PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS

Hello World

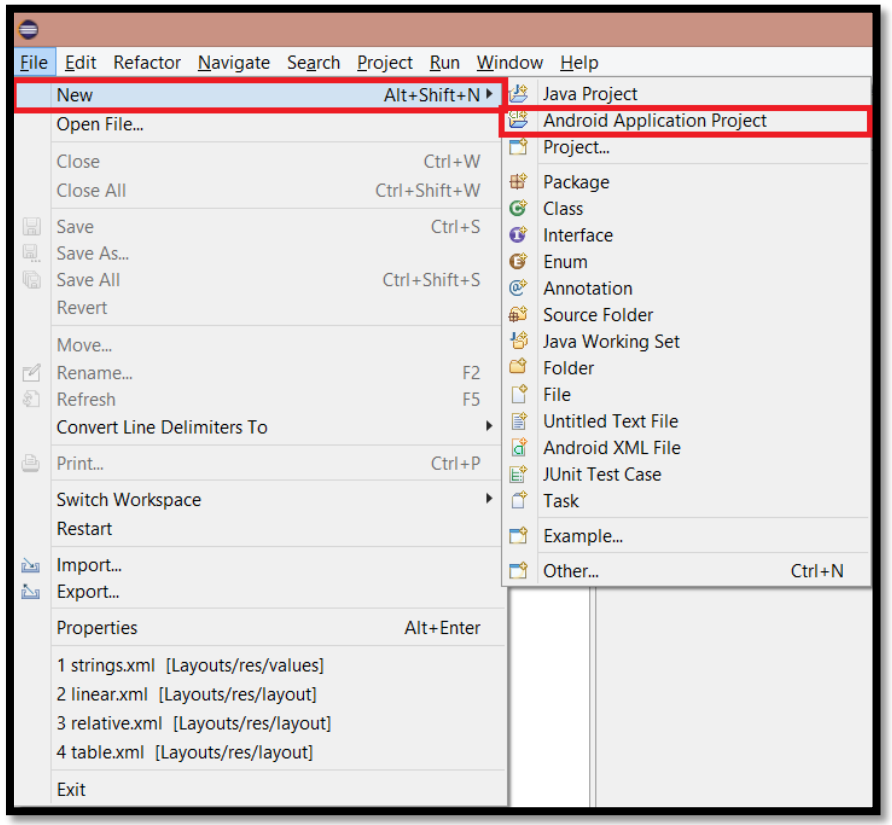
Professor: Danilo Giacobo

OBJETIVOS DA AULA

- Desenvolver o primeiro aplicativo Android
- Se familiarizar com o ambiente de desenvolvimento
- Conhecer os principais arquivos de um projeto Android
- Testar o emulador de dispositivos Android

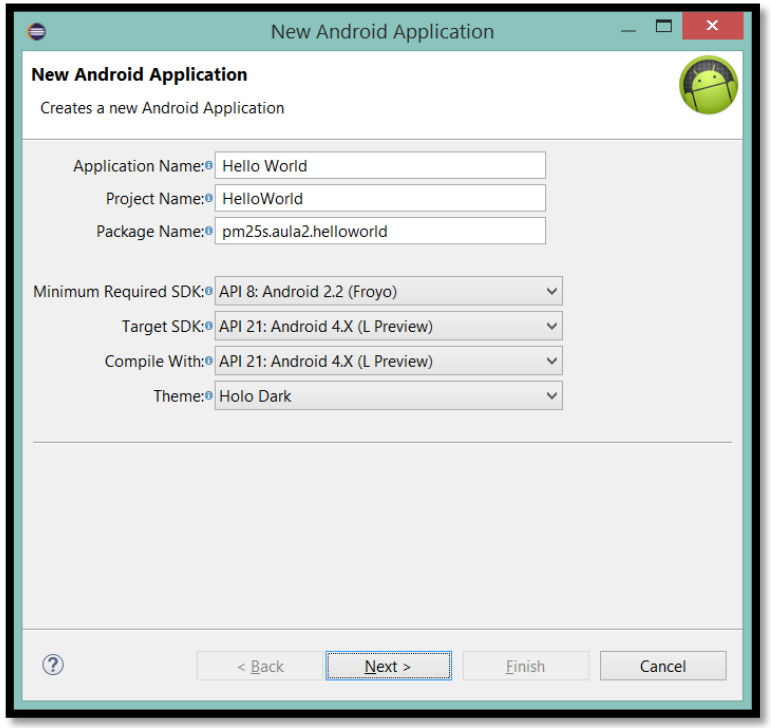
CRIANDO A PRIMEIRA APLICAÇÃO ANDROID

- Dentro do Eclipse acesse o menu **File > New > Android Application Project**



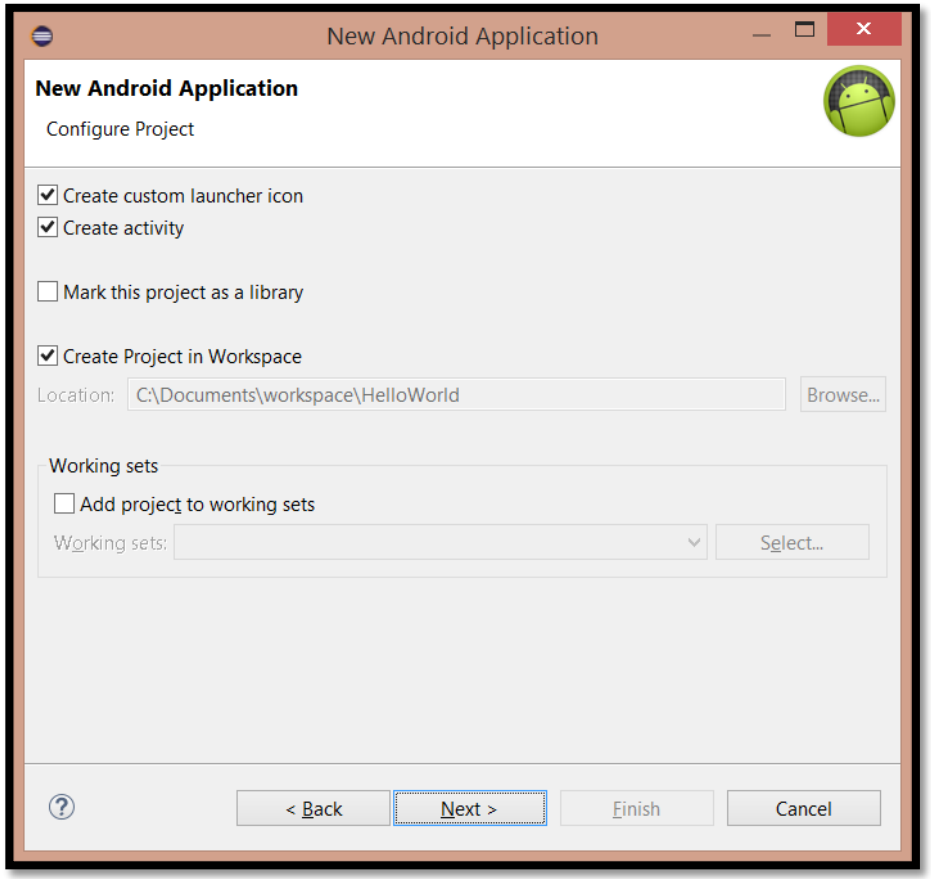
CONFIGURAÇÕES INICIAIS

- Na próxima tela você deverá informar os dados do seu novo projeto. Depois de terminado clique em **Next >**.



CONFIGURAÇÕES INICIAIS

- Na próxima tela você irá definir a criação de um ícone personalizado para a sua aplicação bem como a criação de uma Activity (classe que identifica uma tela da aplicação Android). Clique em **Next >**.

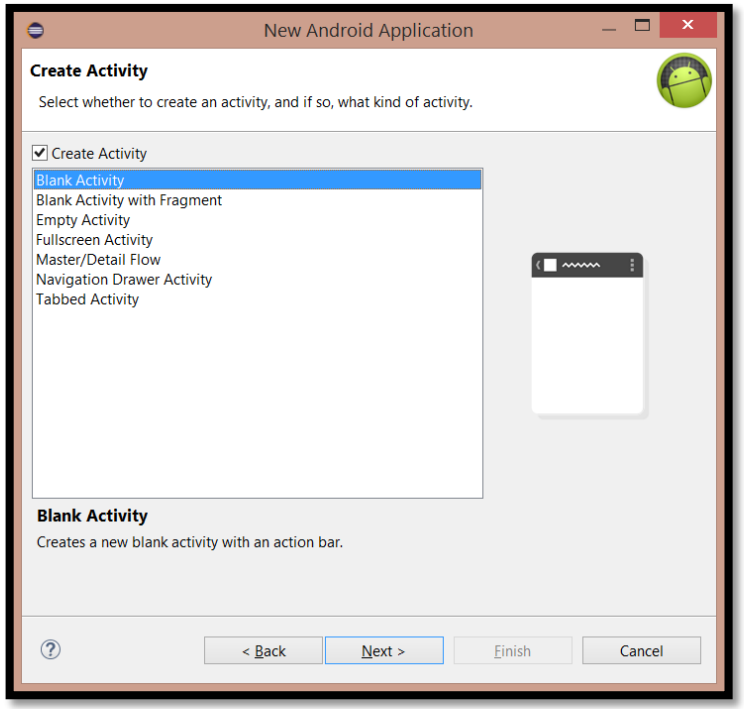


CONFIGURANDO O ÍCONE DA APLICAÇÃO

- É possível usar como ícone da sua aplicação uma imagem do computador, uma imagem do tipo Clip Art ou uma representação textual. É recomendável usar como tipo de imagem um PNG (Portable Network Graphic).

SELECIONANDO O TIPO DE ACTIVITY

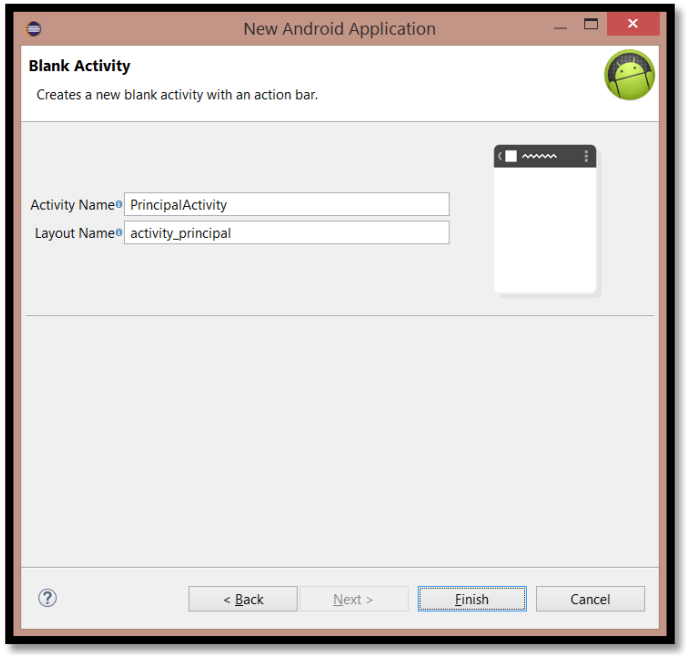
- A próxima tela é referente ao *template* para a *Activity*.
- Selecione a opção Blank Activity.
- Clique em **Next >**.



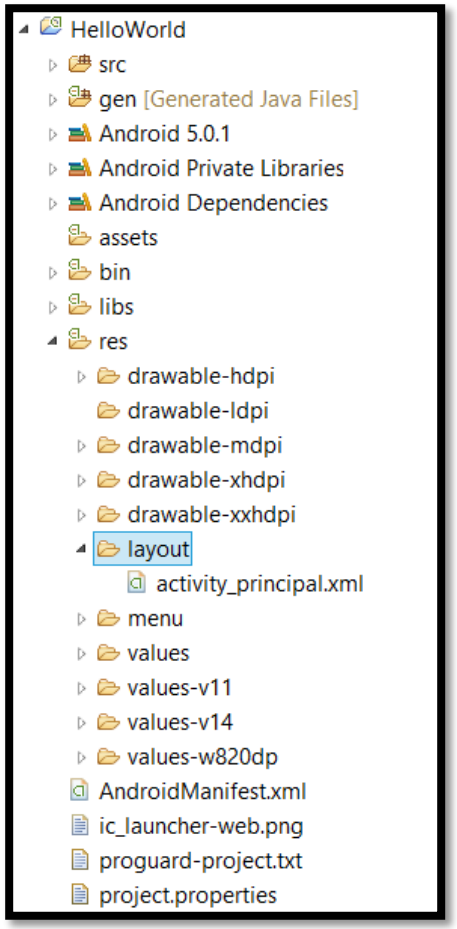
Importante!
Alguns tipos de **Activity** são incompatíveis com as versões anteriores do Android.

INFORMANDO O NOME DA ACTIVITY E DA TELA

- Na última tela é necessário informar o nome da classe da Activity, o nome do arquivo XML da tela e o tipo de navegação entre as telas. Clique em **Finish** para finalmente criar o projeto!



ESTRUTURA DO PROJETO NO ECLIPSE



src: pasta com os códigos-fontes do aplicativo Android. O arquivo **.java** é criado pelo Eclipse. Ele estende a classe *Activity* e representa uma tela do seu aplicativo. Ela é responsável também pelo tratamento de eventos.

gen: pasta que contém o arquivo **R.java**. Ele é um arquivo de referência. Recomenda-se não alterar o mesmo. Ele faz o elo entre a interface gráfica (XML) e o código-fonte (Java).

res: pasta que contém os recursos do projeto, principalmente os objetos gráficos.

assets: pasta que contém os recursos estáticos do projeto (que não dependem da densidade ou do tamanho do display).

libs: pasta com as bibliotecas externas utilizadas pelo aplicativo.

AndroidManifest.xml: "coração" de um aplicativo Android. Há informações nele como a versão do aplicativo, versão mínima do sistema operacional, permissões de uso, uso de bibliotecas externas, entre outras.

INTERFACE GRÁFICA DO APLICATIVO

- O arquivo XML possui a interface gráfica da aplicação.
- A *tag* **RelativeLayout** representa o gerenciador de layout.

```
1  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:tools="http://schemas.android.com/tools"
3      android:layout_width="match_parent"
4      android:layout_height="match_parent"
5      android:paddingBottom="@dimen/activity_vertical_margin"
6      android:paddingLeft="@dimen/activity_horizontal_margin"
7      android:paddingRight="@dimen/activity_horizontal_margin"
8      android:paddingTop="@dimen/activity_vertical_margin"
9      tools:context="pm25s.aula2.helloworld.PrincipalActivity" >
10
11     <TextView
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="@string/hello_world" />
15
16 </RelativeLayout>
```

activity_principal.xml

CLASSE PRINCIPAL DO APLICATIVO ANDROID

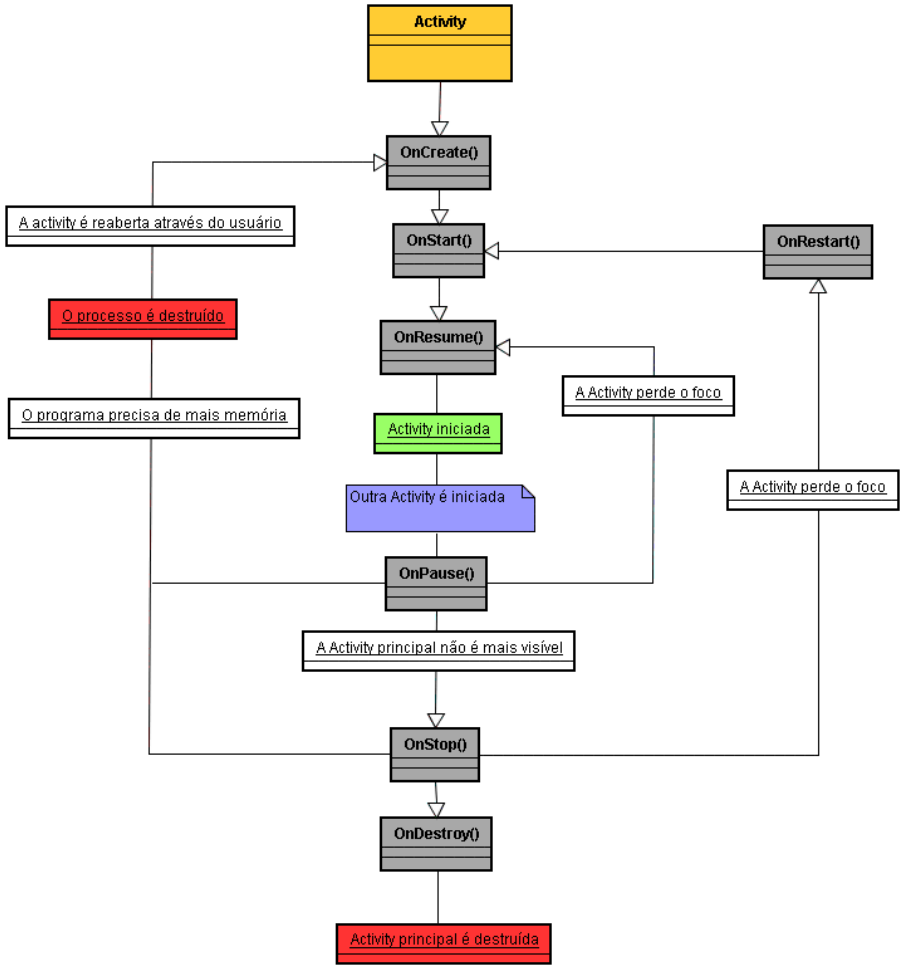
```java
package pm25s.aula2.helloworld;

import android.support.v7.app.ActionBarActivity;

public class PrincipalActivity extends ActionBarActivity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.principal, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

PrincipalActivity.java

CICLO DE VIDA DA ACTIVITY

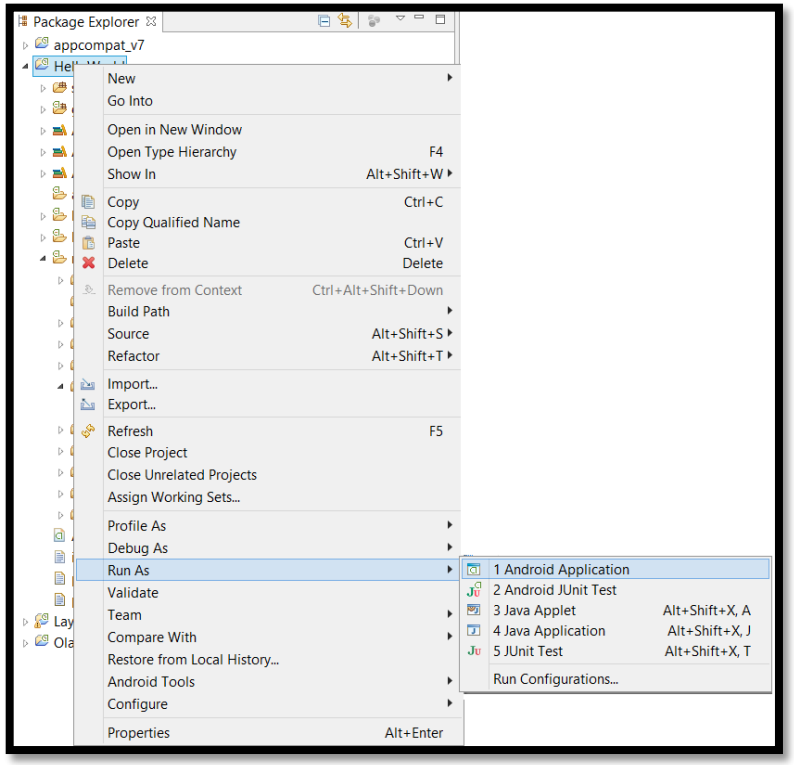
- Quando uma tela é acionada, o primeiro método a ser chamado é o *onCreate*.
- Posteriormente, segue-se a ordem *onStart* e *onResume*.
- Na chamada deste último método a tela já estará disponível para o usuário.
- Se a tela ficar em segundo plano o método *onPause* é acionado.
- Se a Activity perder totalmente o foco, o método *onStop* será chamado depois de *onPause*.
- Se a tela estiver apenas pausada o método *onResume* será chamado.
- Se ela estiver em *onStop* o caminho será *onRestart*, *onStart*, *onResume* .
- O sistema operacional Android pode destruir a mesma se esta estiver com recursos extremamente escassos (memória, por exemplo).
- Em *onStop* ela passa para *onDestroy* para finalmente ser destruída.

CICLO DE VIDA DA ACTIVITY



EXECUTANDO UM PROJETO NO ECLIPSE

- Para a primeira execução do seu aplicativo Android clique com o botão direito sobre o nome do projeto e selecione a opção **Run As > Android Application.**



Dica!

Após a primeira execução do seu aplicativo você poderá usar o botão Run existente na barra de ferramentas da IDE Eclipse para rapidamente executar seu projeto (apenas se o código Java estiver presente no editor de códigos).

HELLO WORLD EM EXECUÇÃO NO EMULADOR!