

Docentes

Sérgio Lopes, sergio.lopes@ipleiria.pt
David Safadinho, david.safadinho@ipleiria.pt

Ficha de Exercícios N/05

Aplicação para gestão de livros

APP “Livros”

A aplicação “Livros” é uma aplicação para *Android* que permite a gestão de uma pequena biblioteca de livros. Esta aplicação apresenta ao utilizador um menu principal que dá acesso às opções de **Lista de Livros**, **Galeria** e **Novo Livro**.

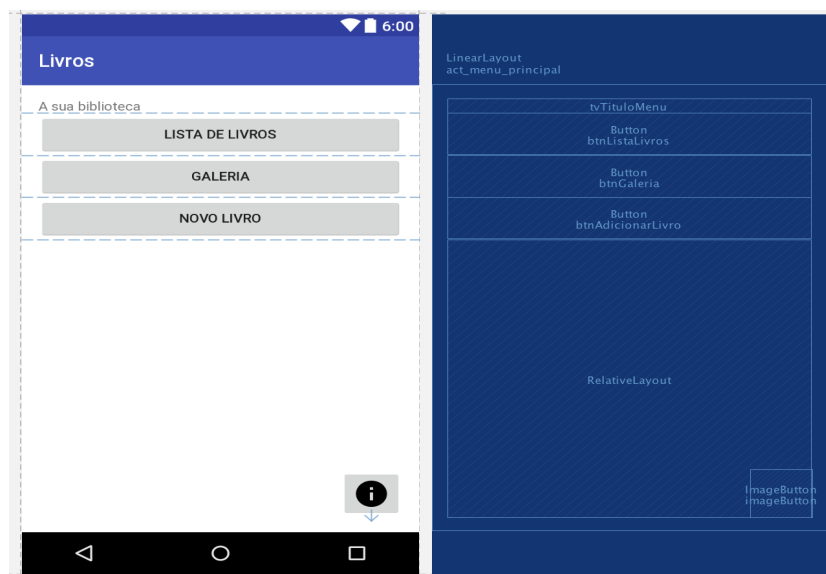
Desenvolva uma aplicação, seguindo o layout geral apresentado nas imagens seguintes. Tendo o cuidado de manter todos os textos no ficheiro strings.xml, cores usadas no ficheiro colors.xml e tamanhos/margens no ficheiro dimens.xml.

Desenho da interface gráfica

Comece por adicionar uma atividade para criar o menu principal, esta deve ser a atividade inicial:

- Nome: MenuPrincipal
- Layout: ac_menu_principal
- Template: Empty Activity

O menu principal é conseguido usando um *LinearLayout* com orientação vertical como base, uma *TextView* para o título, três botões (*Button*) e um *ImageButton* para dar acesso à atividade com descrição da aplicação. O botão “Sobre” é colocado no fundo do ecrã usando um *RelativeLayout* e definido, no botão, as propriedades *layout_alignParentEnd* e *layout_alignParentBottom* com o valor **true**.



Ajuste o tamanho da letra usada no título e nos botões, as margens do layout base e as cores de fundo do layout base e dos botões, incluindo do botão “Sobre”. Corrija também o ficheiro styles.xml para usar as novas cores.

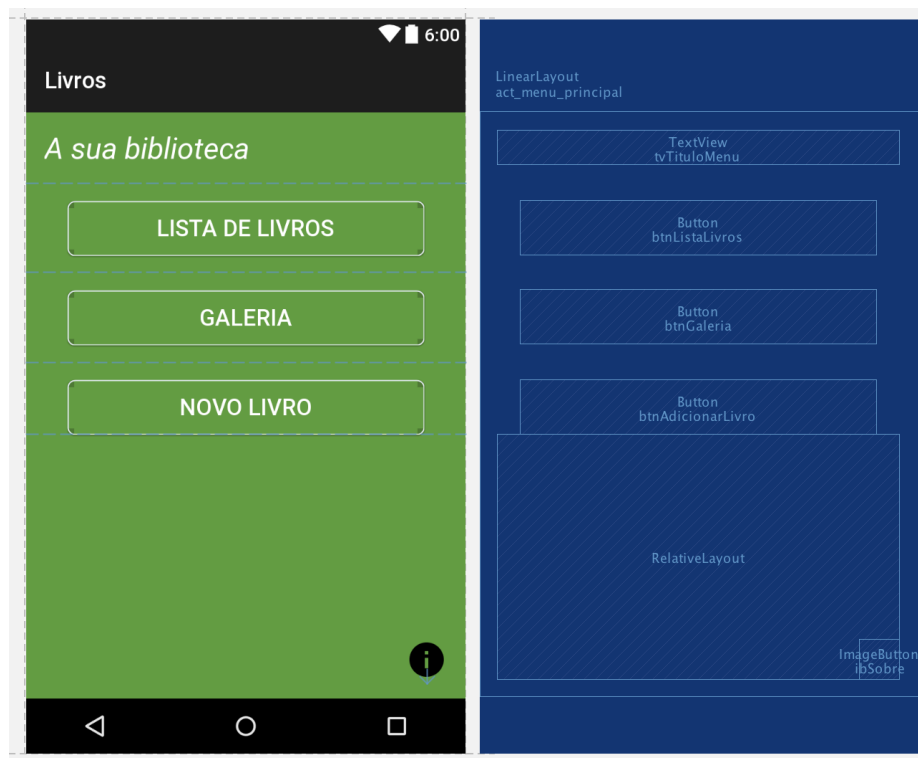
Valores:

- Cor
 - fundo: #639C42
 - letra: #FFFFFF
 - “accent”: #EFF4FF (usar em styles.xml)
 - escura principal: #1D1D1D (usar em styles.xml, colorPrimary e colorPrimaryDark)
- Tamanho da letra:
 - botões: 20sp
 - título: 26sp
- Margens
 - botões: 30dp (vertical), 20dp (horizontal)
 - título: 30dp (vertical)

Com as alterações acima os botões perdem-se no fundo e parecem apenas textos. Para alterar o aspeto destes botões vamos adicionar uma margem (cor “accent”), com 1dp de espessura e cantos arredondados com 6dp de raio. Para personalizarmos um botão vamos precisar de um ficheiro XML com as configurações a aplicar. Crie um novo recurso XML na pasta *drawable* com o nome **border_btn_menu**, com o seguinte código:

```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <corners android:radius="6dp"/>
    <stroke
        android:width="1dp"
        android:color="@color/accent"/>
</shape>
```

O menu principal está quase completo, falta adicionar os eventos aos botões de modo a que seja possível abrir as restantes atividades.



Eventos

Adicione as atividades em falta para que estas sejam adicionadas ao ficheiro `AndroidManifest.xml` automaticamente.

- Nomes: `ListaLivros`, `Galeria`, `AdicionarLivro` e `Sobre`
- Layouts: `ac_lista_livros`, `ac_galeria`, `ac_adicionar_livro` e `ac_sobre`
- Template: `Empty Activity`

Edite o ficheiro `AndroidManifest.xml` e adicione a cada *tag* **activity** (correspondente às quatro atividades existentes) a propriedade **android:label** que permite alterar o título da atividade que é apresentado pelo Android. Não se esqueça que os textos devem estar no ficheiro `strings.xml`.

No mesmo ficheiro, adicione as configurações de cada um dos *intents* necessário para a abertura das atividades, por exemplo (adapte e repita para as restantes atividades):

```
<intent-filter>
    <action android:name="pt.ipleiria.estg.livros.LISTA"></action>
    <category android:name="android.intent.category.DEFAULT"></category>
</intent-filter>
```

Adicione, a cada um dos botões do menu principal, um evento que inicie a atividade correspondente.

Dados

Os dados para a nossa aplicação são dados exemplo, e por isso as imagens serão acedidas através dos seus ID automáticos e a lista de livros será criada de forma estática/fixa numa classe auxiliar. Para ajudar no desenvolvimento futuro vamos já criar algumas classes de apoio.

Crie uma classe nova, de nome **Dados**, e coloque-a num *package* novo chamado *modelos* (nome do *package* a criar pode ser adicionado quando se cria a classe). Crie outra classe, no mesmo *package*, com nome **Livro** e com as propriedades:

- `titulo` (`String`)
- `serie` (`String`)
- `autor` (`String`)
- `ano` (`String`)
- `isbn13` (`String`)
- `sinopse` (`String`)
- `classificacao` (`float`)

Para esta classe gere, automaticamente, os getters e setters necessários (menu `Code` > `Generate ...` > `Getter and Setter`), crie também o método `toString()` que devolva apenas o título do livro.

Na classe `Dados` introduza o código necessário para ter alguns dados de exemplo (use os dados do ficheiro `book.txt` dentro do arquivo ZIP disponível):

```

public static final Integer[] capas = {
    R.drawable.amongthieves,
    R.drawable.childrenofdune,
    R.drawable.darknessatsethanon,
    R.drawable.dune,
    R.drawable.endersgame,
    R.drawable.magician,
    R.drawable.silverthorn,
    R.drawable.starbound,
    R.drawable.talesofheresy,
    R.drawable.thebeatingofhiswings,
    R.drawable.thelastfourthings,
    R.drawable.theflefthandofgod
};

private static List<Livro> livros;

public static List<Livro> getLivros() {
    if (livros == null) {
        livros = new ArrayList<>();
    }

    // livros.add()

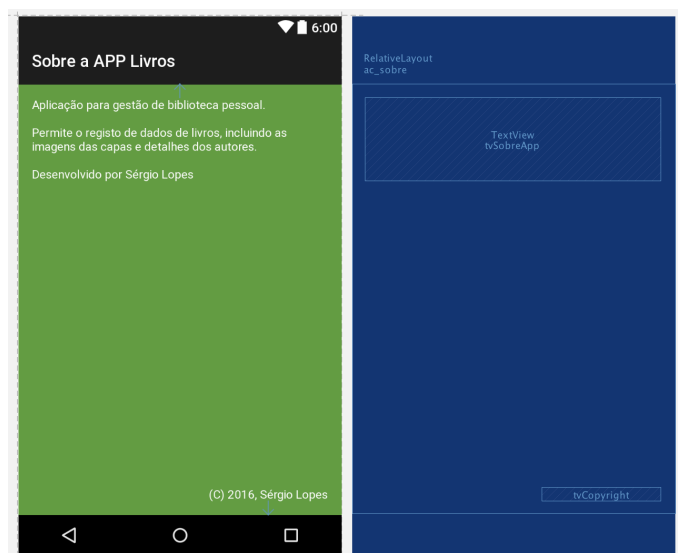
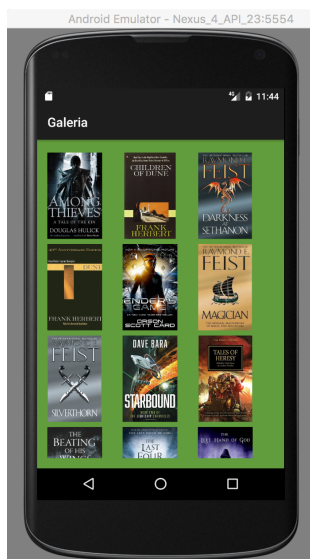
    return livros;
}

```

Galeria

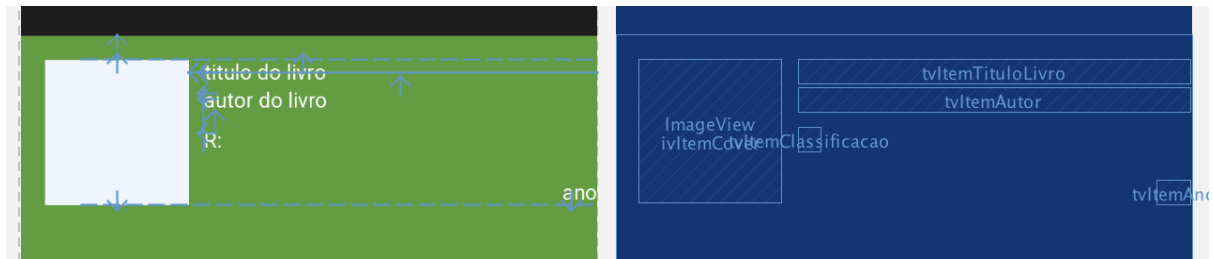
Recorrendo à implementação dos exercícios da ficha N/03 (ListView + GridView) implemente a classe **ImagemAdapter** que permita apresentar a lista de imagens/fotos de capa dos livros. Configure a atividade Galeria para apresentar 3 colunas com imagens de dimensão 170x285 (WxH). Adicione as imagens exemplo (ficheiro livros.zip) à pasta drawable e crie um vetor de Integer com os ID das várias imagens para simular a lista de capas.

Implemente também a atividade “Sobre”, com a descrição da aplicação e informação de copyright (substitua o nome do autor/copyright).



Item da lista de livros

A lista de livros deve ser implementada com uma *ListView*, mas ao contrário dos exemplos das fichas anteriores, iremos implementar um layout mais completo para os detalhes do livro recorrendo a um *RelativeLayout*. Para isso vamos precisar de adicionar um novo recurso XML, com nome **item_lista_livro**, e no qual devemos desenhar o seguinte layout:



Com o apoio do docente, implemente o *adapter* necessário para apresentar itens na lista.