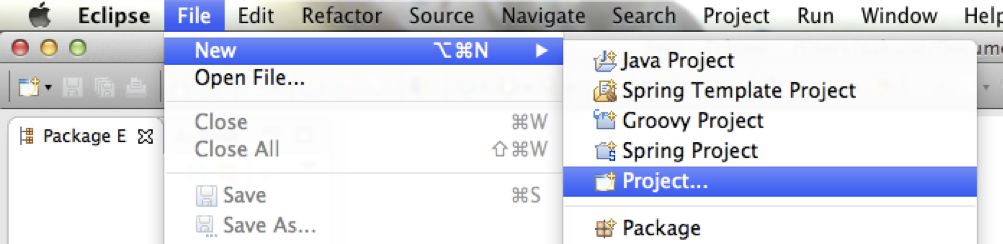
**Android - Olá Mundo, criando uma aplicação com TextView, ImageView e Toast.**

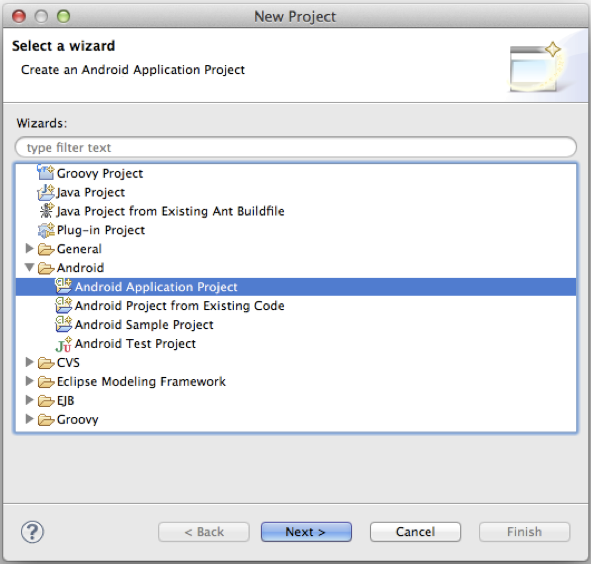
Vamos criar uma aplicação inicial (Olá Mundo!!!) para o Android que utiliza uma Activity e os componentes TextView e ImageView. A aplicação ficará conforme a imagem a seguir:

[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-01.png)

Para criar um projeto Android escolha no Eclipse a opção **File (Arquivo) → New (Novo) → Project… (Projeto…)**.

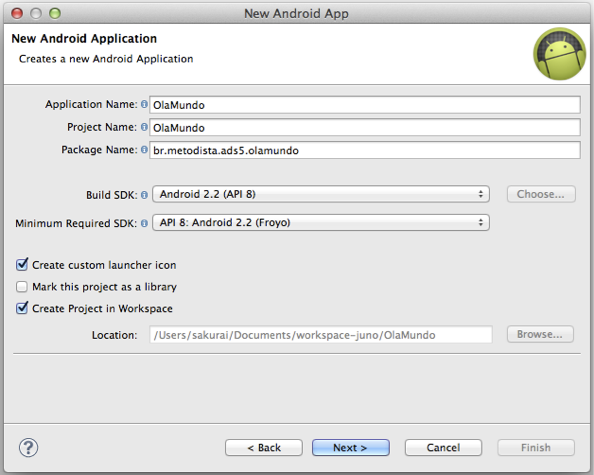
[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-02.png)

Na tela **New Project (Novo Projeto)** escolha no item **Android** o tipo de projeto **Android Application Project (Projeto de Aplicação Android)** e clique em **Next > (Próximo >)**.

[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-03.png)

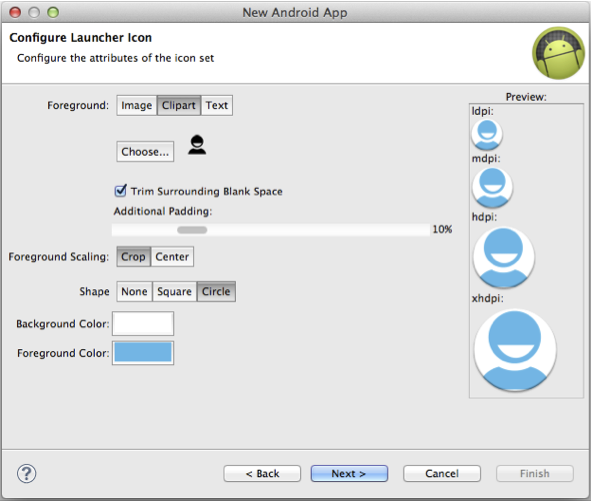
Na tela **New Android App (Nova Aplicação Android)** configure:

* **Application Name (Nome da aplicação):** OlaMundo o nome da aplicação;
* **Package Name (Nome do pacote):** br.metodista.ads5.olamundo;
* **Build SDK (SDK de construção):** Android 2.2 (API 8) - está é a versão da API do Android utilizada no projeto;
* **Minimun Required SDK (SDK mínimo requerido):** API 8: Android 2.2 (Froyo) - é a versão mínima do Android que está aplicação suporta.

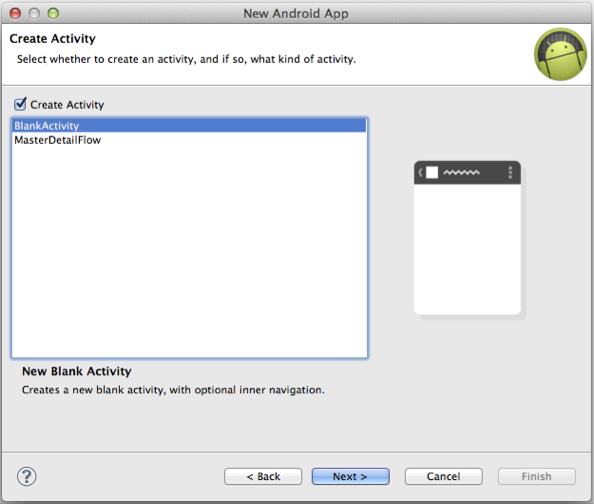
[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-04.png)

Note no **Location (Local)** onde será salvo o projeto, utilize este local para obter o código fonte do projeto. Clique em **Next > (Próximo >)** para continuar a criação do projeto.

Nesse momento é possível definir qual será o ícone inicial da aplicação, podendo escolher uma imagem, um clipart ou um texto; cores; tipo do ícone quadrado, circular ou nenhum; entre outros. No**Preview (Pré Visualização)** há diversos tamanhos de ícones apresentados, como **ldpi** (resolução baixa com ~120dpi), **mdpi** (resolução média com ~160dpi), **hdpi** (resolução alta com ~240dpi) e**xhdpi** (resolução muito alta com ~320 dpi). Com base no tamanho da resolução do aparelho, o Android irá apresentar o ícone com a melhor qualidade possível.

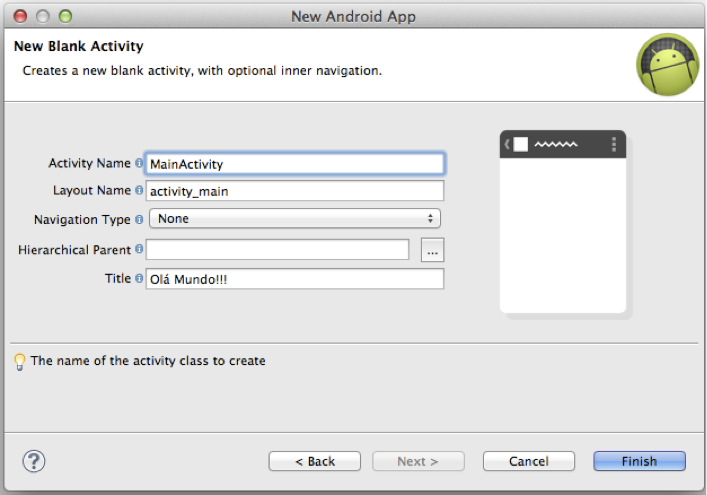
[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-05.png)

Clique em **Next > (Próximo >)** para definir se será criado uma Activity inicial para o projeto, utilizaremos as configurações padrões, clique em **Next > (Próximo >)** para prosseguir.

[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-06.png)

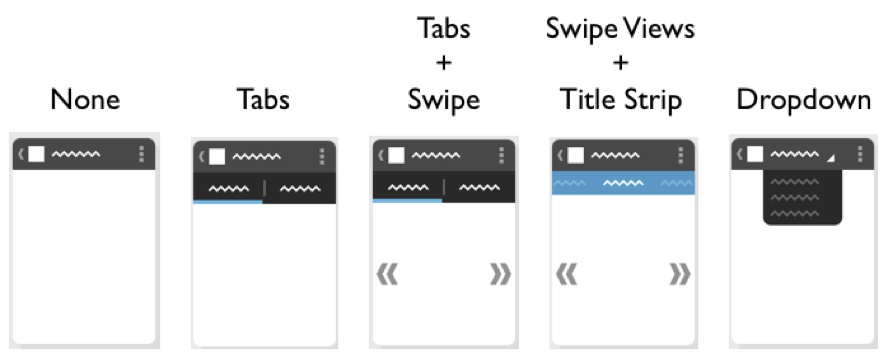
Nessa tela será criada uma nova Activity, que será a tela inicial da aplicação respondendo pelas ações da tela, faça as seguintes configurações:

* **Activity Name (Nome da activity):** MainActivity;
* **Layout Name (Nome do layout):** activity\_main.xml - este é um arquivo xml que contém os componentes que serão apresentados na página;
* **Navigation Type (Tipo de navegação):** None - este tipo é utilizado para escolher como será a navegação entre as páginas;
* **Hierarchical Parent (Parente hierárquico):** Em branco - essa página é chamada quando o usuário está nesta activity e clica no botão voltar do dispositivo móvel;
* **Title (Titulo):** Olá Mundo!!!

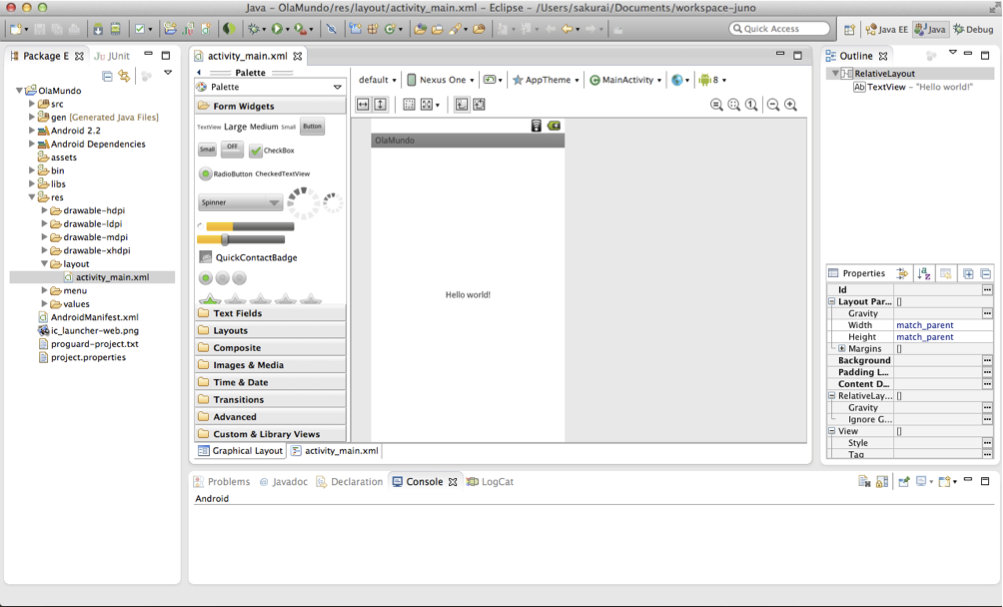
[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-07.png)

A navegação padrão entre as páginas podem ser dos tipos:

* **Tabs** (navegação por abas);
* **Tabs + Swipe** (Navegação por abas e passando o dedo na tela para a esquerda ou direita;
* **Swipe Views + Title Strip** (Passando o dedo na tela ou no menu para a esquerda ou direita);
* **Dropdown** (Menu que quando clicado apresenta uma lista com os itens).

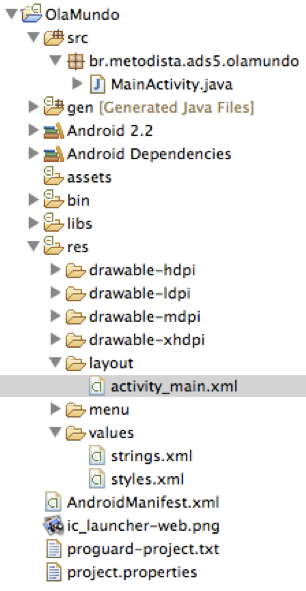
[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-08.png)

Clique em **Finish (Finalizar)** para terminar a criação do projeto. O Eclipse irá alterar a visualização da tela para o modo de edição de projetos Android, conforme a imagem a seguir:

[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-09.png)

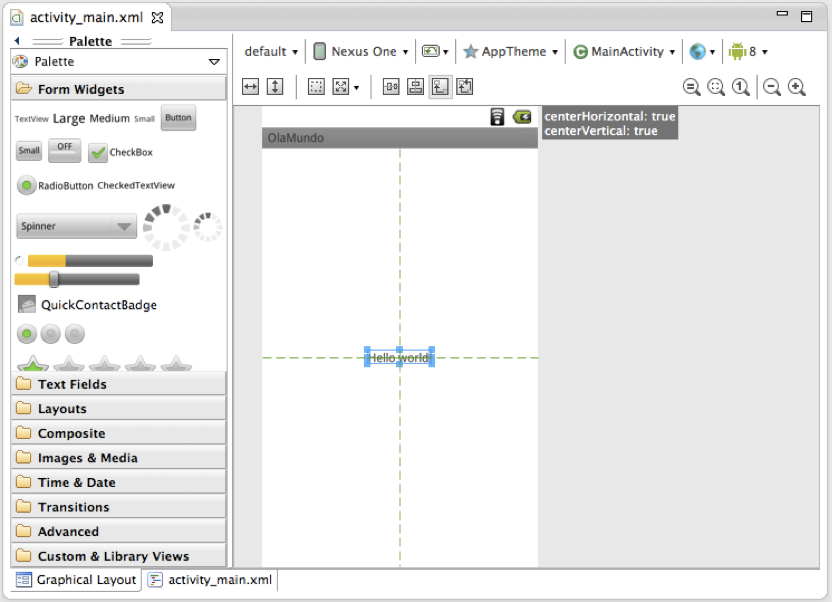
Nessa visualização é apresentado a estrutura do projeto (a esquerda), as classes e arquivos abertos (no centro) e o **Outline** (com a estrutura dos componentes) e o menu de **Properties** (Propriedades) (a direita).

Na estrutura do projeto temos:

[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-10.png)

* **src** - pacote de códigos fonte;
* **res** - pacote de recursos;
* **res → drawable-\_\_dpi** - pacote com as imagens de acordo com a resolução;
* **res → layout** - arquivos .xml com a estrutura do layout das páginas;
* **res → values** - arquivo **strings.xml** com os textos strings da aplicação e o **styles.xml** com o estilo de páginas da aplicação.

A edicão de uma tela pode ser feita de modo visual (Graphical Layout) adicionando os componentes através da paleta e configurando pelas propriedades ou editando o arquivo XML (activity\_main.xml).

[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-11.png)

Vamos alterar na versão XML o tamanho e posição do texto, clique em **activity\_main.xml** e altere o código do TextView:

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="100dp"

android:gravity="center\_horizontal"

android:text="@string/hello\_world"

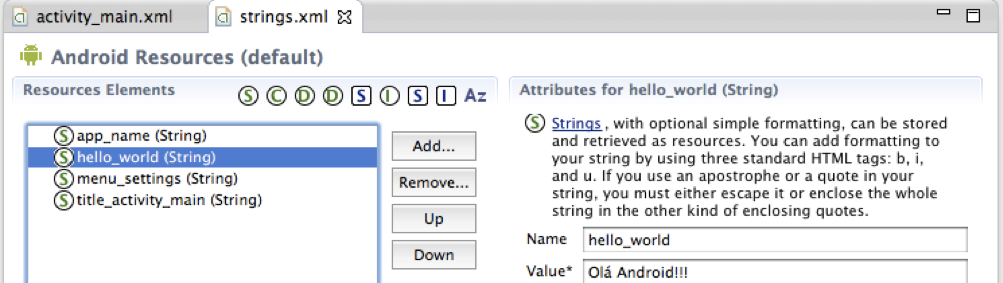
android:textSize="40dp"

tools:context=".MainActivity" />

O TextView representa um texto na tela, como um label, no qual pode ser definido o seu texto, como será o seu layout, tamanho do texto, cor do texto, entre outros.

Observação: Durante o desenvolvimento é costume verificar como o layout da página está ficando, para isso altere para a aba Graphical Layout (Layout Gráfico).

Esse TextView está utilizando uma string chamada **hello\_world**, para alterar o conteúdo desse texto, no menu **res → values** abra o arquivo **strings.xml** e altere o conteúdo da string hello\_world para **Olá Android!!!**

[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-12.png)

Vamos adicionar uma imagem nessa tela, copie a figura android.png para as pastas drawable-hdpi, drawable-ldpi, drawable-mdpi e drawable-xhdpi, após a declaração do TextView, adicione a declaração do ImageView que fará uso da imagem do android.png:

<ImageView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:src="@drawable/android"

android:contentDescription="@string/hello\_world" />

Após adicionar a figura nas pastas drawable, o Android gera automaticamente um registro para “@drawable/android”. Como queremos deixar a imagem embaixo do texto, vamos alterar o layout padrão para usar o LinearLayout, no qual podemos especificar a orientação que desejamos apresentar os componentes:

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:orientation="vertical"

android:layout\_width="fill\_parent"

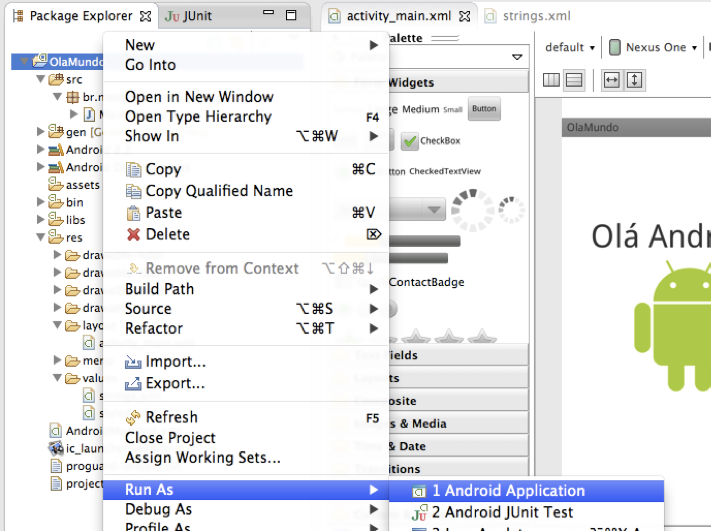
android:layout\_height="fill\_parent" >

<TextView ... />

<ImageView ... />

</LinearLayout>

Agora o layout ficou mais agradável, apresentando o texto em cima da figura. Para executar a aplicação, clique com o botão direito do mouse em cima do projeto e selecione **Run As (Executar Como) → Android Application**.

[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-13.png)

Quando o simulador terminar de carregar, desbloqueie a tela e a aplicação será automaticamente carregada:

[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-14.png)

Observação: O simulador do Android demora um pouco para carregar, portanto após inicia-lo deixe o aberto enquanto você estiver desenvolvimento.

Vamos adicionar uma mensagem “Olá!” quando a imagem do Android for clicada, para isso altere no arquivo activity\_main.xml a código xml da declarada da ImageView para:

<ImageView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:src="@drawable/android"

android:contentDescription="@string/hello\_world"

android:clickable="true"

android:onClick="mostrarMensagem"/>

Foi adicionado a propriedade **clickable=”true”** informando que essa imagem pode ser clicada e também a propriedade **onClick=”mostrarMensagem”** informando que ao clicar na imagem será chamado o método mostrarMensagem.

Abra a classe **br.metodista.ads5.olamundo.MainActivity**, essa é a Activity responsável pela tela inicial da aplicação, nela adicione o método mostrarMensagem:

**public** **void** **mostrarMensagem**(View view**)** **{**

Toast toast **=** Toast**.**makeText**(this,** "Olá!"**,** Toast**.**LENGTH\_SHORT**);**

toast**.**show**();**

**}**

O **Toast** é uma mensagem que aparece e desaparece da tela e o **Toast.LENGTH\_SHORT** especifica que é uma mensagem que fica por um tempo curto na tela.

O código completo da classe MainActivity ficará assim:

**import** android.os.Bundle**;**

**import** android.app.Activity**;**

**import** android.view.Menu**;**

**import** android.widget.Toast**;**

**public** **class** **MainActivity** **extends** Activity **{**

**@Override**

**public** **void** onCreate**(**Bundle savedInstanceState**)** **{**

**super.**onCreate**(**savedInstanceState**);**

setContentView**(**R**.**layout**.**activity\_main**);**

**}**

**public** **void** mostrarMensagem**(**View view**)** **{**

Toast toast **=** Toast**.**makeText**(this,** "Olá!"**,** Toast**.**LENGTH\_SHORT**);**

toast**.**show**();**

**}**

**}**

Execute novamente a aplicação no emulador do Android e clique sobre a figura para visualizar a mensagem:

[](http://www.universidadejava.com.br/images/2013-03-13-android-helloworld-textview-imageview-toast-15.png)