

Baby

Steps



Android Studio

# Instalação - Requisitos Mínimos

## WINDOWS

Windows 8/7/Vista/2003 (32 ou 64 bits)

Recomendado 4gb de RAM

Espaço livre 2gb

Kit de desenvolvimento Java (JDK) 8]

## LINUX

GNOME ou KDE

Biblioteca GNU 2.15 ou superior

Espaço livre 2gb

Kit de desenvolvimento Java (JDK) 8



# O que é o Android?

**Android** é um sistema operacional baseado no núcleo Linux e atualmente desenvolvido pela empresa de tecnologia Google. Com uma interface de usuário baseada na manipulação direta, o Android é projetado principalmente para dispositivos móveis com tela sensível ao toque.





A green pyramid divided into four horizontal layers. The top layer is labeled 'Aplicativos' and is surrounded by social media icons: Instagram, Facebook, Twitter, and WhatsApp. The second layer is labeled 'Frameworks, Serviços e Bibliotecas' and is flanked by the Android logo and the Retrofit logo with their respective URLs. The third layer is labeled 'Bibliotecas e Serviços nativos' and is flanked by rounded rectangular boxes containing the terms 'Intents', 'Widgets', 'Activities', and 'Services'. The bottom layer is labeled 'Linux' and is topped by a Tux penguin and an Android robot icon.

Aplicativos

Frameworks, Serviços e Bibliotecas

Bibliotecas e Serviços nativos

Linux



<http://androidannotations.org/>

Retrofit

<http://square.github.io/retrofit/>

Intents

Widgets

Activities

Services



A figura anterior mostra todas as camadas da plataforma e podemos ver que a base de todo Android é desenvolvida baseada em Linux:

- Aplicativos: São os aplicativos e jogos desenvolvidos em Java.
- Frameworks, serviços bibliotecas, geralmente, também escritos em Java e servem para facilitar o desenvolvimento de aplicativos e jogos.
- Bibliotecas e serviços nativos são recursos que já vêm com o Android, para serem utilizados pelo desenvolvedor.
- Linux é a base de tudo e inclui todos os drivers de hardware e redes (Bluetooth, câmeras, USB, GPS's etc), sistemas de arquivos e processamento.



Todos estão  
com o ambiente  
configurado ?







# Welcome to Android Studio

## Recent Projects

No Project Open Yet

## Quick Start



Start a new Android Studio project



Open an existing Android Studio project



Import an Android code sample



Check out project from Version Control



Import Non-Android Studio project



Configure



Docs and How-Tos







## New Project

Android Studio

### Configure your new project

Application name:

Company Domain:

Package name:

androidpro.com.br.meuapp

Edit

Project location:



Previous

Next

Cancel

Finish

Os elementos dessa primeira tela antes de você começar as configurações:

- **Application Name:** Nome do aplicativo que será configurado automaticamente. É o nome que aparece no ícone, quando você instala um aplicativo no seu celular.
- **Company Domain:** É o nome do pacote onde ficarão seus códigos Android. No desenvolvimento Java, é uma boa prática utilizar desta forma, considerando que estamos no Brasil, `br.com.suaaplicacao`.
- **Project Location:** É a pasta ou local onde seu projeto será guardado dentro do seu computador.



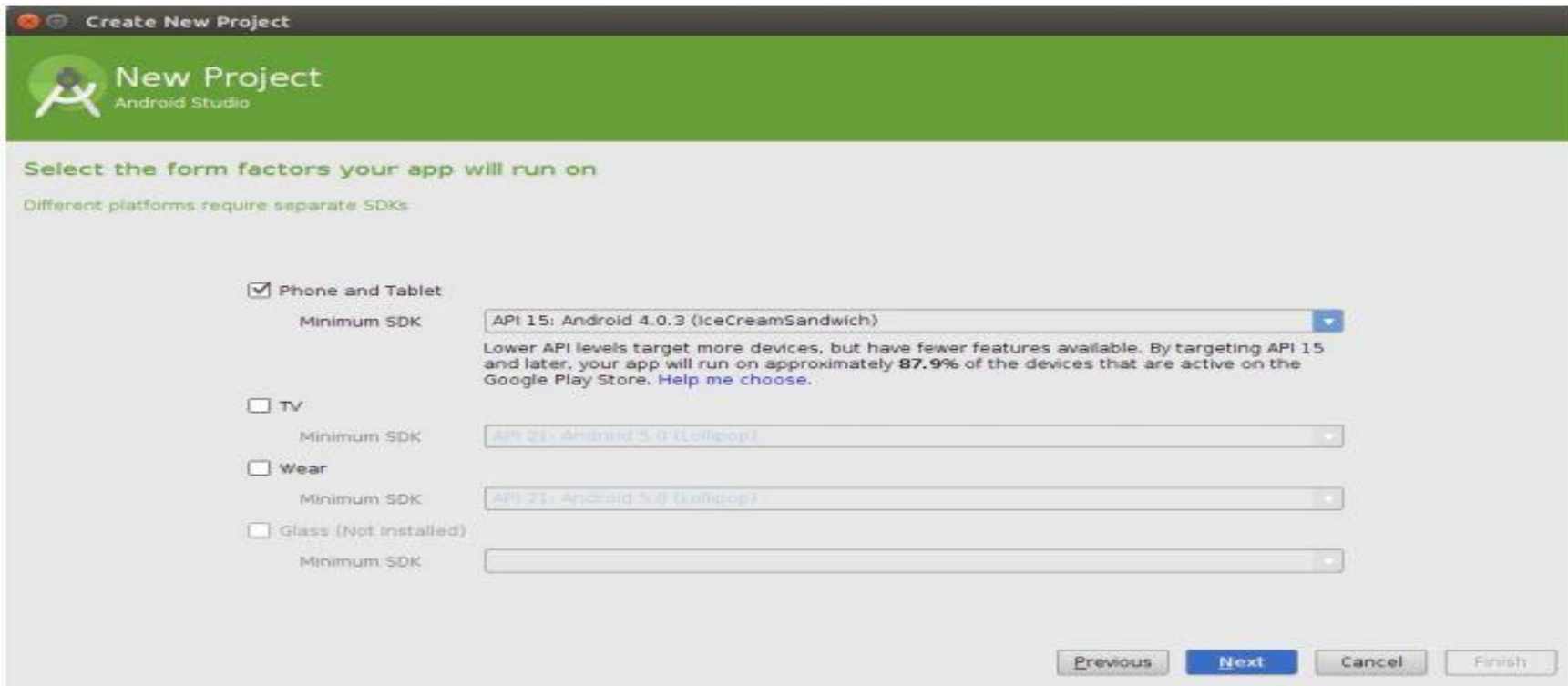
Preencha os dados da tela conforme mostra a figura anterior:

- Application Name: MeuApp
- Company Domain: br.com.seunomesobrenome

Clique no botão Next > para ir para o próximo passo da configuração.



Na tela abaixo, deixe as configurações da forma que estão. Nosso projeto irá suportar qualquer tablet e celular Android a partir da versão 4.0.3 da plataforma.



**Create New Project**

**New Project**  
Android Studio

Select the form factors your app will run on.

Different platforms require separate SDKs.

☒ **Phone and Tablet**

Minimum SDK: **API 15: Android 4.0.3 (IceCreamSandwich)**

Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately 87.9% of the devices that are active on the Google Play Store. [Help me choose.](#)

☐ **TV**

Minimum SDK: **API 21: Android 5.0 (Lollipop)**

☐ **Wear**

Minimum SDK: **API 21: Android 5.0 (Lollipop)**

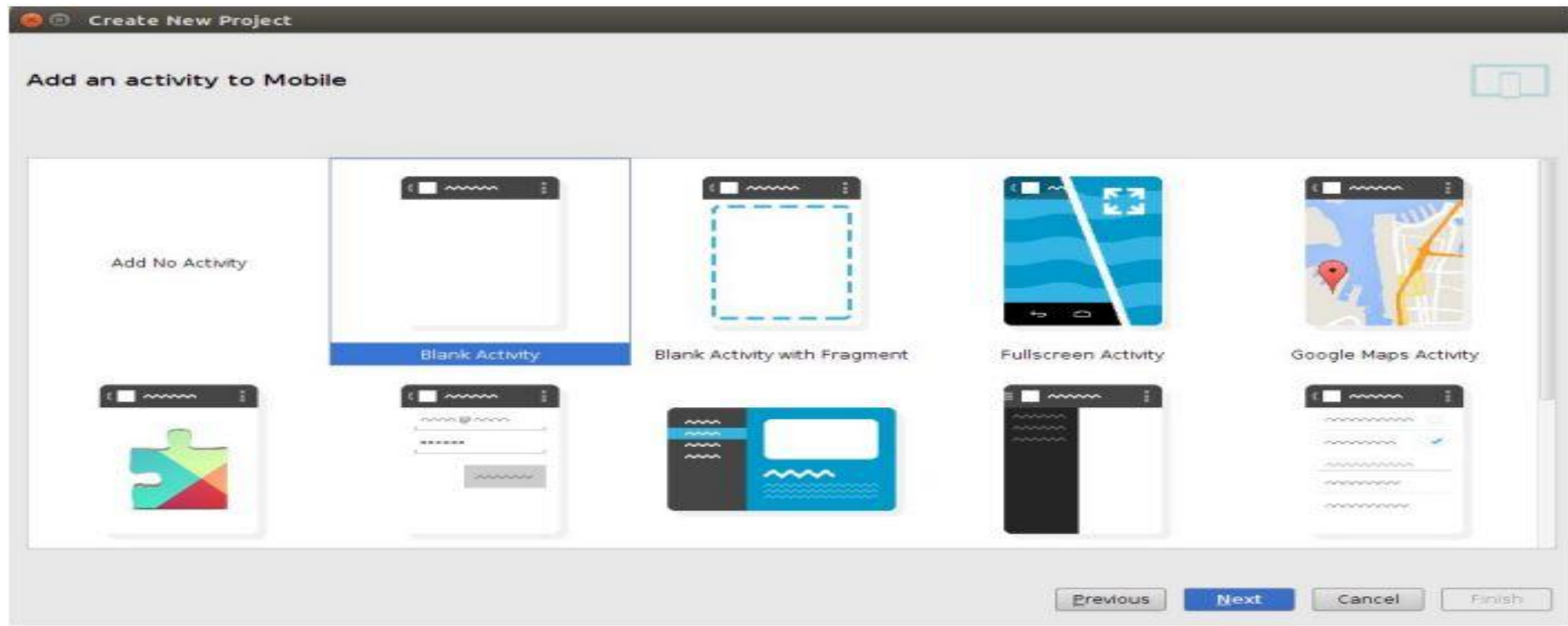
☐ **Glass (Not installed)**

Minimum SDK:

**Previous** **Next** **Cancel** **Finish**

Clique no botão Next > para ir para o próximo passo da configuração.


Existem algumas opções de aplicativo que o Android Studio oferece para a criação do nosso projeto. Isso facilita bastante na hora de começar um projeto novo.



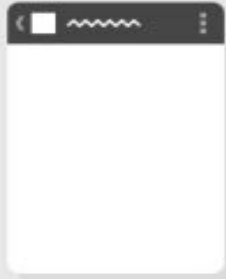
Clique no botão Next > para ir para o próximo passo da configuração.

Create New Project

Choose options for your new file



Creates a new blank activity with an action bar.



Blank Activity

Activity Name: MainActivity

Layout Name: activity\_main

Title: MainActivity

Menu Resource Name: menu\_main

The name of the activity class to create

Previous Next Cancel Finish

Clique no botão Finish para terminar todo o processo.

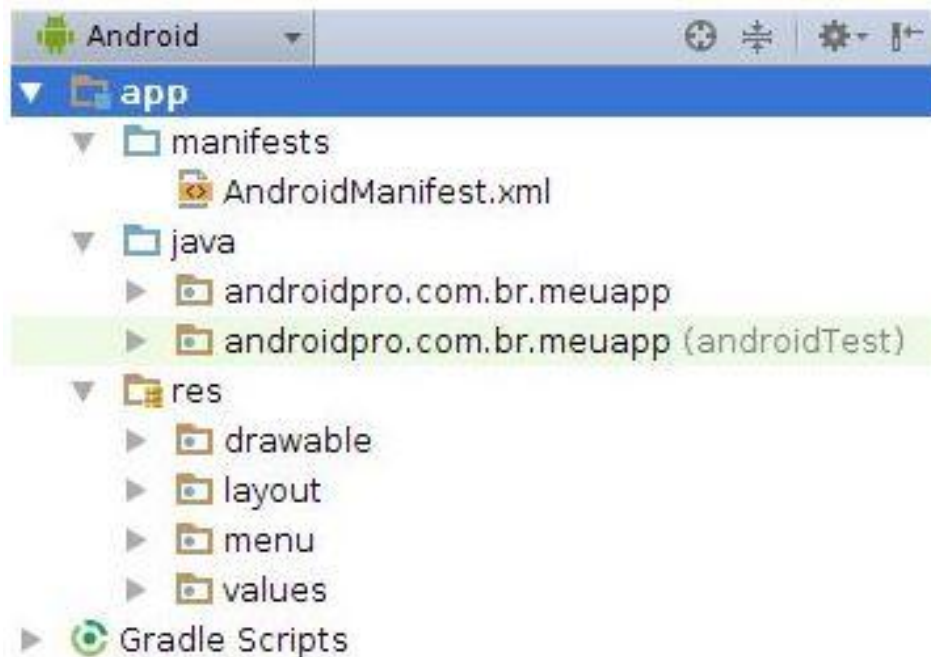


**EXECUTIVE PRODUCERS**

**TREY PARKER  
MATT STONE**

**NO, WE JUST--  
NO, NO, WAIT!**

# Entendendo o projeto





# AndroidManifest.xml

é responsável por definir as características do projeto como permissões, versão, logo, nome e seus componentes. Ao longo dos estudos, você aprenderá várias configurações deste arquivo.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="androidpro.com.br.meuapp" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

## Outras partes que compõem seu projeto Android são:

- `java/`: É a pasta onde ficam as fontes do seu projeto, ou seja, as classes Java que você desenvolve.
- `res/`: Contém vários recursos do projeto como imagens, layout, xmls de configuração como veremos a seguir.



- drawable/: existem várias pastas drawable que contêm as imagens utilizadas no projeto. Cada uma das pastas contém uma versão de uma determinada imagem, separadas por definição de tela.
- layout/: nesta pasta, ficam os arquivos responsáveis pelo design das telas do seu projeto.
- menu/: nesta pasta, ficam os arquivos xml referente aos menus do seu projeto.
- values/: contém outras configurações em xml para parametrização do projeto como cores, mensagens dentre outras

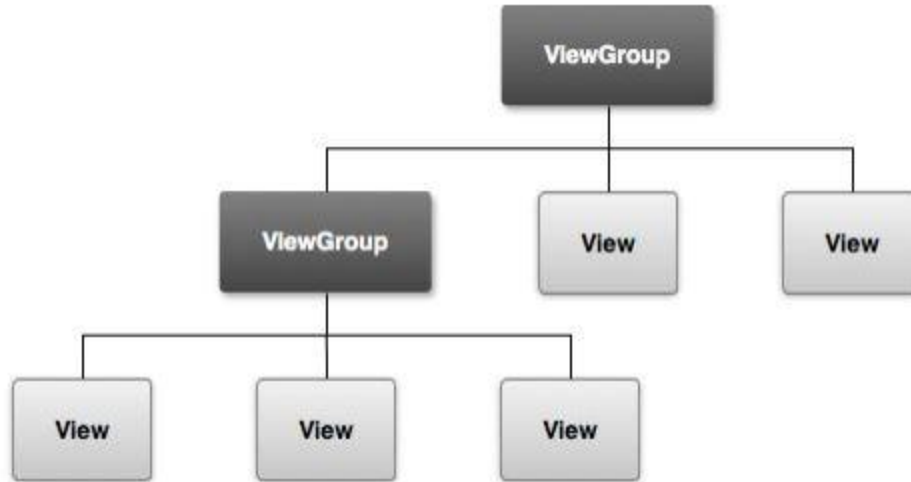


# Interface

Basicamente, todos os componentes visuais do Android são baseados em View e ViewGroup. As Views são componentes como botões, campos de texto, combo-box dentre outros. Os ViewGroups são componentes de organização de layout como listas e tabelas, ou seja, agrupamento de Views.



A hierarquia dos componentes pode ser representada desta forma:



Para entender melhor sobre o View e ViewGroup, você vai usar seu projeto como exemplo, abra o arquivo activity\_main.xml. Quando abrir o arquivo, clique na aba no canto inferior esquerdo chamada Text.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="
match_parent"
    android:layout_height="match_parent" android:paddingLeft="
@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".
MainActivity">

    <TextView android:text="@string/hello_world" android:layout_width="
wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```



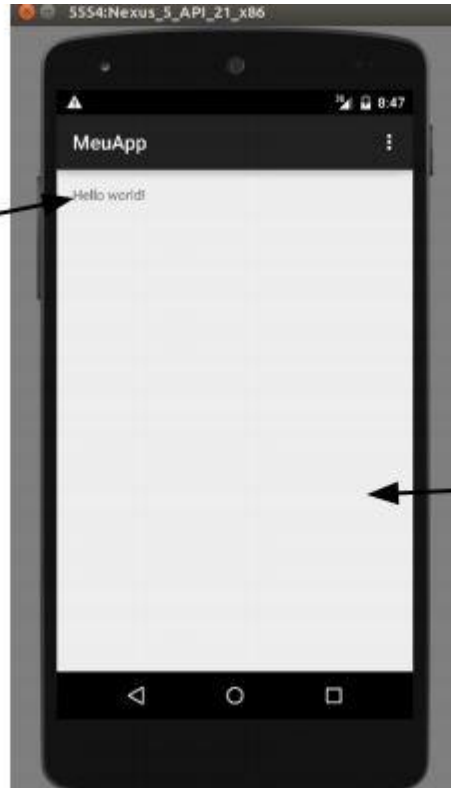
Nesse exemplo, há dois componentes que compõem a interface da tela principal do projeto:

- **RelativeLayout**: é um ViewGroup que posiciona os componentes filhos em relação uns aos outros ou em relação ao próprio RelativeLayout.
- **TextView**: é uma View que basicamente serve para mostrar um texto para o usuário.





TextView



RelativeLayout



## Mão na massa

Agora, coloque a mão na massa, adicionando dois componentes novos em sua tela principal. Remova o TextView e adicione um EditText e um Button.



# Resultado

```
<!-- Campo texto para entrada de dados pelo usuário -->
<EditText
    android:id="@+id/editTextNome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:hint="Digite seu nome"/>

<!-- Botão de ação -->
<Button
    android:id="@+id/buttonAcao"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editTextNome"
    android:text="Saudação" />

</RelativeLayout>
```



# Entenda alguns atributos importantes das Views:

- `android:layout_width/android:layout_height`: definem o tamanho do componente e são obrigatórios para todas as Views.
- `android:id`: é um identificador único que faz referência a View. Sempre que for definir um id para um componente, ele seguirá essa forma `@+id/NOME_DO_ID`, onde o `@+` indica que estamos criando um id para aquela View.
- `android:hint`: é um valor padrão que é mostrado, quando o campo texto está vazio.
- `android:layout_below`: posiciona o componente abaixo de outra View



# Execute





## Aviso

Voltando um pouco para o layout principal `activity_main.xml`, você verá que o Android Studio está sublinhando de amarelo duas linhas do seu arquivo e mostrando a seguinte mensagem.

“[I18N] Hardcoded string "Digite seu nome", should use @string resource”

“[I18N] Hardcoded string "Saudação", should use @string resource”



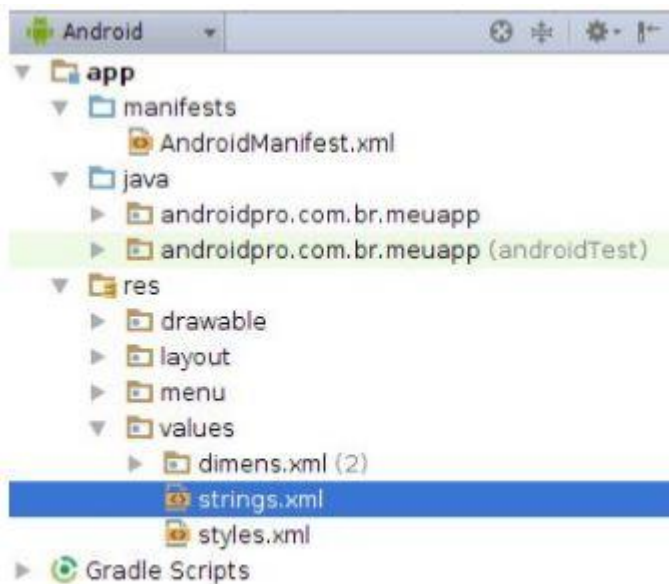
## Mas o que isso quer dizer?

A ferramenta está avisando-o de que não está seguindo uma boa prática de desenvolvimento e está colocando as mensagens direto na sua View. Ele ainda indica para usar uma `@string` resource no lugar das mensagens.

Qualquer texto que for utilizado no seu projeto, sempre deve estar dentro de Resources. Os Resources permitem que você gerencie suas mensagens e outros dados em um único local e ajuda, também, a adicionar suporte de vários idiomas ao seu aplicativo.







# Vamos adicionar mais duas strings no arquivo, que deve ficar assim:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">MeuApp</string>
    <string name="hello_world">Hello world!</string>
    <string name="action_settings">Settings</string>
    <string name="hint_edit_mensagem">Digite seu nome</string>
    <string name="text_btn_acao">Saudação</string>

</resources>
```

Basicamente, você adicionou uma tag string com um conteúdo e deu um nome a ela.  
**<string name="nome\_x">Mensagem X</string>**



Lembrete: todas as mensagens utilizadas no seu projeto devem estar dentro desse arquivo e nunca diretamente no código ou nos arquivos de layout.



**Uhuu! Você adicionou as strings dentro do arquivo, agora precisa atualizar o arquivo activity\_main.xml.**

```
<!-- Campo texto para entrada de dados pelo usuário -->
<EditText
    android:id="@+id/editTextNome"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="text"
    android:hint="@string/hint_edit_mensagem"/>

<!-- Botão de ação -->
<Button
    android:id="@+id/buttonAcao"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/editTextNome"
    android:text="@string/text_btn_acao" />
```



# Programação

- Abra o arquivo MainActivity.java
- Essa é uma Activity, um dos componentes mais importantes do Android. Ela é responsável por fazer a interação da interface gráfica com outras funcionalidades e bibliotecas.
- Todas as telas do seu projeto devem estender uma Activity, dessa forma você fala para o Android que aquela sua classe é uma Activity também.
- Existe um método muito importante dentro de uma Activity que é o onCreate, ele é responsável por fazer a criação dos componentes na tela. Sempre que quiser trabalhar com um componente de tela usará o onCreate.



- Dentro do onCreate, há duas chamadas de métodos, o `super.onCreate(savedInstanceState)` que é o onCreate nativo da plataforma e o `setContentView(R.layout.activity_main)` que é onde relacionamos nosso layout com nossa Activity.
- Sendo assim, você precisa resgatar os componentes pelos id's, para poder trabalhar com eles. A Activity do Android tem um método chamado `findViewById` onde pode pegar seu campo de texto e o botão.



```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        // Resgatamos nossos componentes pelo id  
        final EditText editTextNome = (EditText) findViewById(R.id.editTextNome);  
        final Button btnAcao = (Button) findViewById(R.id.buttonAcao);  
    }  
  
    ...  
}
```



- Clique em cima dos textos em vermelho e aperte os comandos Alt + Enter até todas as classes serem importadas.
- Agora, você vai adicionar uma ação em seu botão, utilizando o método `setOnClickListener`.
- Dentro do clique do botão, pega o nome dentro do campo de texto e mostra uma mensagem na tela do usuário, utilizando um Toast. O Toast é uma mensagem rápida que aparece para o usuário e desaparece depois de alguns segundos.





# Resultado

```
// Resgatamos nossos componentes pelo id
final EditText editTextNome = (EditText) findViewById(R.id.editTextNome);
final Button btnAcao = (Button) findViewById(R.id.buttonAcao);

// Adicionando uma ação no clique do botão
btnAcao.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Pegamos o conteudo do campo de texto
        String nome = editTextNome.getText().toString();

        // Mostramos uma mensagem na tela do usuário
        Toast.makeText(MainActivity.this, "Olá, " + nome + " tudo
bem?", Toast.LENGTH_LONG).show();
    }
});
...

```



# Execute



# Desafios

## #1 Desafio: Boas práticas

Adicione a mensagem de saudação dentro de **strings.xml** e o utilize dentro do **Toast**.

Dica: para recuperar a mensagem na Activity utilize o método `getString(resId)`

## #2 Desafio: Componentes da tela

Adicione mais um campo sobrenome abaixo do campo nome e o utilize na mensagem de saudação.



# COMUNIDADES

- python sorocaba - <https://groups.google.com/forum/m/#!forum/python-sorocaba>
- guru sorocaba - <https://www.facebook.com/GURUSorocaba/>
- hacklab - <https://www.facebook.com/hacklabsorocaba/>
- GDG Sorocaba - <https://www.facebook.com/GDGSorocaba/>
- SorocabaJS - <https://www.facebook.com/sorocabajs/>
- JS Meetup - <http://www.meetup.com/GDG-Sorocaba/>
- WomenTechmakers - <https://www.facebook.com/groups/wtmsorocaba/>
- in.tegra.do - <http://www.meetup.com/In-Tegra-Do/>
- english meetup - <http://www.meetup.com/Sorocaba-English-Conversation-Happy-Hours/>
- AndroidIsIzi - <https://www.meetup.com/pt-BR/Android-is-IZI/>
- Node Sorocaba - <https://www.meetup.com/pt-BR/Sorocaba-Node-User-Group-SNUG/>
- Sorocaba CSS - <http://www.meetup.com/pt-BR/SorocabaCSS/>



# FIM



## Jean Peixoto

Apresentação baseada no  
conteúdo disponibilizado pelo  
**AndroidPro.**

