|  |  |  |
| --- | --- | --- |
|  | | **Departamento de Engenharia Informática**  **TeSP em Programação de Sistemas de Informação**  **Acesso Móvel a Sistemas de Informação**  2016/2017 |
|  |  |  |
|  |  |  |
| **Docentes** | Sérgio Lopes, sergio.lopes@ipleiria.pt | |
|  | David Safadinho, david.safadinho@ipleiria.pt | |

Ficha de Exercícios N/02

LinearLayout e componentes comuns

# Geral

Até ao momento foram desenvolvidos exercícios introdutórios, que tinham como único objetivo facilitar a utilização e proximidade do *Android Studio*. A partir da ficha número 2, os exercícios focam a aquisição e aplicação de conhecimentos práticos sobre o desenvolvimento de aplicações.

Como vimos na aula teórica, o desenho de interfaces gráficas é feito com recurso a componentes (widgets), layouts e composição destes elementos para atingir os objetivos visuais pretendidos. É assim possível combinar um conjunto de vistas/componentes, ou de layouts, na construção de interfaces mais elaboradas.

Por exemplo, um LinearLayout base pode conter outras instâncias de LinearLayout, permitindo a disposição na vertical de vários componentes e áreas de disposição horizontal

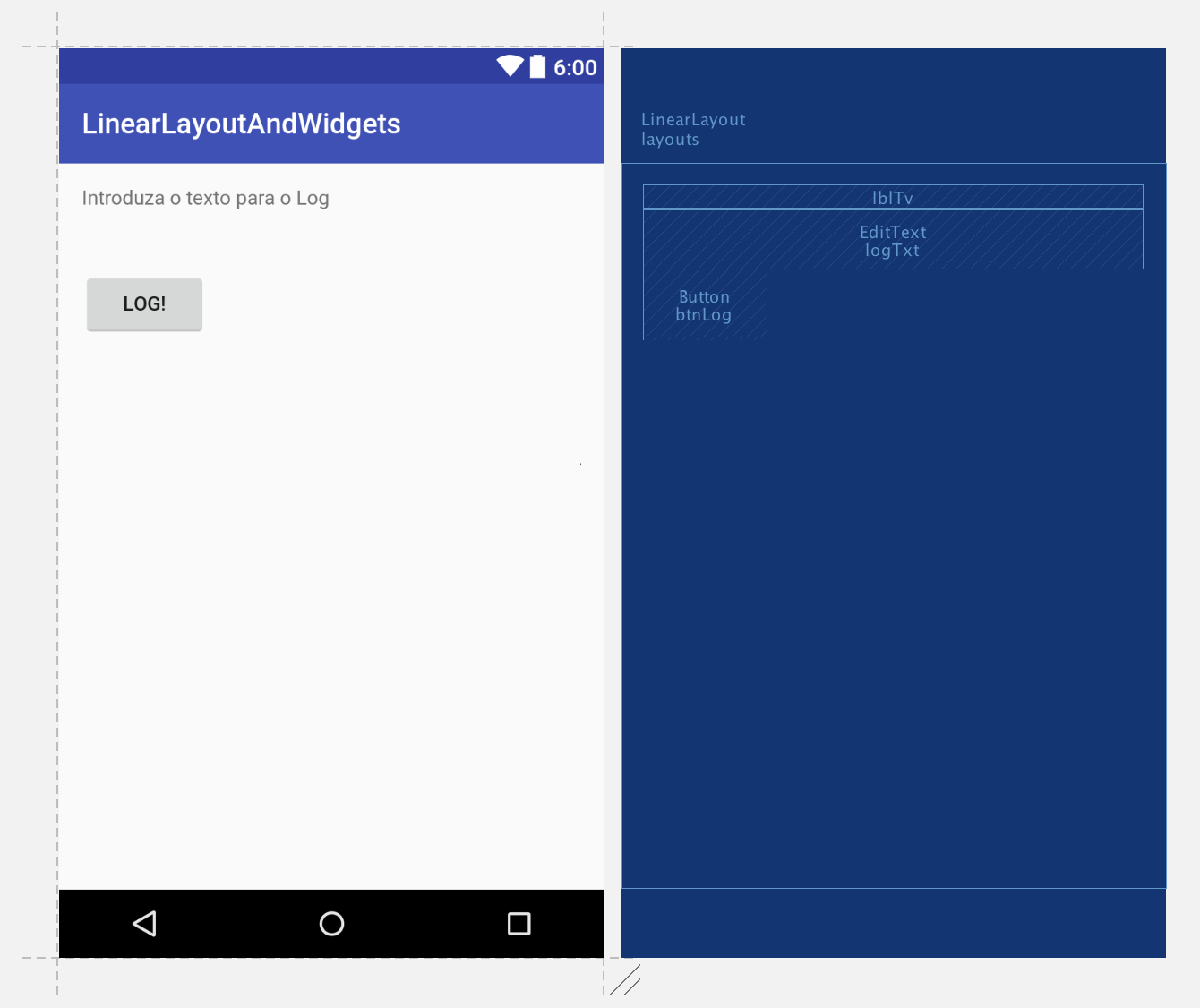
# Exercício 1 – Logs

Crie um projeto com as seguintes características:

* Nome do projeto: LinearLayoutAndWidgets
* SDK: API 19
* Modelo: Empty Ativity
* Nome da atividade: Widgets
* Nome do layout: widgets

Depois de criar o projeto, comece por desenhar a interface gráfica seguinte o modelo da imagem abaixo, tendo atenção à utilização/disposição dos componentes e layouts, e consultando os passos indicados como apoio ao desenvolvimento.

Figura 1 – Interface com componentes comuns.



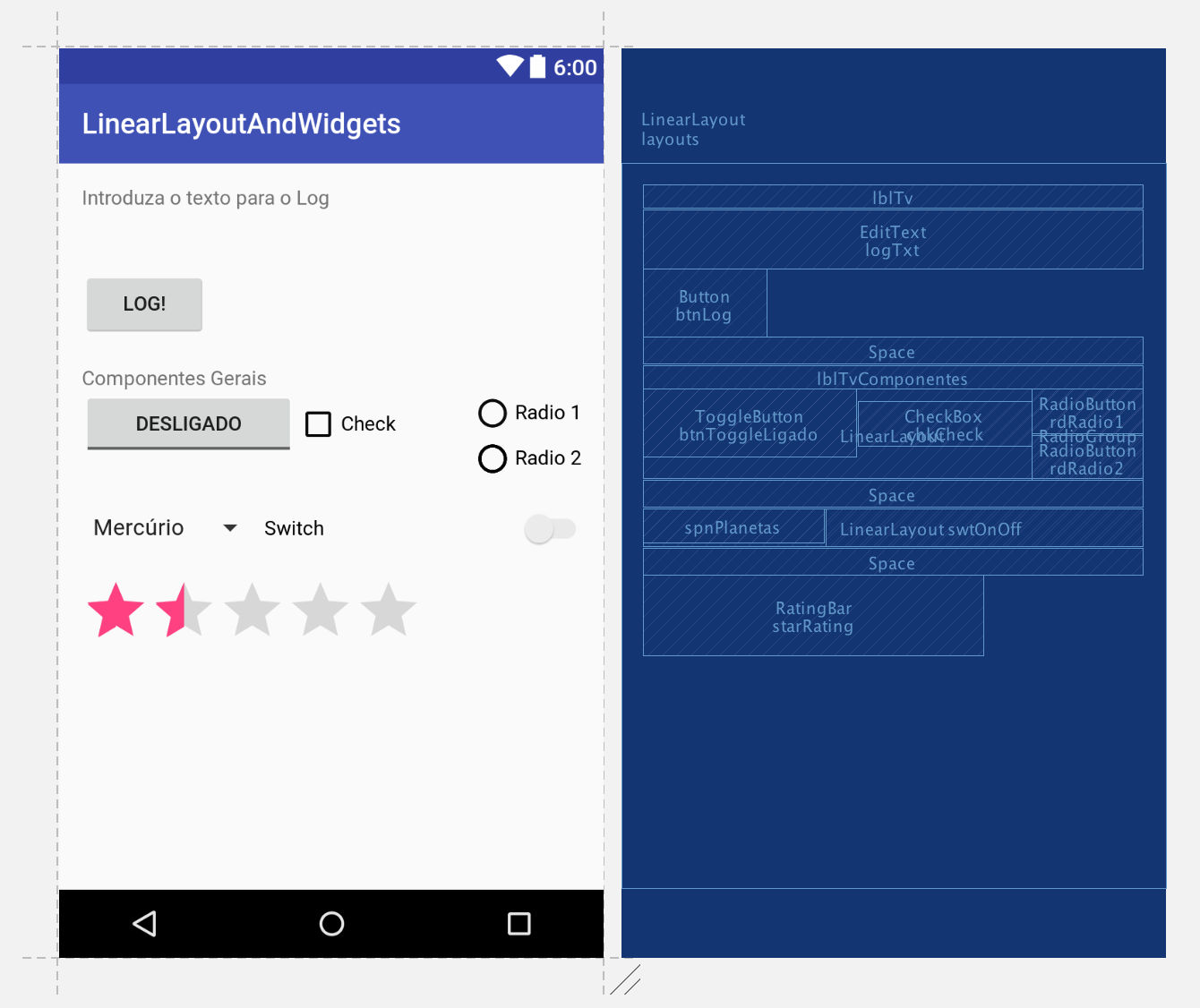
Passos:

1. Comece por alterar o **RelativeLayout** criado automaticamente para um **LinearLayout**. Corrija a orientação de modo a que os componentes fiquem dispostos verticalmente.
2. Adicione uma **TextView**, uma **EditView** e um **Button**.
3. Ao botão adicione um evento de modo a que, ao ser pressionado, sejam gerados *logs* com o texto introduzido na **TextView**:
   1. Utilizador insere texto na **TextView**;
   2. Utilizador clica no botão;
   3. O texto da **TextView** é impresso no terminal através dos métodos da classe *Log*;

# Exercício 2 – LinearLayout e Widgets

Partindo a atividade anterior. Adicione os componentes necessários de modo a obter a mesmo resultado que apresentado na figura seguinte. Dica: A disposição horizontal pode ser conseguida recorrendo a mais instâncias de *LinearLayout*.

Figura 2 - Componentes comuns.



Para cada um dos componentes implemente o evento que permite tratar a interação com o utilizador. Nessas implementações uso o método **Log.d()** para apresentar uma mensagem com o estado do componente.

Eventos:

* **ToggleButton**, **CheckBox**, **RadioButton** e **Switch**: *View.OnClickListener*
* **Spinner**: *OnItemSelectedListener*
* **RatingBar**: *OnRatingBarChangeListener*

# Exercício 3 – Anagrama

Um anagrama é uma palavra, ou frase, feita com as letras usadas noutra palavra, ou frase, colocadas por ordem diferente, ex.: litigad é um anagrama de digital. Podemos dizer que duas palavras são anagramas se todas as letras de uma existirem nas letras de outra e as duas palavras tiverem o mesmo tamanho.

Crie um projeto com as seguintes características:

* Nome do projeto: Anagrama
* SDK: API 19
* Modelo: Empty Ativity
* Nome da atividade: Anagrama
* Nome do layout: anagrama

Implemente um validador de anagramas que receba duas palavras (ou frases) e que diga ao utilizador se as palavras introduzidas são, ou não, anagramas. A aplicação deve possuir um botão para remover as palavras introduzidas (reiniciar) e um botão para comparar as palavras introduzidas. O resultado de validação deve ser apresentado numa TextView, seguindo o exemplo da imagem seguinte.

Figura 3 - Validador de anagramas.

