



INSTALAÇÃO E INTRODUÇÃO

Luiz Carlos Querino Filho

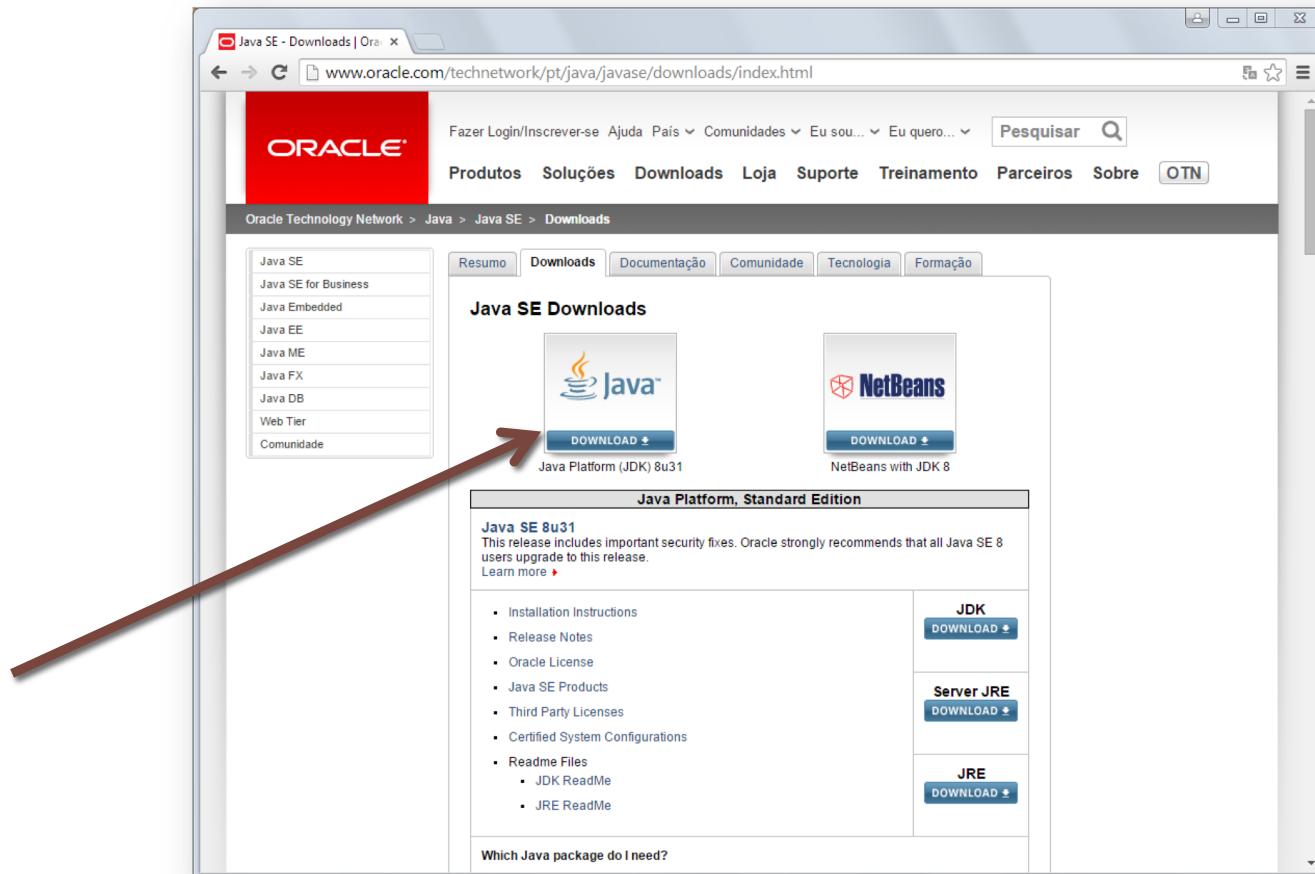
luiz.querino@fatec.sp.gov.br

ETAPAS

1. Baixar e instalar o Java SE Development Kit (JDK)
2. Baixar e instalar o Android Studio
3. Criar passo-a-passo um aplicativo Android para testar a instalação
4. Montar a interface do seu app
5. Escrever o código Java do aplicativo (agora sim!)
6. SE NECESSÁRIO: Criar um dispositivo virtual Android (Android Virtual Device)
7. Testar o aplicativo no emulador

1.1 – Baixe o JDK (Java SE Development Kit) no endereço:

<http://www.oracle.com/technetwork/pt/java/javase/downloads/index.html>



1.2 – Aceite a licença e escolha a versão adequada ao seu S.O.

The screenshot shows a web browser window with the URL www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html?ssSourceSiteId=otnpt. The page title is "Java SE Development Kit 8 Downloads". The main content area displays the "Java SE Development Kit 8u31" download section, which requires accepting the Oracle Binary Code License Agreement. Below the acceptance checkbox, there is a table of download links for various operating systems and architectures. The right sidebar contains links to "Java SDKs and Tools" and "Java Resources". At the bottom right of the page, there is an advertisement for the "ORACLE VIRTUAL TECHNOLOGY SUMMIT" scheduled for February 11th, 25th, and March 4th, with a "REGISTER!" button.

Java SE Development Kit 8u31 Downloads

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	135.24 MB	jdk-8u31-linux-i586.rpm
Linux x86	154.91 MB	jdk-8u31-linux-i586.tar.gz
Linux x64	135.62 MB	jdk-8u31-linux-x64.rpm
Linux x64	153.45 MB	jdk-8u31-linux-x64.tar.gz
Mac OS X x64	209.17 MB	jdk-8u31-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	136.91 MB	jdk-8u31-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	97.11 MB	jdk-8u31-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	137.51 MB	jdk-8u31-solaris-x64.tar.Z
Solaris x64	94.82 MB	jdk-8u31-solaris-x64.tar.gz
Windows x86	157.96 MB	jdk-8u31-windows-i586.exe
Windows x64	170.36 MB	jdk-8u31-windows-x64.exe

Java SE Development Kit 8u31 Demos and Samples Downloads

You must accept the Oracle BSD License to download this software.

Accept License Agreement Decline License Agreement

1.3 – Instale o JDK no seu Sistema Operacional



- Você conhece o esquema: **Next, Next, Next, ..., Finish**

2.1 – Obtenha o Android Studio no site oficial (developer.android.com)

The screenshot shows the official Android Developers website. The top navigation bar includes the 'Developers' logo, a search bar labeled 'Pesquisa', and a 'PLAY CONSOLE' button. On the left, a sidebar menu lists categories: INÍCIO, DESIGN, DEVELOP, DISTRIBUTE, STORIES, and PREVIEW. The main content area features a large graphic of an Android robot standing next to a stylized yellow ring. The text 'Android O' and 'Developer Preview' is prominently displayed. Below this, a description reads: 'A mais recente Developer Preview do Android, que traz melhorias de desempenho e novos recursos para seus aplicativos.' followed by a link 'Saiba mais'. At the bottom of the page, there is a footer bar with three links: 'Obter o Android Studio', 'Procurar exemplo de código', and 'Assistir a histórias'.

INÍCIO

DESIGN DEVELOP DISTRIBUTE

Pesquisa

PLAY CONSOLE

INÍCIO

Android

Wear

TV

Auto

Things

DESIGN

DEVELOP

DISTRIBUTE

STORIES

PREVIEW

Android O
Developer Preview

A mais recente Developer Preview do Android, que traz melhorias de desempenho e novos recursos para seus aplicativos.

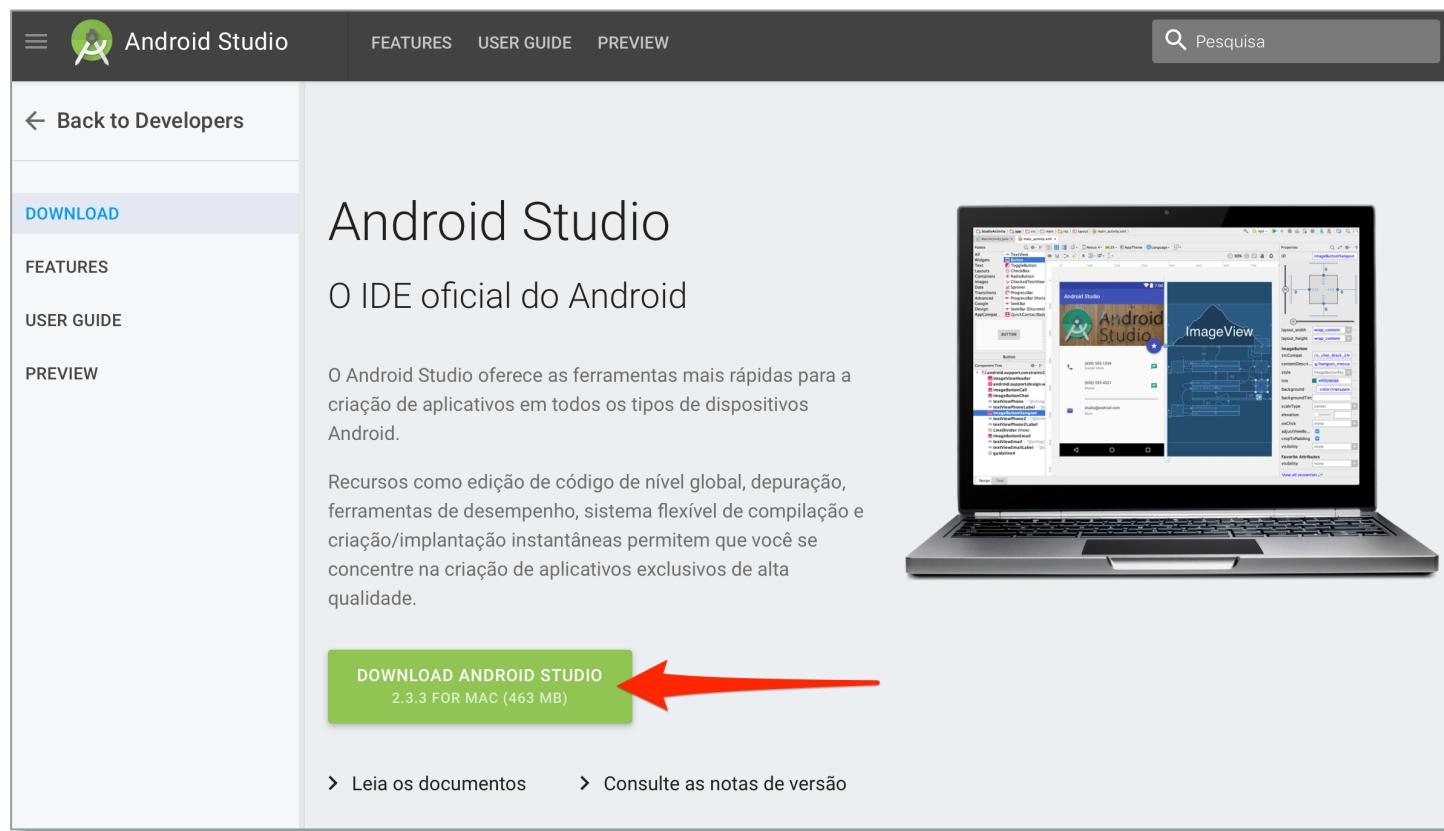
» Saiba mais

Obter o Android Studio

Procurar exemplo de código

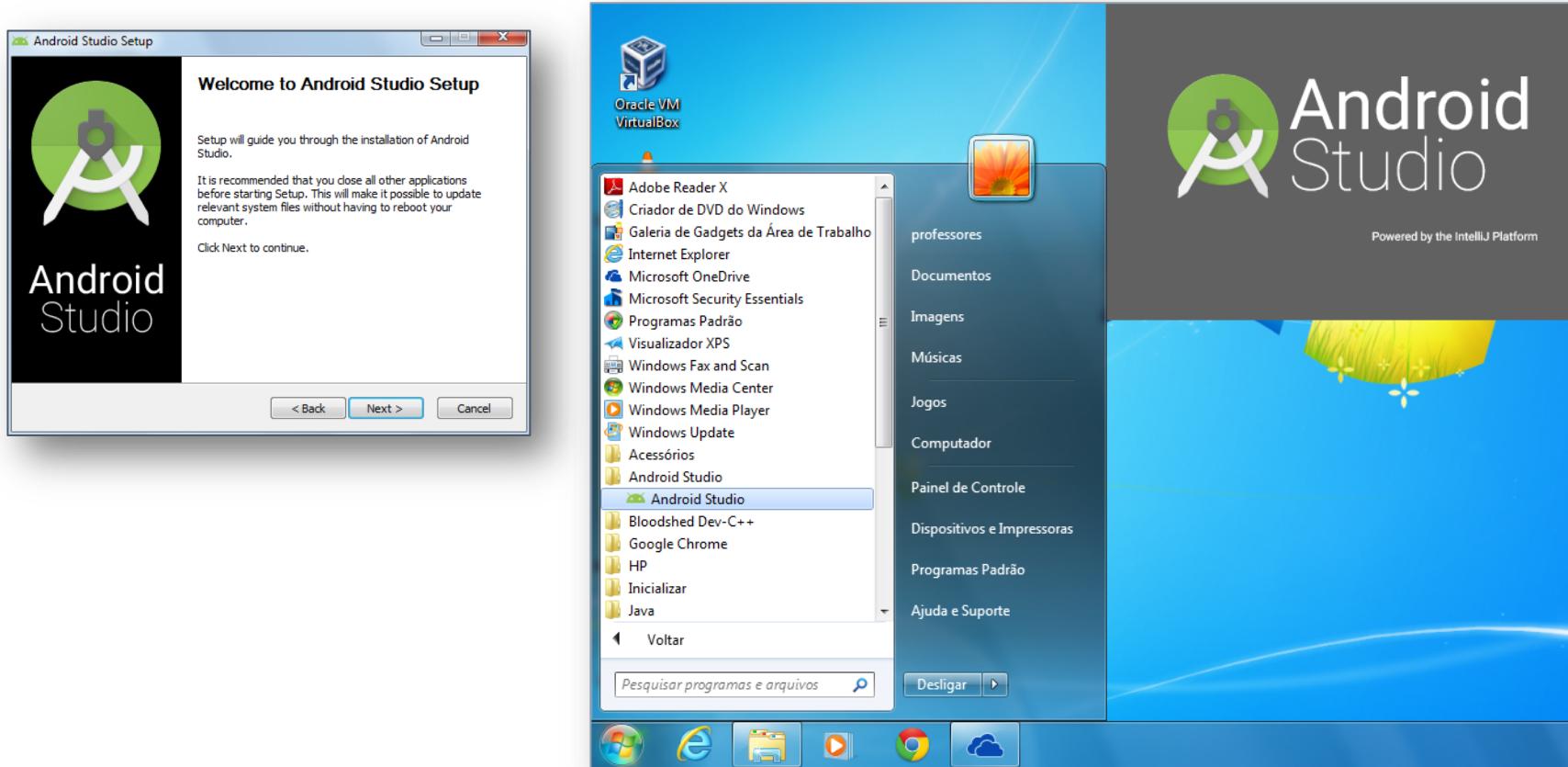
Assistir a histórias

2.2 – Clique no botão para obter a versão do Android Studio para o seu S.O.



- Após clicar no botão, você deverá concordar com a licença do Android Studio antes que o download seja iniciado

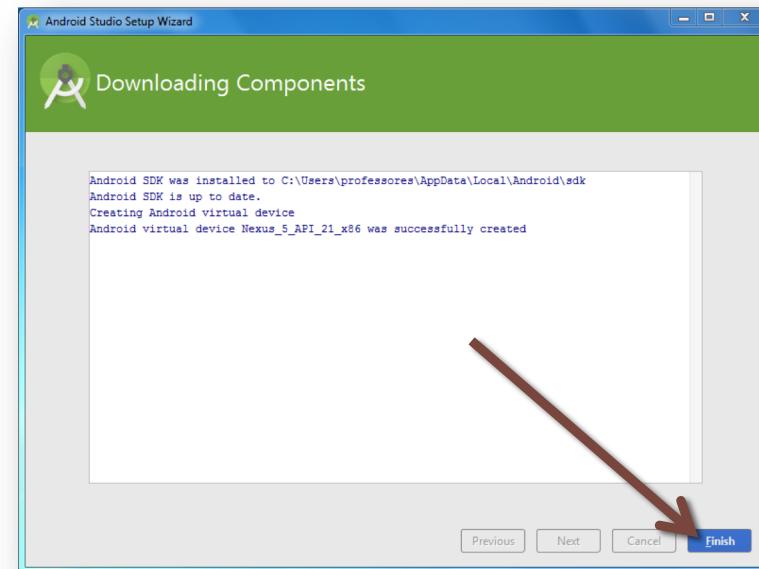
2.3 – Realize a instalação do Android Studio



- Tutorial de Instalação do Android Studio: **Next, Next, ..., I Agree, Next, ..., Install** (ou algo parecido...).

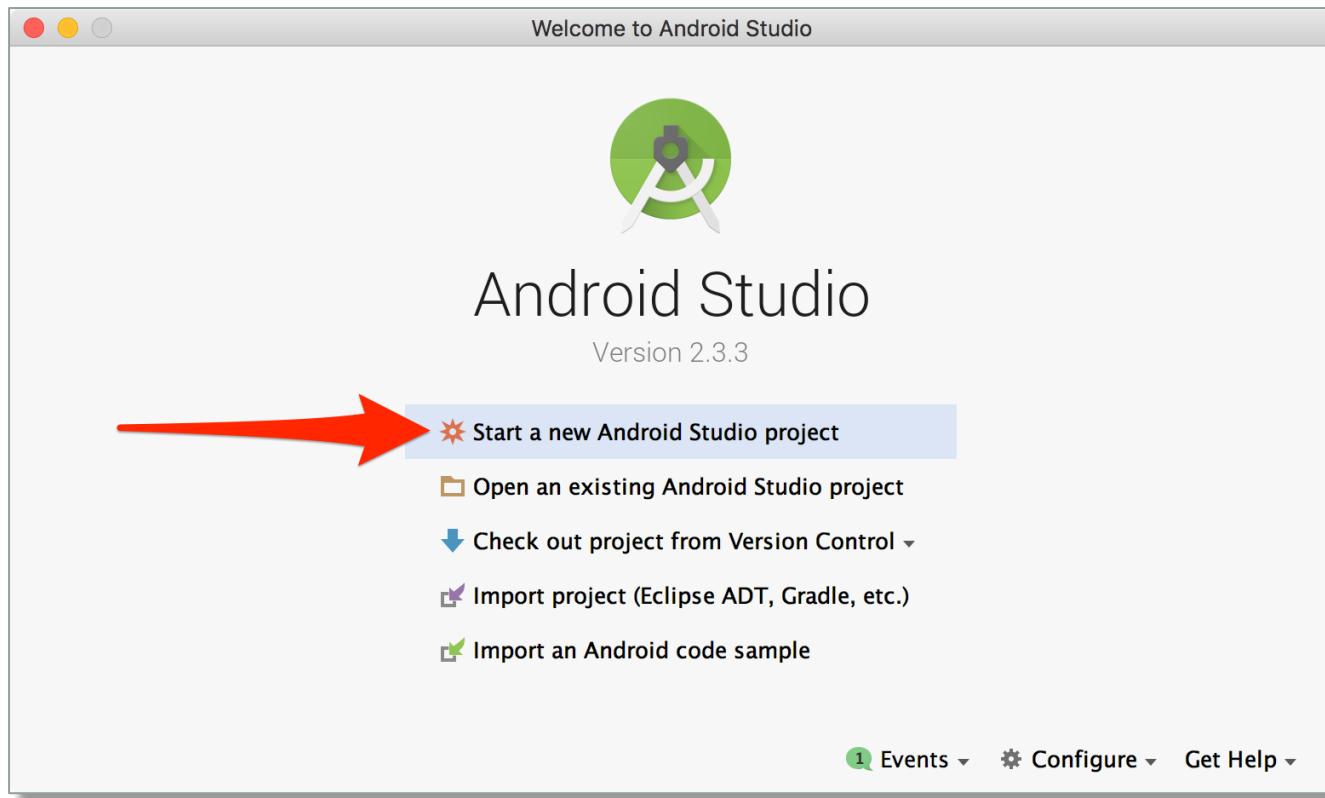
2.4 – Primeira Inicialização do Android Studio

- Na primeira inicialização do aplicativo, serão feitas algumas tarefas básicas, como a criação de um **AVD** padrão.
- Também serão baixados componentes adicionais necessários.
- Um **AVD** (Android Virtual Device) permite que você teste os aplicativos em uma versão emulada do Android.



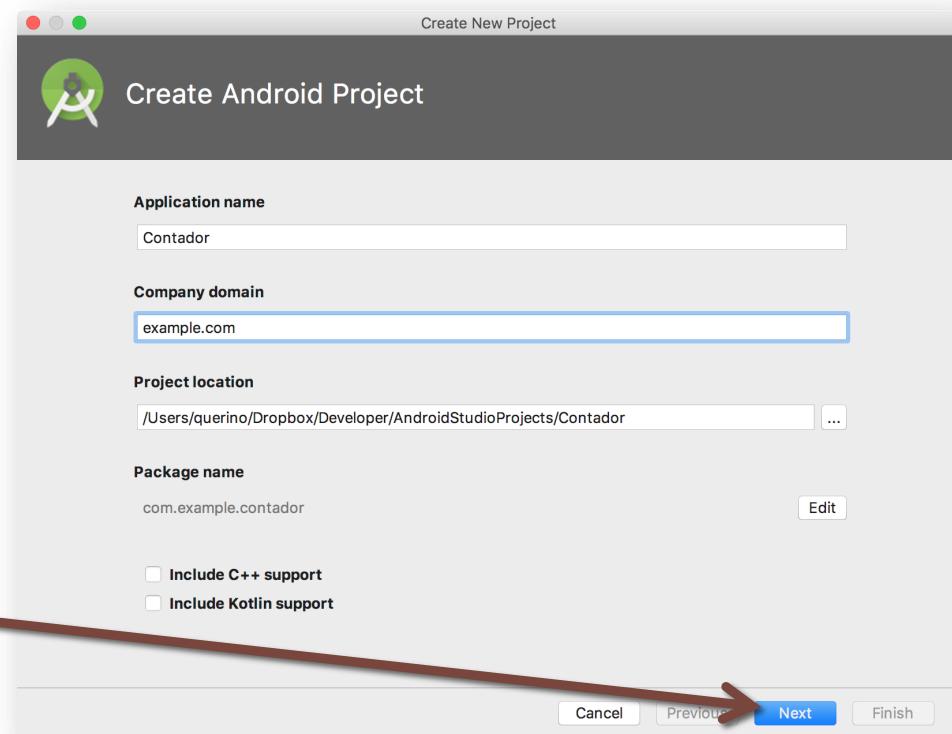
3.1 – Criando seu primeiro aplicativo

- Na tela de abertura do Android Studio, clique em **Start a new Android Studio project** para criar seu projeto:



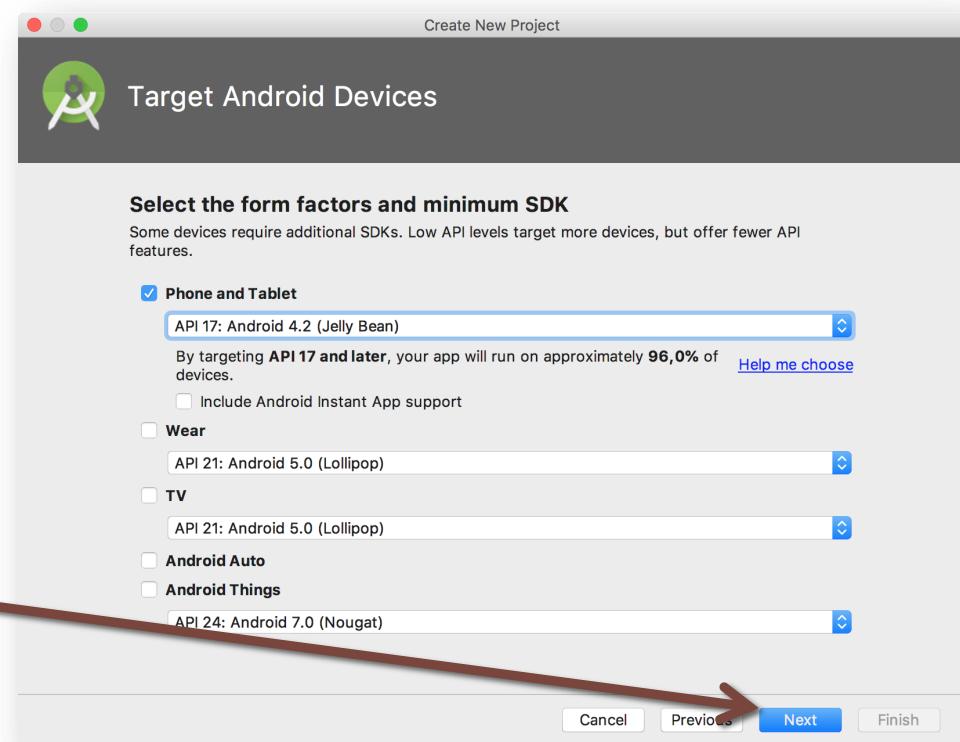
3.2 – Especificando um nome para o projeto

- Vamos criar um aplicativo tipo “Alô Mundo” um pouco (mas bem pouco) mais útil. Ele se chamará **Contador** (especifique seu nome em **Application name**). Clique em **Next**.
- Em **Company Domain**, você pode informar um domínio Internet para usar nos pacotes das classes Java do projeto.
- Clique em **Next**.



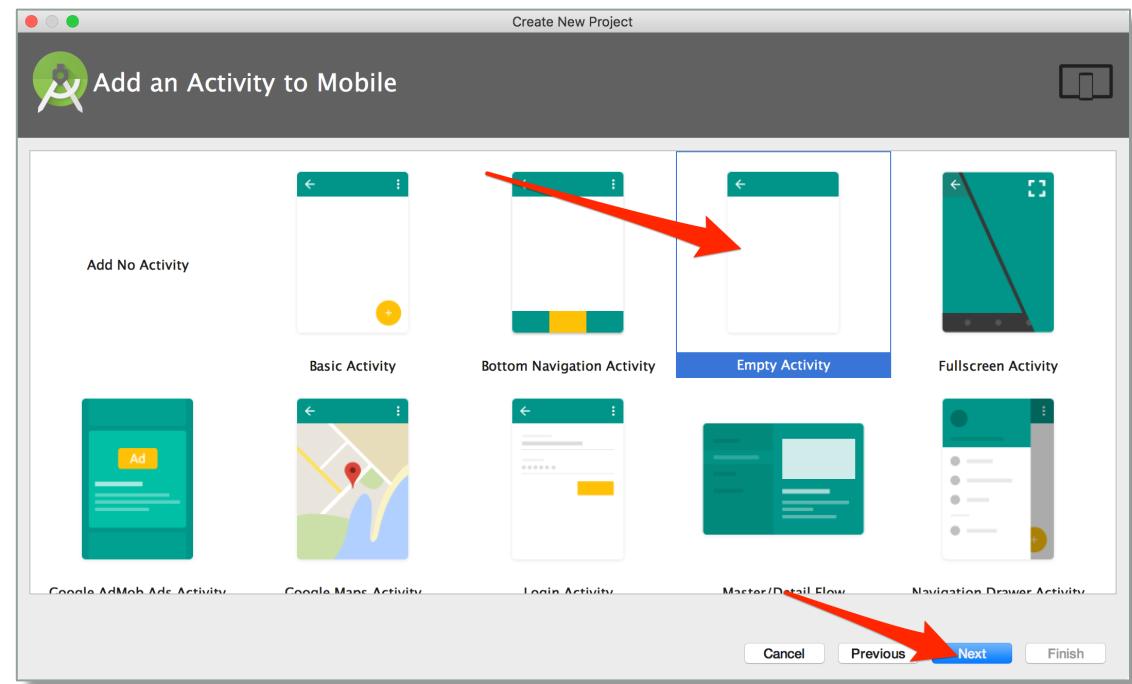
3.3 – Definindo a plataforma e versão alvo do aplicativo

- Aqui devemos determinar qual a plataforma destino do aplicativo. Mantenha selecionado **Phone and Tablet**.
- Em **Minimum SDK**, deve ser determinada a versão mínima do Android necessária para execução do aplicativo.
- Versões muito antigas aumentam a quantidade de usuários aptos a rodar o programa, mas restringe o uso de recursos novos.
- Clique em **Next**.



3.4 – Adicionando a primeira tela do aplicativo

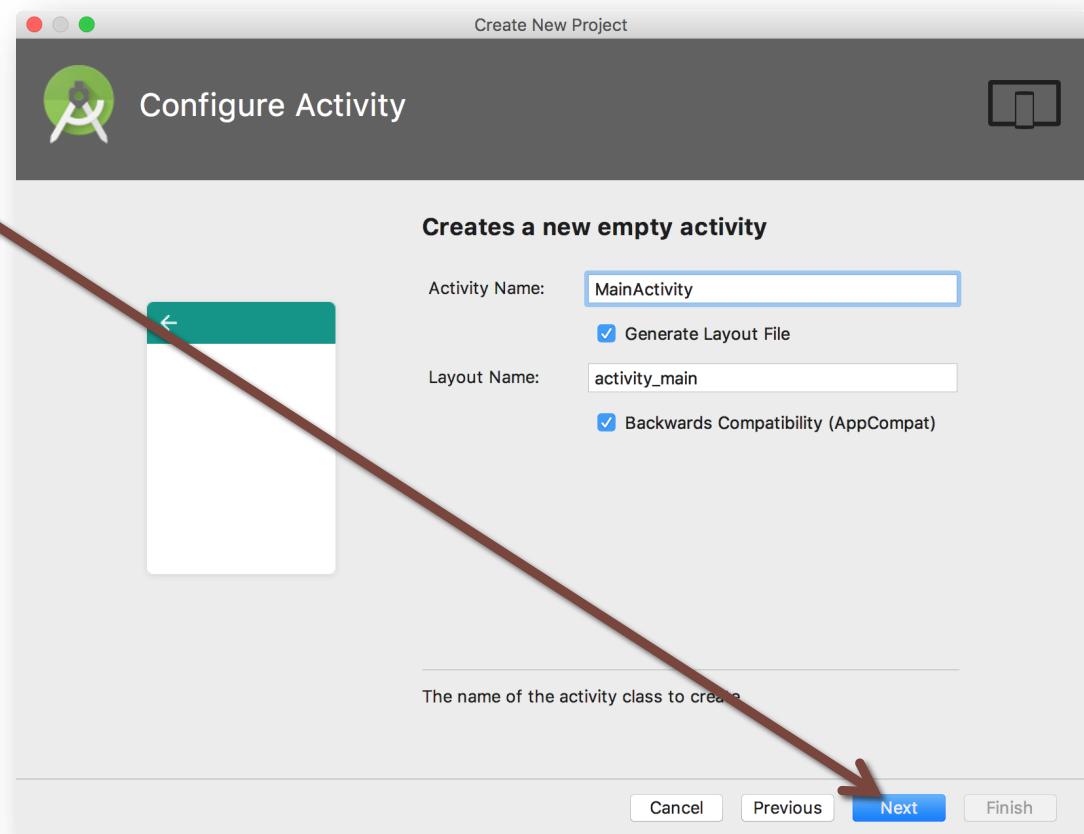
- No Android, uma **Activity** é uma tela do seu aplicativo.
- Aqui, selecionaremos **Empty Activity** para que o projeto inicie com uma tela em branco.
- Clique em **Next**.



3.5 – Última tela: definindo o nome da Activity e do arquivo de layout

- **Activity Name:** o nome da classe Java controladora da tela inicial
- **Layout Name:** nome do arquivo XML com o layout da tela inicial

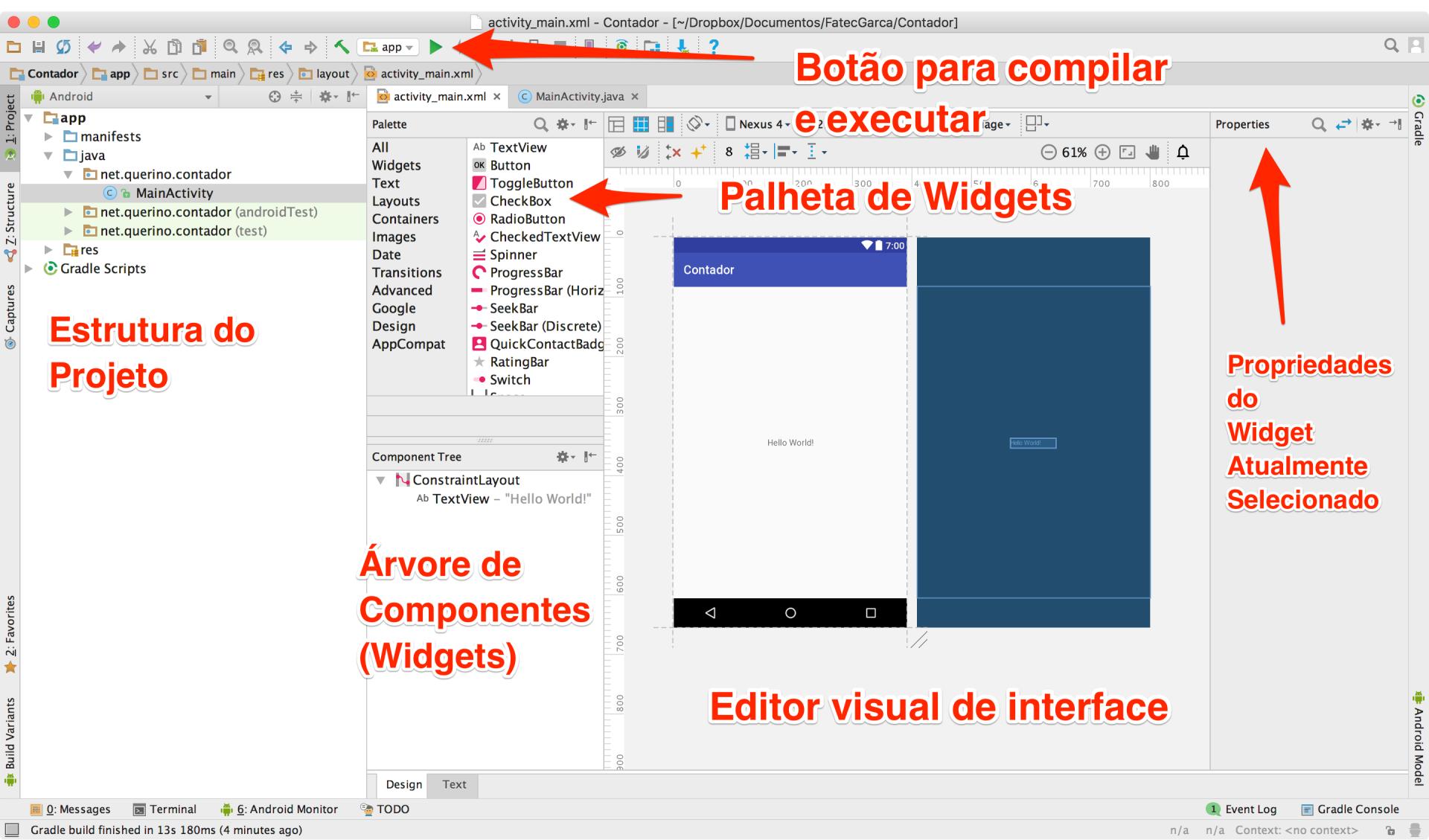
- Clique em **Finish**.



CONCEITO IMPORTANTE

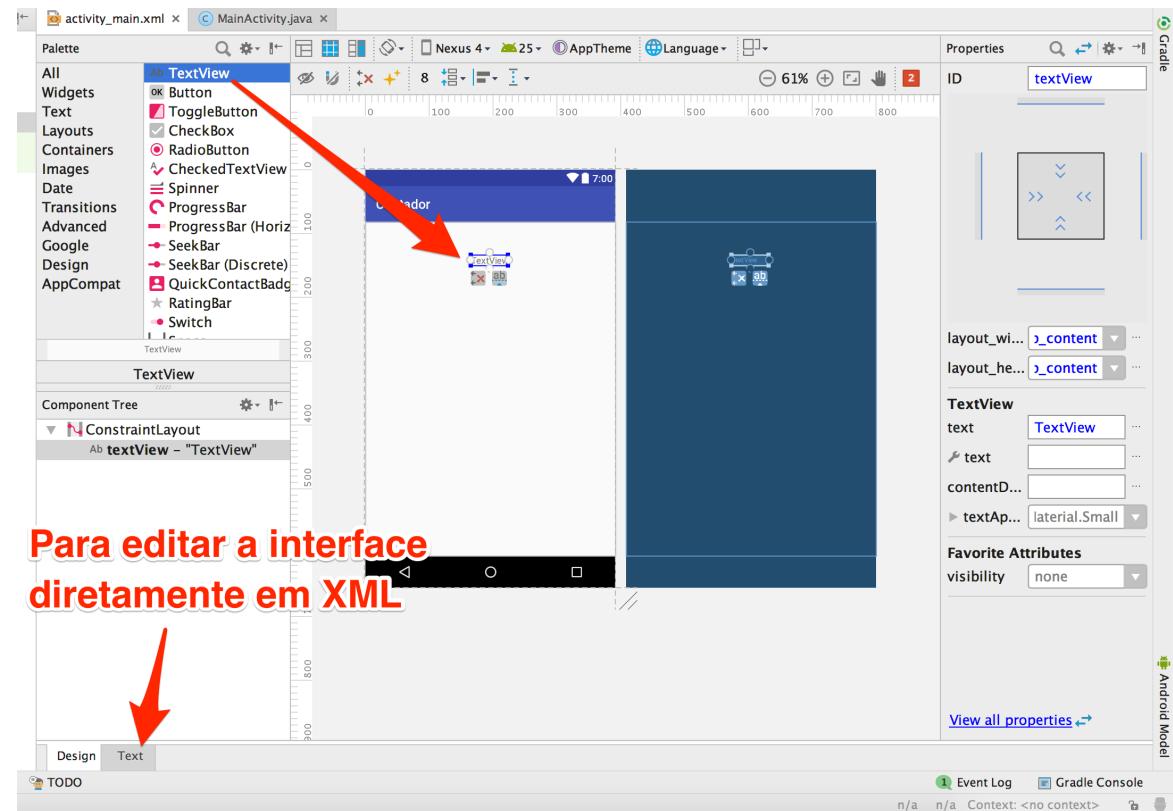
- Uma **tela** no Android é chamada de **Activity**.
- Para controlar o funcionamento da tela, criamos uma subclasse da **Activity** padrão existente no SDK (ou de uma de suas variações como **ActionBarActivity** e **ListActivity**).
- Os **layouts da tela** (seus componentes e suas configurações) ficam em um **arquivo XML** separado da classe Java, localizado na pasta **res\layout** do projeto.
- O arquivo onde você deve montar a interface é o **activity_main.xml**.
- Nas classes Activity, escrevemos o código Java que vai manipular a tela e seus componentes por meio de **eventos** (como o “toque” – ou clique – em um botão).

A tela principal do Android Studio



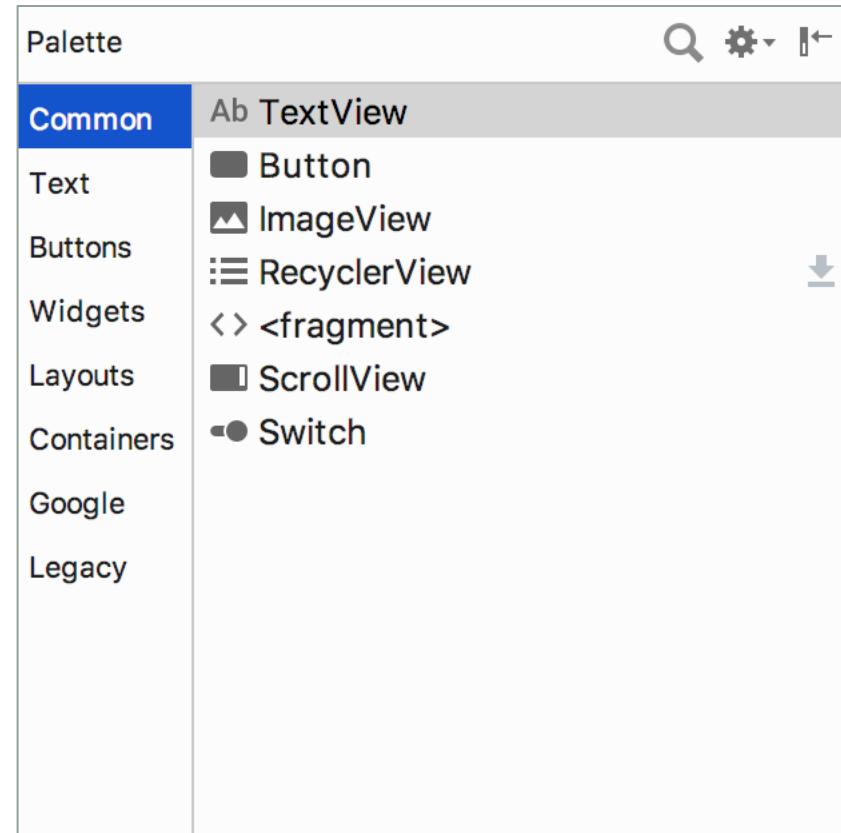
4.1 – Criando a interface do app

- No Android Studio, definimos a interface de uma tela do aplicativo “arrastando e soltando” elementos da **Palette** (palheta) diretamente pelo editor visual.



WIDGETS

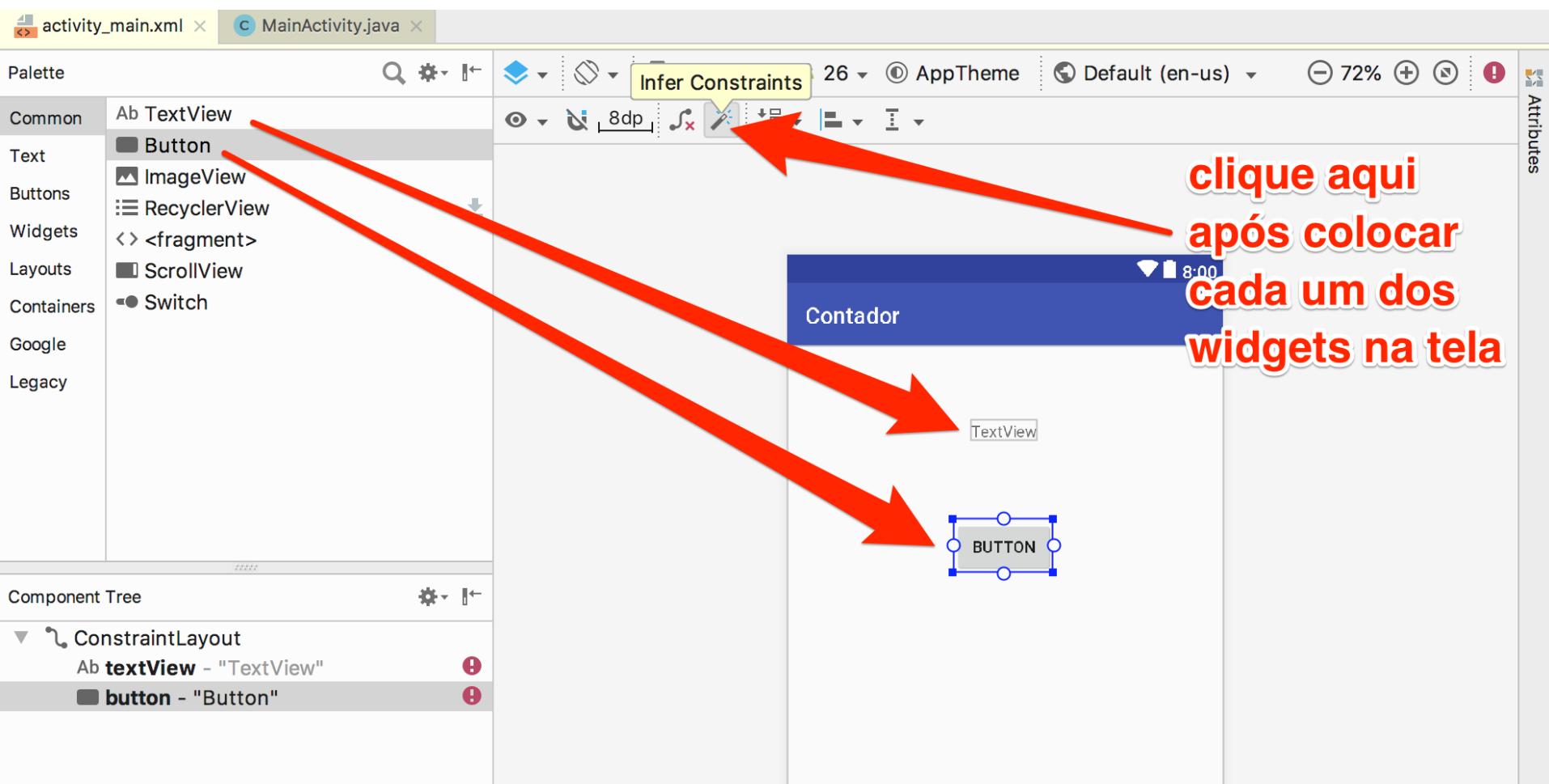
- Widgets são os componentes visuais que usamos para construir as telas de um aplicativo Android.
- Todos eles são definidos por classes dentro do pacote **android.widget**.
- Todos são subclasses de **android.view.View**, que representa um espaço retangular na tela.
- No Editor de telas do Android Studio, os widgets estão disponíveis na Palheta (**Palette**)



4.2 – Posicionando os widgets necessários ao Contador

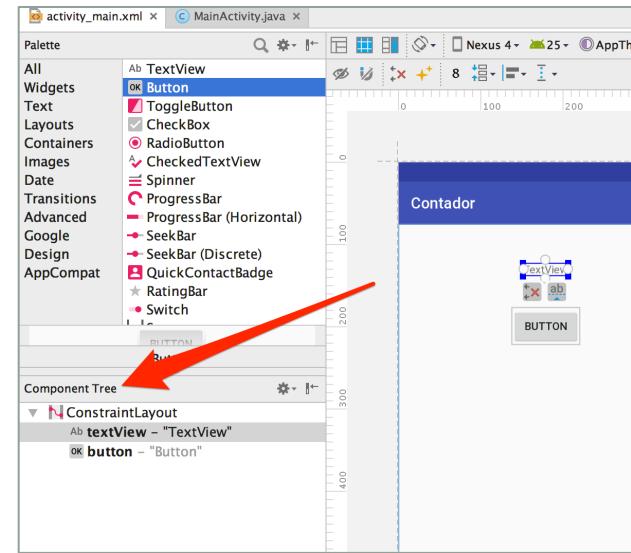
- Para o **Contador**, precisaremos ter dois widgets na tela.
- Um TextView: para exibição de texto que não pode ser alterado pelo usuário. São objetos da classe `android.widget.TextView`
 - Depois de colocá-lo centralizado na tela, clique sobre o botão indicado no próximo slide (**Infer Constraints**, com o ícone de uma varinha mágica)
- Um Button: para a ação do usuário (**clique**). É um objeto da classe `android.widget.Button`
 - Depois de colocá-lo centralizado na tela, clique de novo no botão **Infer Constraints**, assim como fez com o **TextView**.
- Veja o próximo slide como localizar os widgets e o botão Infer Constraints.

4.2 – Posicionando os widgets necessários ao Contador



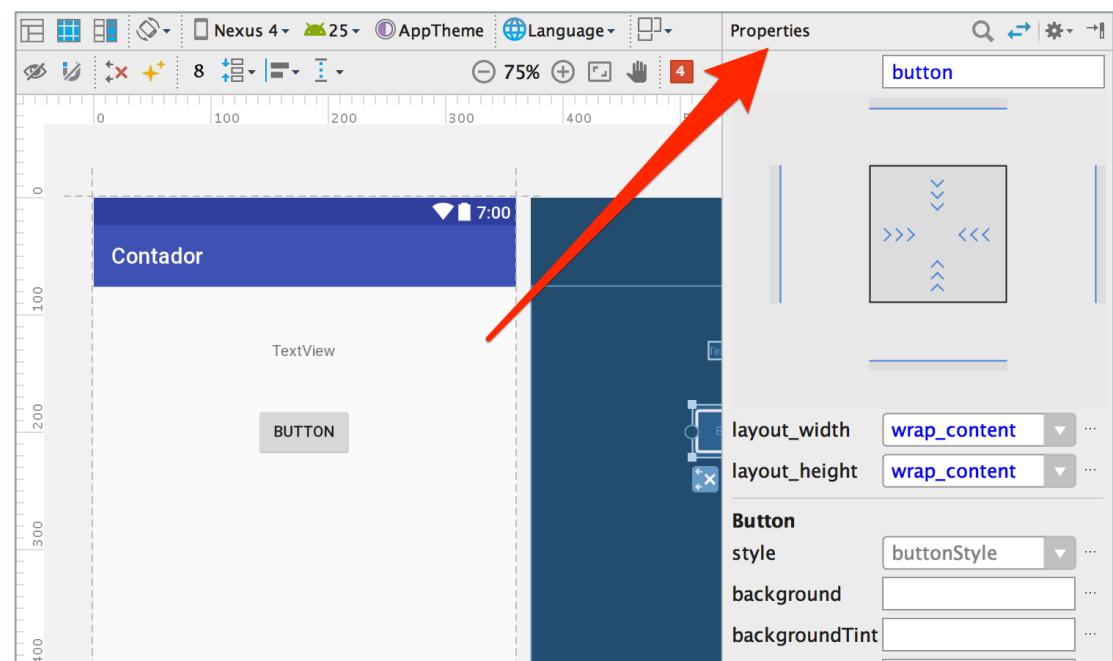
MAIS SOBRE WIDGETS

- Widgets possuem um nome, o seu **id**. Ele pode ser visto na **janela Component Tree**.
- Widgets como bons objetos, tem **propriedades e métodos**.
- Para acessar e alterar as **propriedades** dos widgets no código Java, utilizamos seus **getters e setters**.
- Mas enquanto estamos construindo a interface do app, podemos modificar as propriedades do widget clicando sobre ele para selecioná-lo e alterando suas propriedades pela janela Properties do Android Studio.



MAIS SOBRE WIDGETS

- Mas enquanto estamos construindo a interface do app, podemos modificar as propriedades do widget clicando sobre ele para selecioná-lo e alterando suas propriedades pela janela Properties do Android Studio.



4.3 – Configurando propriedades dos widgets

- A propriedade **text** de um widget serve para definirmos o texto que será exibido dentro dele.
- Altere as seguintes propriedades para os Widgets:
- Large Text (textView):
 - **text**: Contador
- Button (button)
 - **text**: Contar
 - **onClick**: contar
- Já a propriedade **onClick** (que usamos apenas para o botão) é usada para indicar o método Java que será executado quando o widget for tocado pelo usuário.

5.1 – Escrevendo o código Java para o evento de toque no botão

- Quando o usuário tocar no botão, incrementaremos uma variável contadora do tipo `int`, global na classe Java `MainActivity`, e mostraremos seu novo valor no `TextView`.
- Abra a classe Java **MainActivity** (existente na pasta **java** do projeto) e acrescente ao seu código o método **contar** e a declaração da variável contador, mostradas no slide seguinte.

```
package net.querino.contador;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    int contador = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

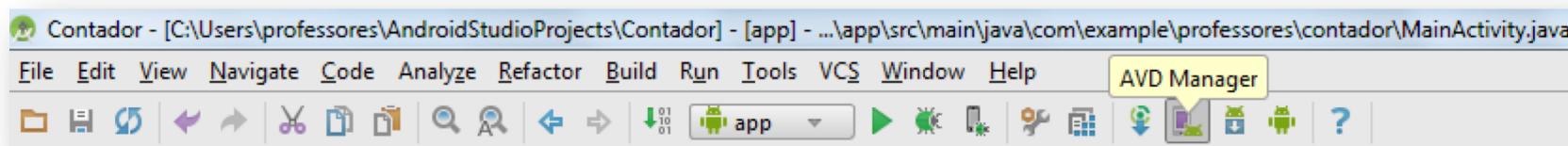
    public void contar(View view) {
        TextView textView = (TextView) findViewById(R.id.textView);
        contador++;
        textView.setText(String.valueOf(contador));
    }
}
```

5.2 – Entendendo o código

- A variável **contador**, local da classe **MainActivity**, vai guardar o valor da contagem, que será incrementado sempre que o usuário tocar no botão.
- O método **contar** será disparado quando o usuário tocar no botão. Dentro dele:
 - Fazemos uma conexão ao **TextView** que exibe a contagem, chamado **textView** no arquivo interface, e que terá esse mesmo nome na classe Java.
 - Incrementamos a variável **contador**.
 - Usamos o *setter* **setText** do **textView** para exibir o novo valor da variável contador nele. Repare que precisamos primeiro converter o valor de **int** para **String**.

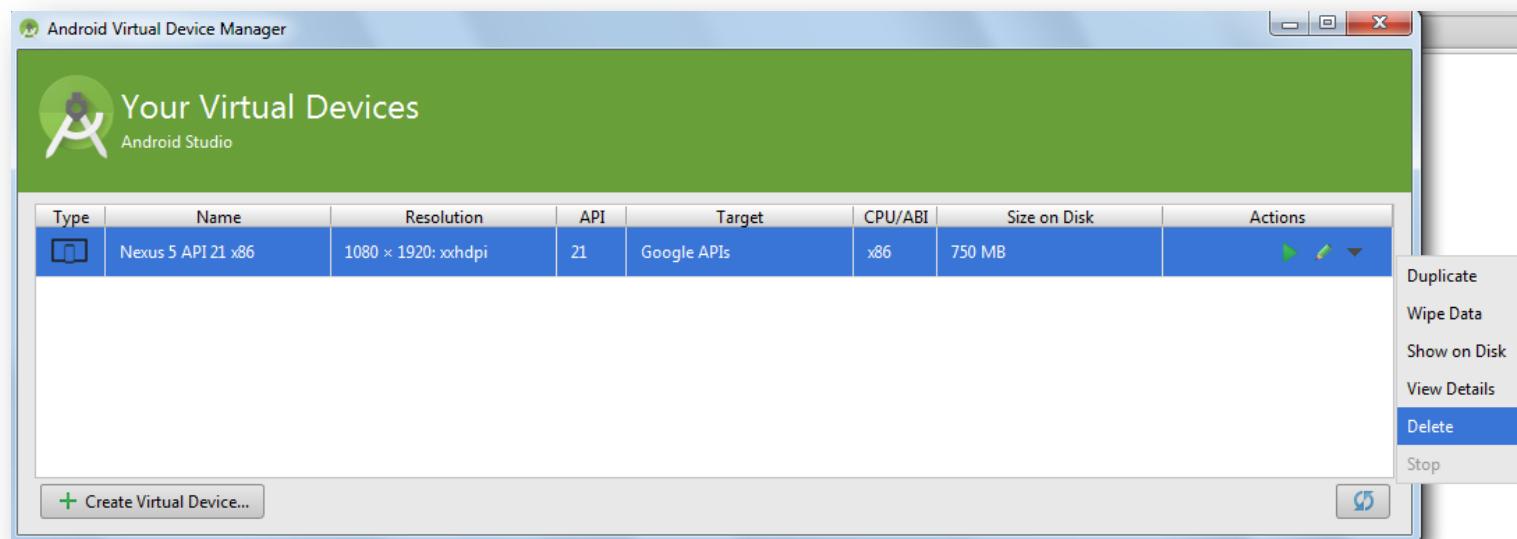
6.1 – Criar um AVD

- Você precisará criar um AVD se:
 - A versão do Android Studio que você baixou não venho com uma ou...
 - ...o AVD que veio com o Android Studio (plataforma x86) não funcionou no seu computador (**erro relativo ao Intel HAXM**)
- Para testar, avance para a ETAPA 7 e tente executar o aplicativo. Se o emulador não abrir, volte a este slide e...
- ...vamos criar um AVD novo que funcione! Abra o AVD Manager clicando neste botão da barra de ferramentas do Android Studio:



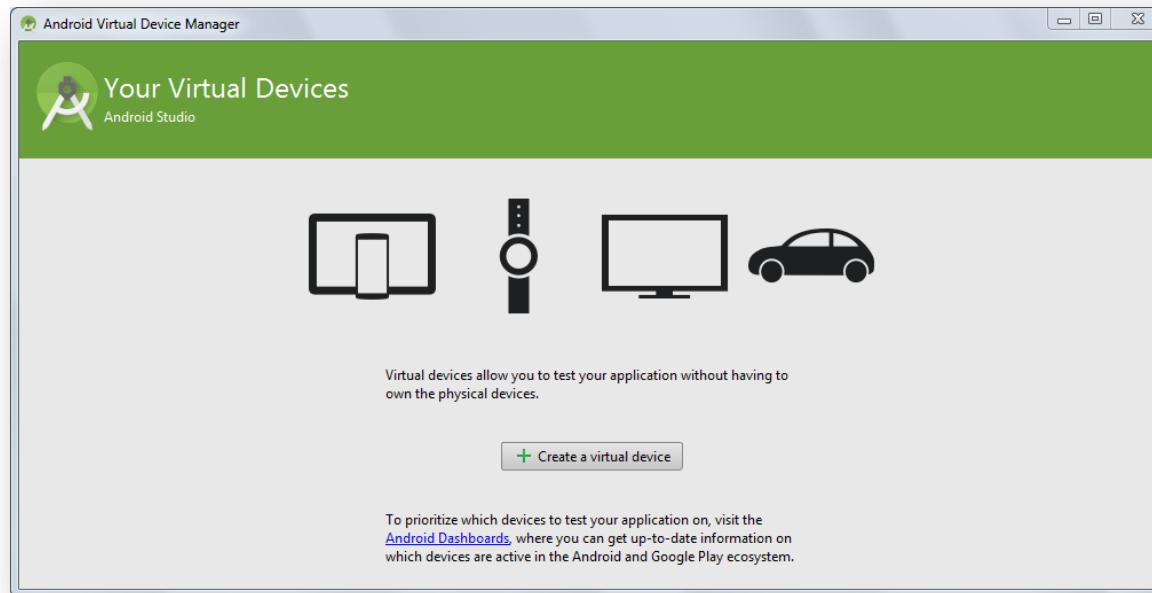
6.2 – Remova o AVD existente

- Clique com o botão direito sobre o AVD existente e selecione Delete para excluí-lo:



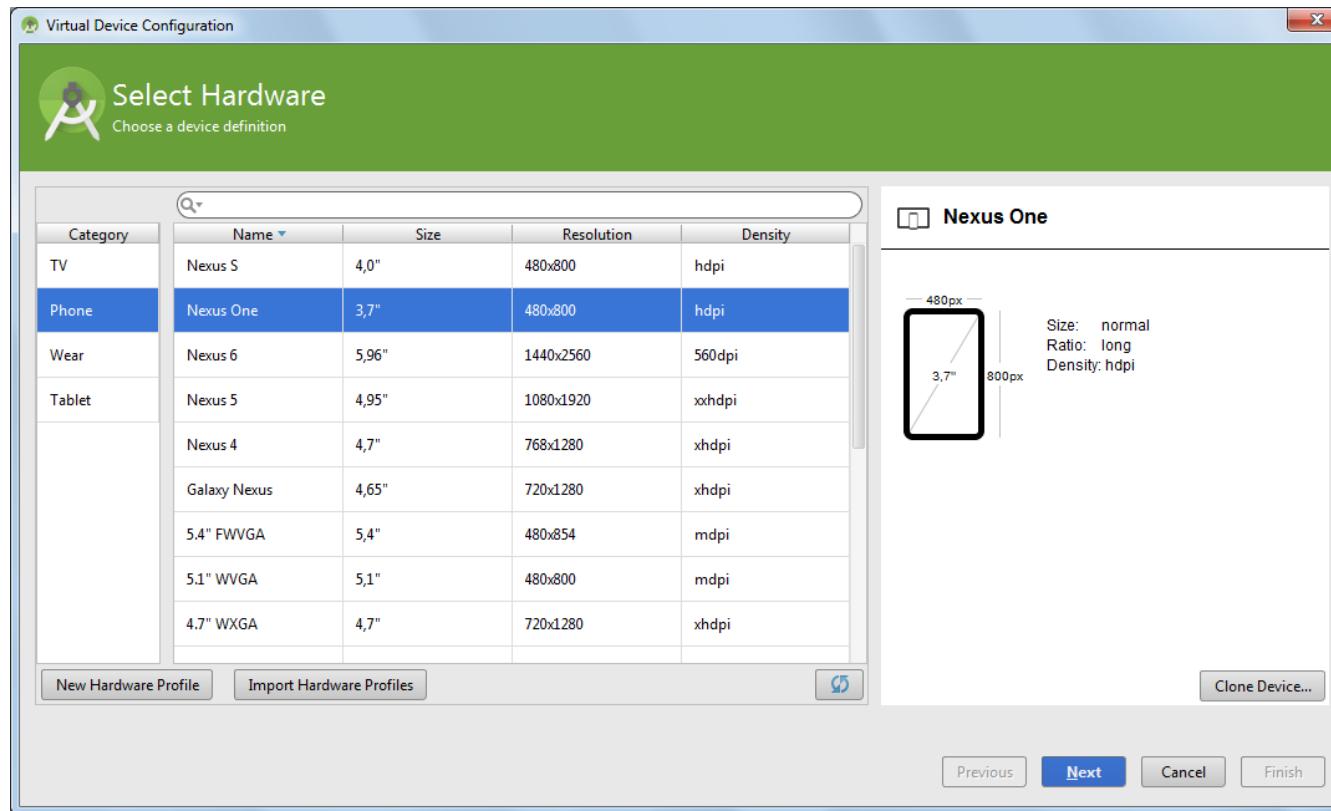
6.3 – Comece a criar o novo AVD

- Clique no botão **Create a virtual device** para iniciar a criação do novo AVD:



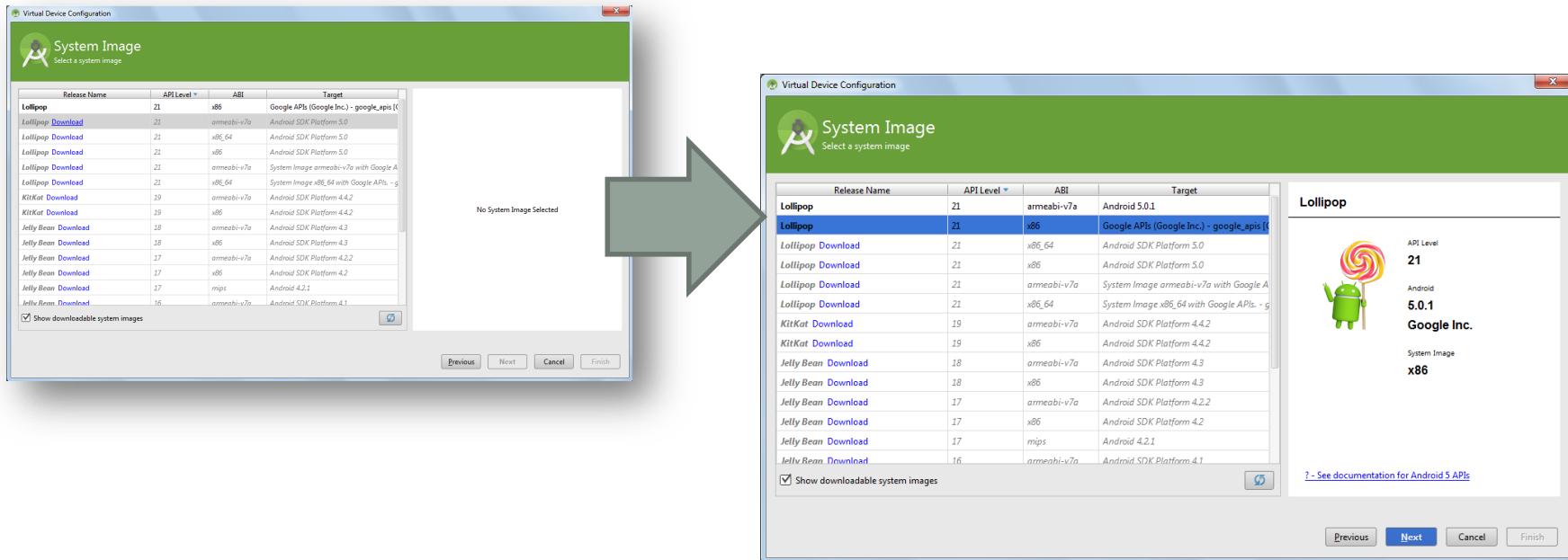
6.3 – Selecione uma configuração de aparelho para usar como modelo

- Selecione **Phone > Nexus One** e clique em **Next**



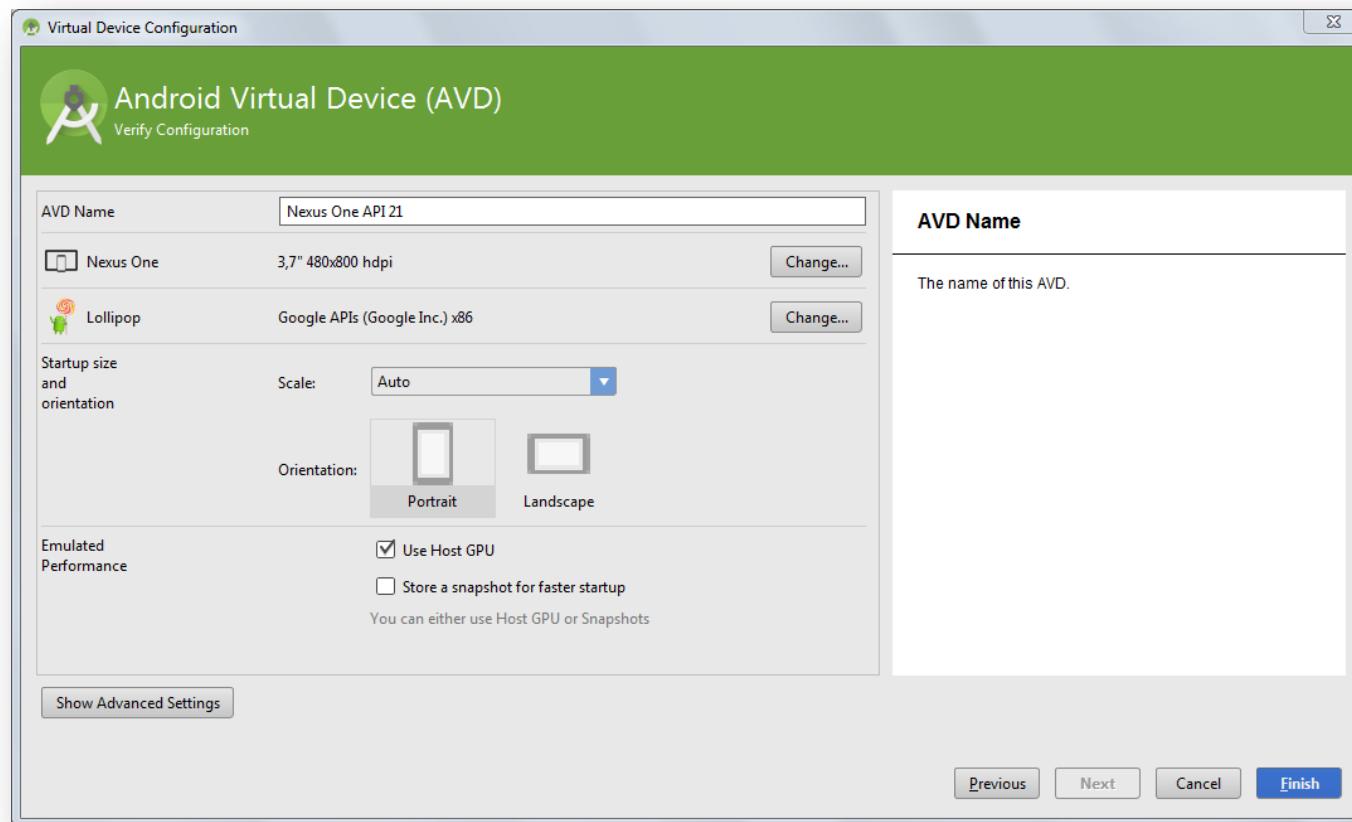
6.4 – Baixe a imagem da versão ARM do Android Lollipop (ou versão mais recente)

- Localize a linha relativa à versão mais recente do Android para a plataforma (ABI) **armeabi-v71** e clique em **Download**.
- Na tela que será mostrada, clique em **Accept** e **Next** para iniciar o download.
- Quando o download tiver finalizado, selecione a linha relativa a versão baixada e clique em **Next**.



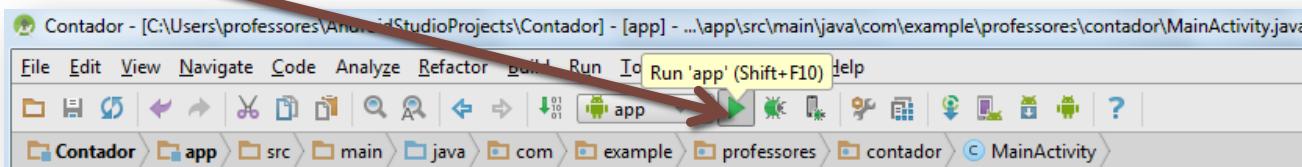
6.5 – Finalizando a criação do AVD

- Última tela! Clique em **Finish** para criar o AVD.

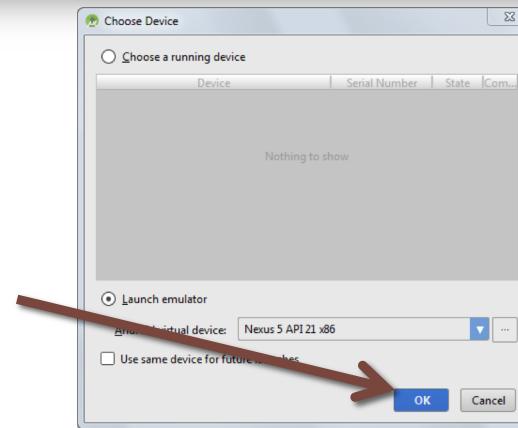


7.1 – Executando no emulador

- Para compilar e executar o projeto no emulador, clique no botão **Run app** da barra de ferramentas:



- Confirme a abertura do emulador clicando em **OK** na tela ao lado:



- Após o emulador ser inicializado, o aplicativo será instalado a iniciado automaticamente. Talvez seja necessário “desbloquear” a tela de abertura do Android (arraste o mouse com o botão esquerdo pressionado).

PARA VOCÊ TENTAR FAZER

- Crie um projeto com a interface exibida a direita.
- Para cada linha, coloque dois **TextView** grandes, um ao lado do outro, seguidos de um **Button**.
- Quando o usuário clicar no botão da linha de cima, exiba a data atual no formato dd/mm/yyyy no **TextView** à sua esquerda.
- Quando o usuário clicar no botão da linha de baixo, exiba a hora atual no formato hh/MM/ss no **TextView** à sua esquerda.
- Dica: use as classes **Date** e **SimpleDateFormat** do Java.

