



Android e JUnit

Trabalhando com as Unidades de Teste

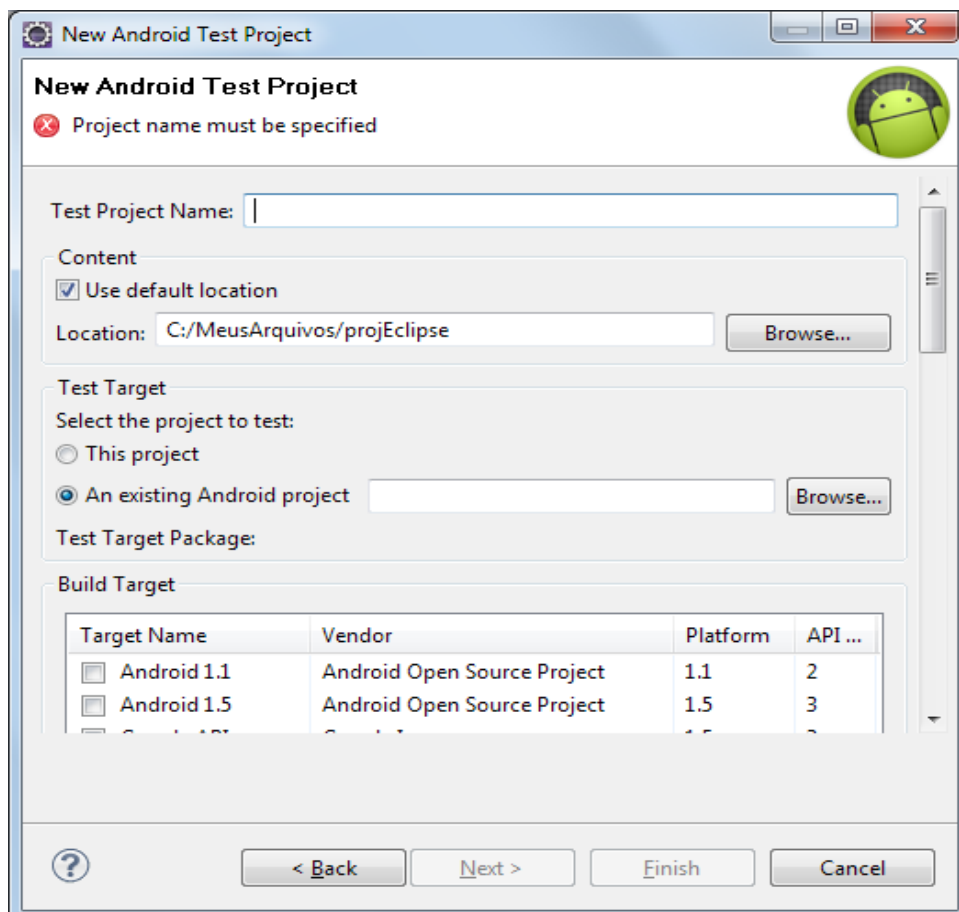
Para criar os códigos de Padrão Internacional, é necessário que a aplicação desenvolvida contenha zero erros, isso é conseguido utilizando as Unit Tests (Unidades de Teste). O JUnit já é bem conhecido por programadores Java e neste exemplo criaremos uma aplicação simples com o objetivo de testá-la.

1. Criar o Projeto de Teste

Para este tutorial, foi criado um projeto chamado *Exemplo*, sendo da versão 1.6, com a atividade chamada *ExemploActivity* e com a seguinte codificação no layout principal:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TextView android:id="@+id/texto"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
    <Button android:id="@+id/botao"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
</LinearLayout>
```

Clicar com o botão direito neste projeto e selecionar *New | Other | Android | Android Test Project* e veremos a seguinte tela:



Definir as seguintes opções:

- **Test Project Name:** ExemploTeste
- **Test Target:** Marque a opção: *An existing Android project*, e selecione seu projeto (no nosso exemplo usarei um projeto chamado Exemplo)
- **Build Target:** Android 1.6
- **Application Name:** Exemplo de Teste
- **Package Name:** com.exemplo.test
- **Min SDK Version:** 4

Pressionar o botão *Finish* e um novo projeto será criado no Eclipse, chamado *ExemploTest*.

2. Criar a Classe de Teste

Para este novo projeto, criar uma nova classe (*New | Class*), com as seguintes opções:

- **Package:** com.exemplo.test
- **Name:** ExemploTest
- **Superclass:** android.test.ActivityInstrumentationTestCase2<ExemploActivity>

Uma vez criada esta classe, altere seu código para:

```
package com.exemplo.test;

import com.exemplo.ExemploActivity;
import com.exemplo.R;
import android.test.ActivityInstrumentationTestCase2;
import android.widget.Button;
import android.widget.TextView;

public class ExemploTest extends
    ActivityInstrumentationTestCase2<ExemploActivity> {

    private ExemploActivity mActivity;
    private TextView texto;
    private Button botao;

    public ExemploTest() {
        super("com.exemplo", ExemploActivity.class);
    }

    @Override
    protected void setUp() throws Exception {
        super.setUp();
        mActivity = getActivity();
        texto = (TextView) mActivity.findViewById(R.id.texto);
        botao = (Button) mActivity.findViewById(R.id.botao);
    }
}
```

Usamos o **construtor** para passar a classe Pai dois parâmetros, o pacote da classe que está sendo testada e uma instância da classe que será testada. O método **setUp()** realiza uma inicialização do estado da classe de teste. Este método é executado automaticamente antes de cada caso de teste.

Um **Método de Teste** é um método público, de retorno vazio (void), sem parâmetros, iniciado pela palavra *test*. Desta forma, o **Android Test Runner** (responsável por executar os testes) identificará quais métodos devem ser executados. Iremos agora criar o seguinte

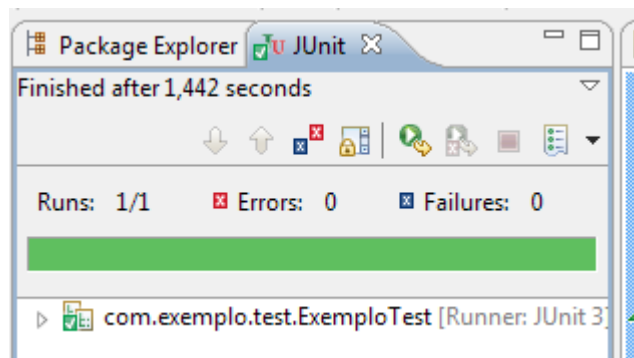
método, simplesmente para verificar se existem realmente os campos de tela:

```
public void testCamposTela() {  
    assertTrue(mActivity != null);  
    assertTrue(texto != null);  
    assertTrue(botao != null);  
}
```

Para executar as verificações, utilizamos os métodos do framework JUnit (recomendo que você veja mais sobre a documentação do JUnit no site <http://www.junit.org/>). O método *assertTrue*, verifica se o valor lógico passado é verdadeiro.

3. Executar o Teste

Para executar os testes, clicar com o botão direito no projeto e selecionar as opções *Run As | Android JUnit Test*. E obteremos como resultado no Eclipse a seguinte figura:



Indicando que o teste foi executado corretamente.