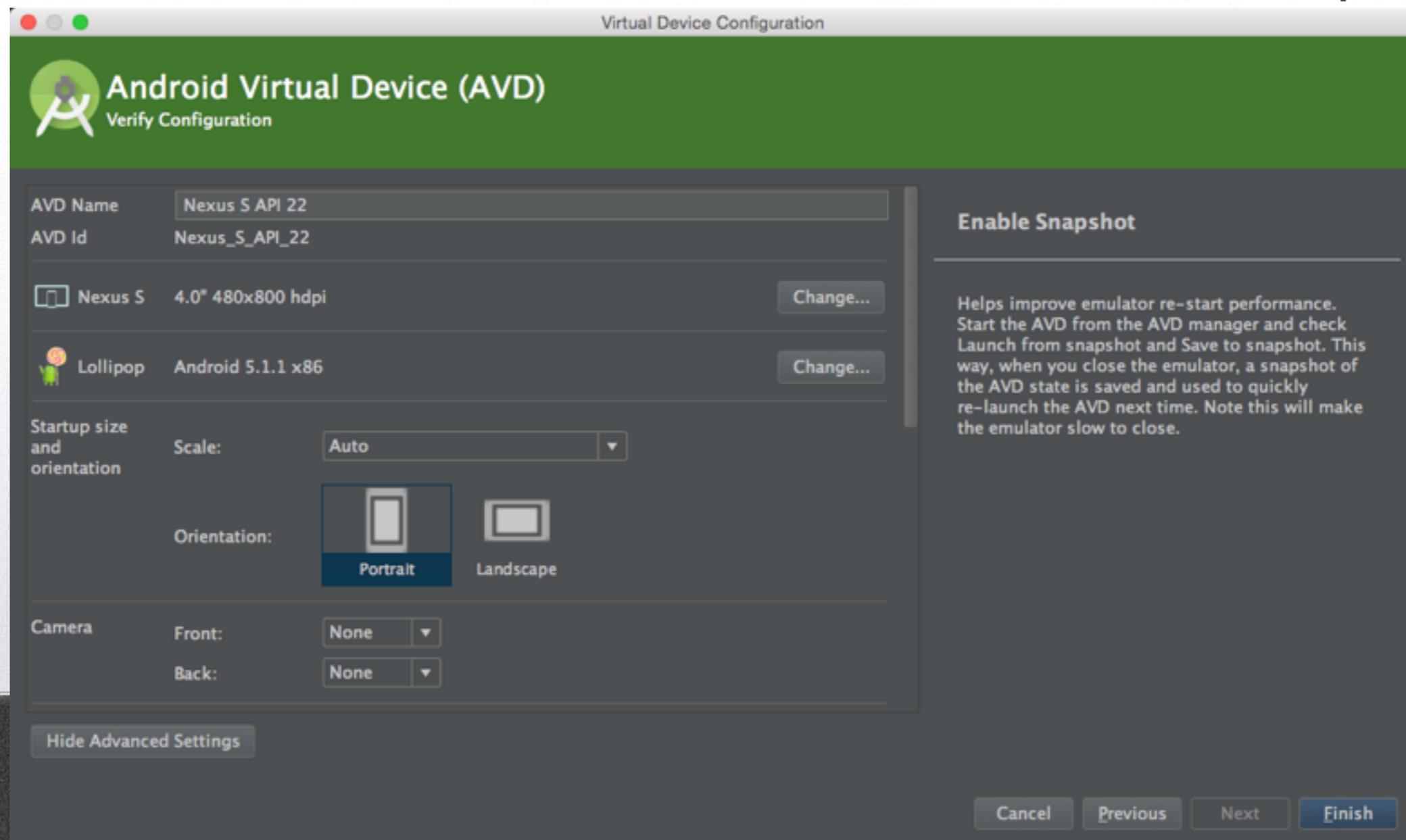
Cancel Previous **Next** Finish



# Criação de um Emulador **Android (AVD)**

## - Criar novo AVD

✓ Opções de diversas configurações de *hardware* são apresentadas. Recomendação: Aumentar memória RAM disp.

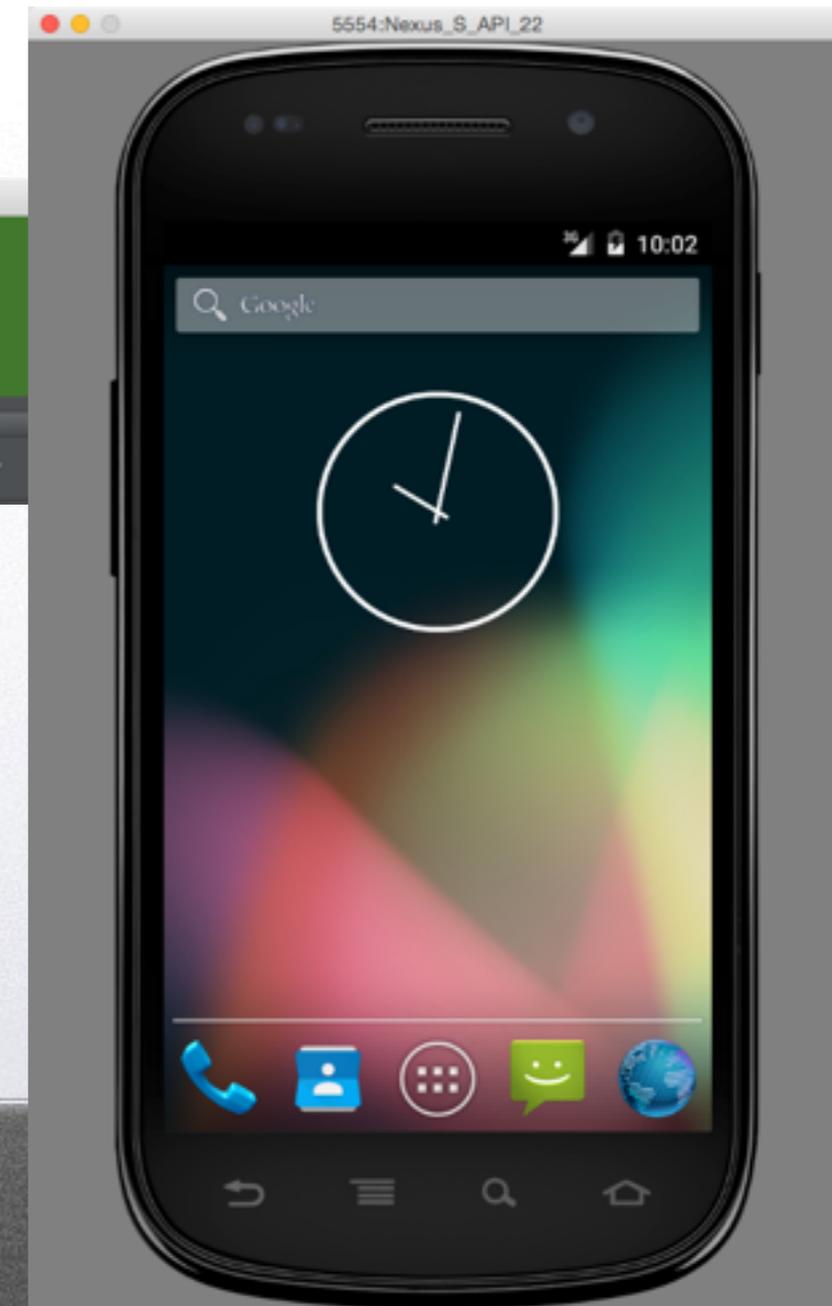
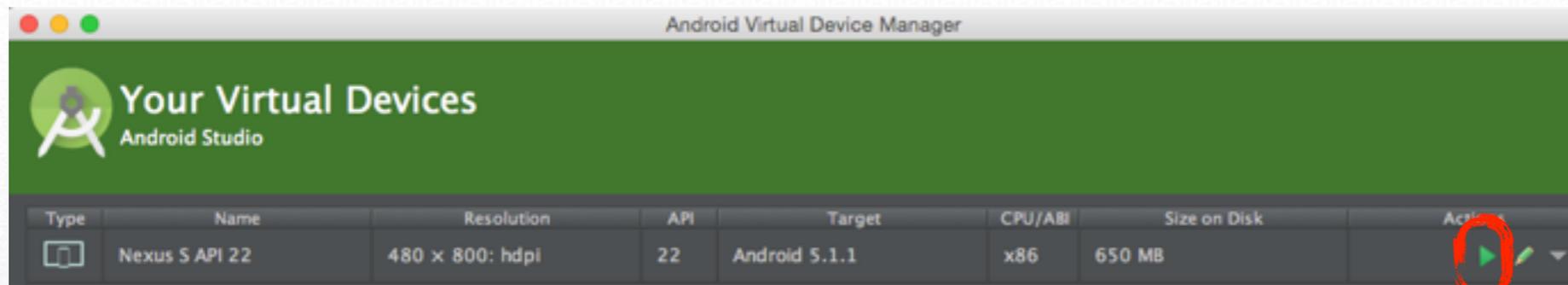




# Criação de um Emulador **Android (AVD)**

## - Criar novo AVD

- ✓ Teste: Na aba “Actions”, clique no ícone de play (seta verde).
- ✓ Verificar correta execução do AVD.





# Execução de Aplicativo no **AVD**

---

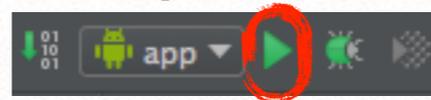


## Execução de Aplicativo no AVD

### - Executar aplicativo criado

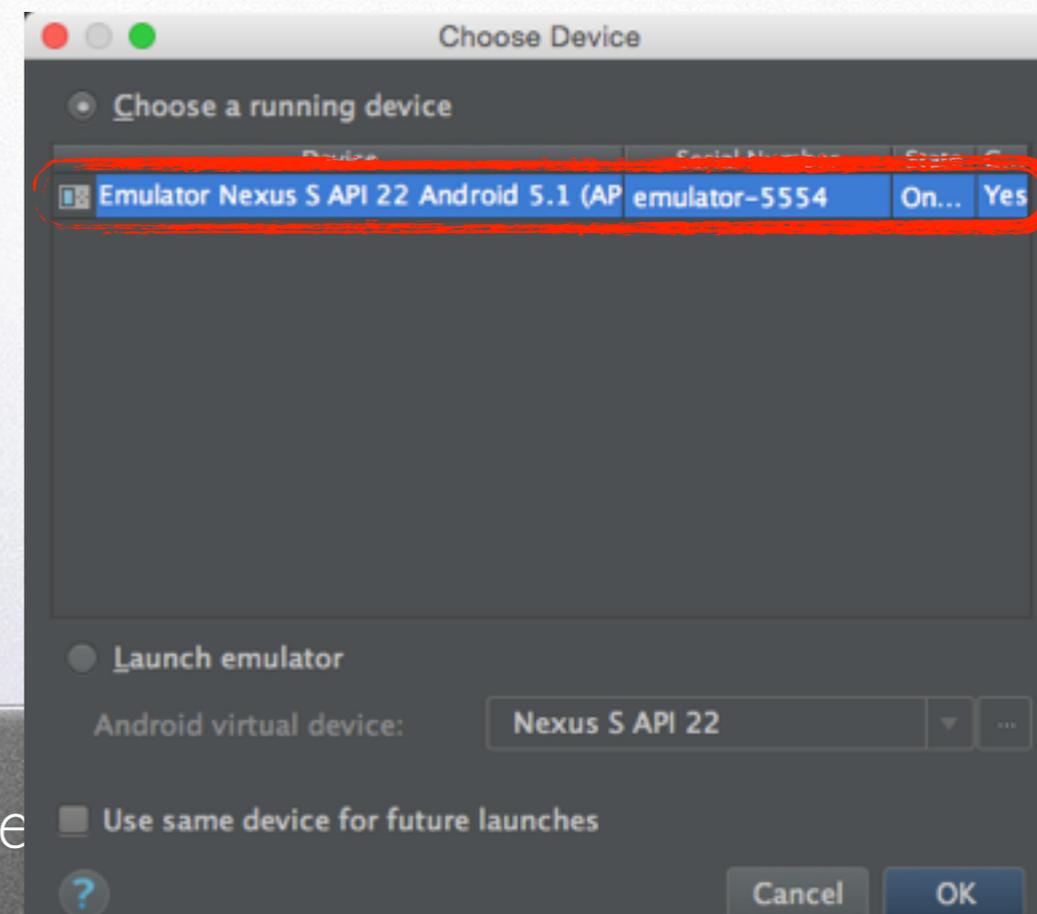
✓ Volte na aplicação criada anteriormente.

✓ Clique no botão “Run App” na parte superior da janela principal do Android Studio.



✓ Uma janela irá aparecer perguntando sobre qual AVD utilizar.

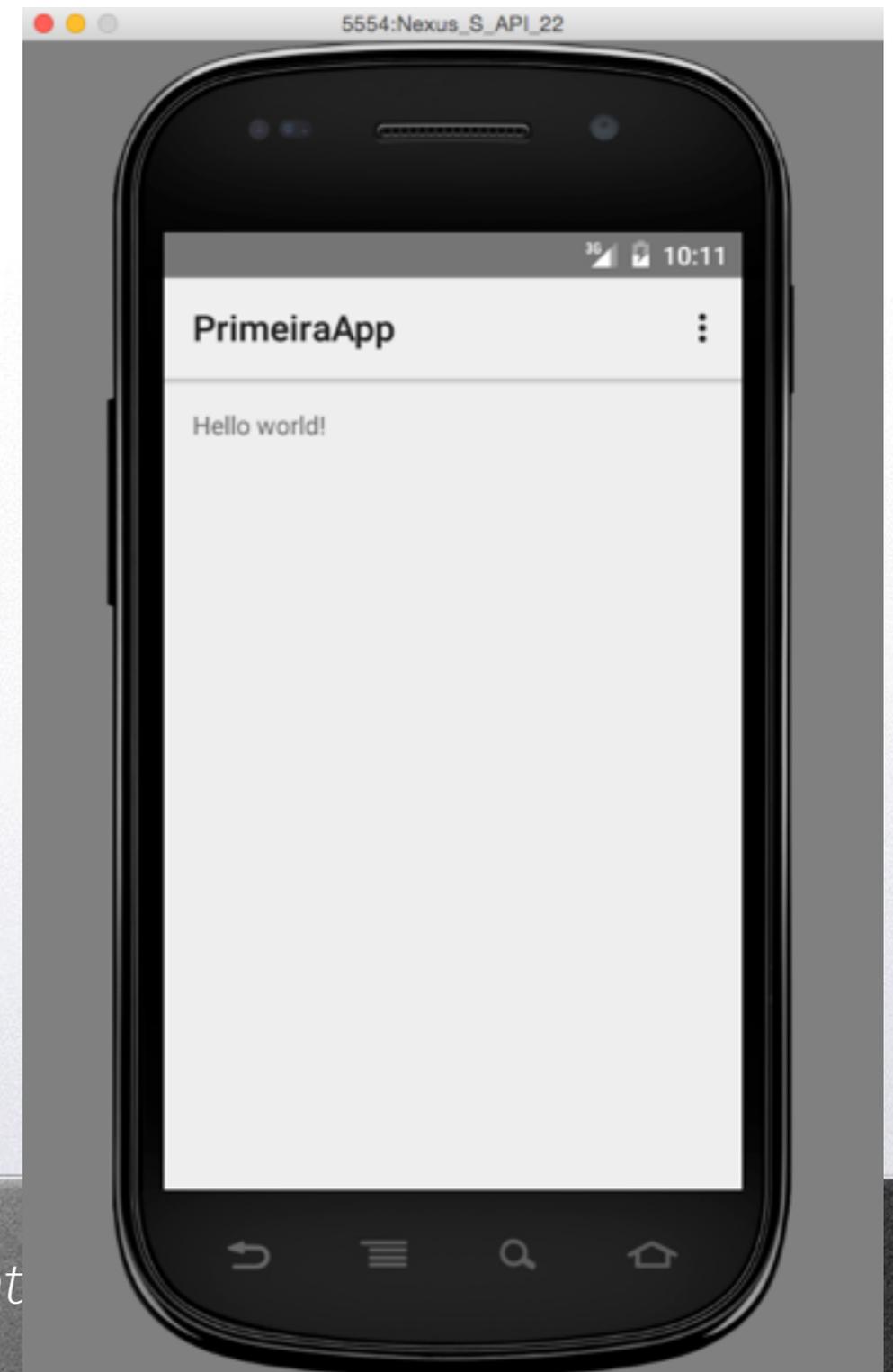
✓ Caso possua mais de um AVD, deixe marcado o que criamos anteriormente e clique em OK.





## Execução de Aplicativo no AVD

- Executar aplicativo criado
  - ✓ Confira o resultado na janela do AVD:





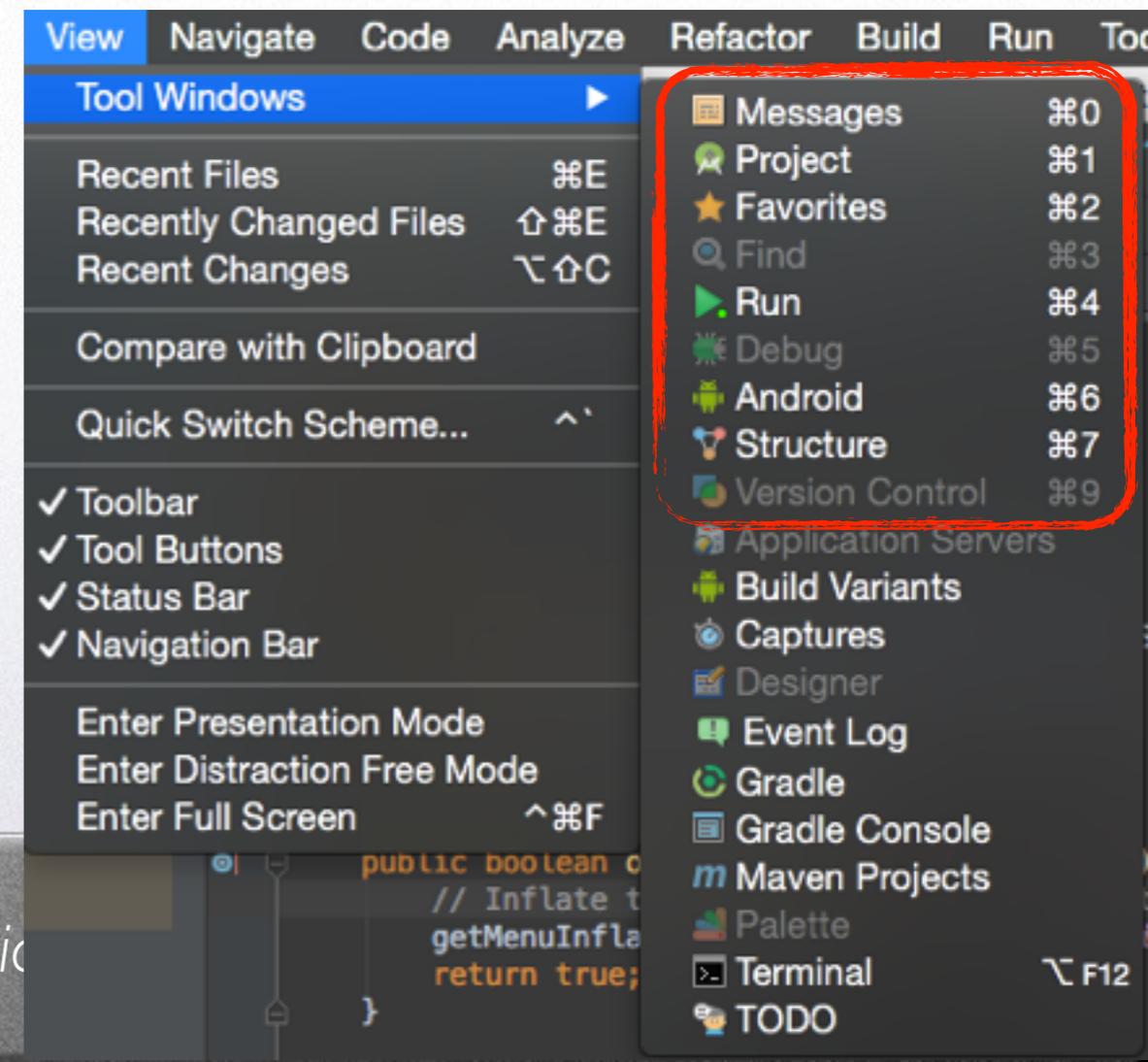
# Principais Janelas do **Android Studio**

---



# Principais Janelas do **Android Studio**

- Executar aplicativo criado
  - ✓ Trabalhar com um ambiente requer que conheçamos todas as suas opções.
  - ✓ Algumas das principais janelas do **Android Studio**:
    - . Mensagens;
    - . Projeto;
    - . Execução;
    - . Debug;
    - . etc.





# *Android* Debug Bridge (ADB)

---



## Android Debug Bridge

---

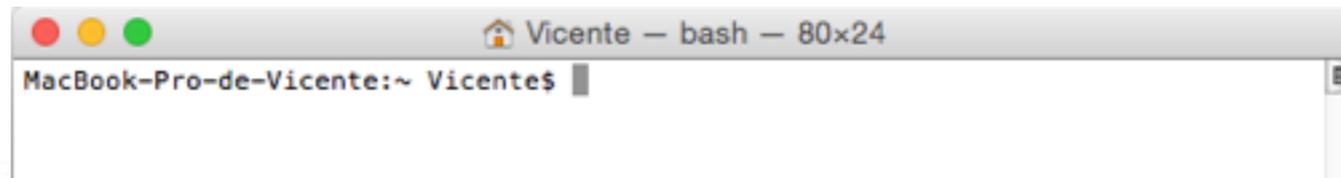
- Permite gerenciar e controlar o emulador ou qualquer dispositivo físico **Android**.
- Possui interface textual.
- É uma aplicação cliente-servidor que executa durante todo tempo. O servidor é executado na máquina e o cliente no AVD/dispositivo físico.
- Comandos aceitos são os mesmos do SO Linux (uma vez que esse é a base do **Android**).



## Android Debug Bridge

### - Exemplos:

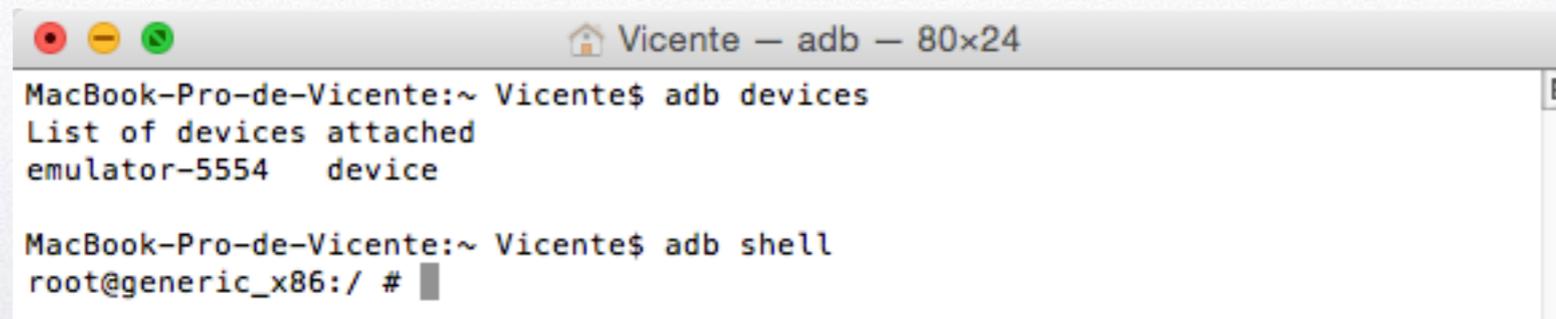
✓ Abrir um terminal no computador local (Cmd no windows / shell ou terminal no Linux).



```
MacBook-Pro-de-Vicente:~ Vicente$
```

✓ Digitar: `$ adb devices <enter>`

✓ Digitar: `$ adb shell <enter>`



```
MacBook-Pro-de-Vicente:~ Vicente$ adb devices
List of devices attached
emulator-5554    device

MacBook-Pro-de-Vicente:~ Vicente$ adb shell
root@generic_x86:/ #
```

✓ Digitar qualquer comando shell Linux. Exemplo: `$ ls`



## Android Debug Bridge

### - Exemplos:

✓ ADB também pode ser utilizado para monitorar e depurar aplicativos desenvolvidos em **Android** através do **LogCat**.

✓ **LogCat** é a interface de depuração do **Android**.

✓ Interface pode ser acessada através da janela 6:Android:View -> Tool Windows -> Android.

```
08-24 10:10:44.742 2194-2210/com.cea436.primeiraapp W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xb431bc40, error=EGL_SUCCESS
08-24 10:32:57.850 2194-2210/com.cea436.primeiraapp W/EGL_emulation: eglSurfaceAttrib not implemented
08-24 10:32:57.850 2194-2210/com.cea436.primeiraapp W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xb431bc40, error=EGL_SUCCESS
08-24 10:34:07.753 2194-2210/com.cea436.primeiraapp W/EGL_emulation: eglSurfaceAttrib not implemented
08-24 10:34:07.753 2194-2210/com.cea436.primeiraapp W/OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xb431bc40, error=EGL_SUCCESS
08-24 10:45:19.960 2194-2196/com.cea436.primeiraapp I/art: Debugger is no longer active
```





## Android Debug Bridge

---

### - Exemplos:

✓ Através dos comandos `adb push` / `adb pull` é possível copiar arquivos **de** e **para** os dispositivo ou AVD.

✓ ADB e LogCat também funcionam quando se conecta um dispositivo físico via interface USB (não somente com o AVD).

✓ Mais adiante veremos como integrar mensagens de depuração diretamente no código do aplicativo **Android** em desenvolvimento.



## Android Debug Bridge

---

### - Outras opções:

✓ **ADB (Android Debug Bridge)**: Interface de debug do Android.

. \$ adb device : Listagem dos dispositivos conectados no computador.

. \$ adb shell: Acesso à shell do dispositivo.

. \$ adb shell ls: Listagem do sistema dos arquivos presentes na raiz do root filesystem.

. \$ adb shell ls /data/app: Listagem das apps instaladas no sistema.