

Docentes:

Sónia Luz, sonia.luz@ipleiria.pt
David Safadinho, david.safadinho@ipleiria.pt
Cátia Ledesma, catia.ledesma@ipleiria.pt

Ficha Prática 7

Books – Menus e Widgets

Objetivos da Ficha

- Criar e manipular menus
- Criar e manipular *FloatingActionButton*
- Utilizar widgets para manipulação e apresentação de informação

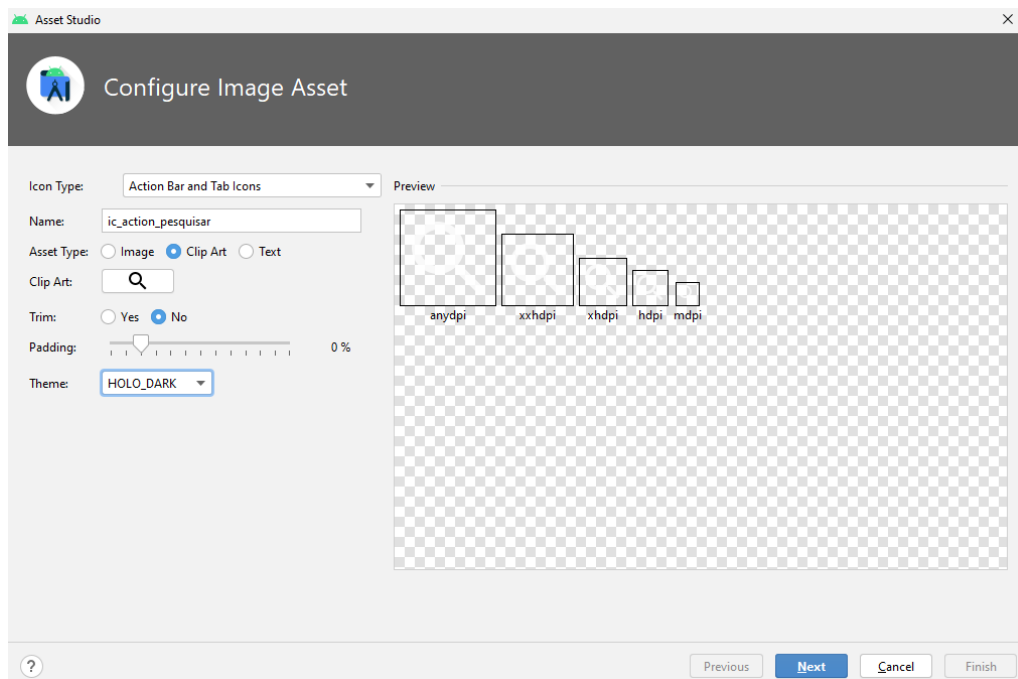
Introdução

Nesta ficha vamos continuar com a aplicação anterior. Caso queira manter o estado de evolução da aplicação, deve fazer uma cópia e trabalhar sobre a mesma.

Pretende-se salientar o uso de **Menus e Widgets**, para melhorar a apresentação e utilização da aplicação.

Abrir a pasta do projeto da aula anterior

1. Na atividade **MenuMainActivity**, utilize as **sharedPreferences** para guardar os dados do email do utilizador.
2. Para poder criar menus com ícones, deve adicionar um novo **Image Asset**, para cada ícone, de acordo com a imagem seguinte. Esses ícones irão representar as funcionalidades para **Pesquisar, Adicionar, Remover e Guardar** um livro;
 - 2.1. Deve clicar com o botão direito do rato na pasta **drawable** e escolher a opção **Image Asset**:

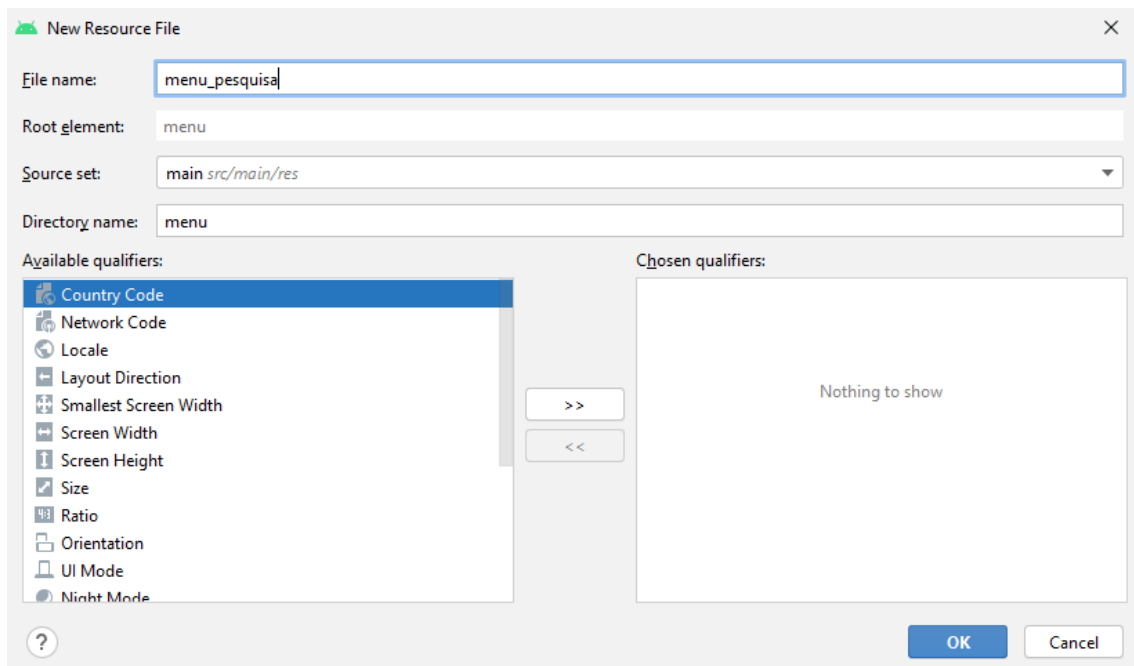


3. Para permitir adicionar um novo livro a partir dos fragmentos **ListaLivrosFragment** e **GrelhaLivrosFragment**, deve adicionar um botão do tipo **FloatingActionButton**;

- 3.1. Adicione um **FloatingActionButton**, em cada fragmento, com as características descritas na imagem seguinte:

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fabLista"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|end"
    android:layout_margin="16dp"
    android:clickable="true"
    android:src="@drawable/ic_action_adicionar"
    tools:ignore="SpeakableTextPresentCheck" />
```

- 3.2. Para criar a ação do **FloatingActionButton**, crie no método **onCreateView** o listener **setOnClickListener** e redirecione o utilizador para a atividade **DetalhesLivroActivity**.
4. De seguida deve criar um menu de opções **menu_pesquisa.xml** através de um **"Menu Resource File"** (selecionando com o botão direito do rato em cima da pasta **res/menu** e escolher esta opção), tal como pode ver na imagem seguinte:



4.1. Este menu deve conter o item: **itemPesquisa** com as características apresentadas na imagem seguinte:

```
<item
    android:id="@+id/itemPesquisa"
    android:icon="@drawable/ic_action_pesquisar"
    android:title="@string/menuSearch"
    app:actionViewClass="android.widget.SearchView"
    app:showAsAction="always"/>
```

5. No fragmento **ListaLivrosFragment** deve associar o menu de opções;

5.1. Para que o fragmento permita mostrar um menu de opções, é necessário adicionar a seguinte linha de código no método **onCreateView**:

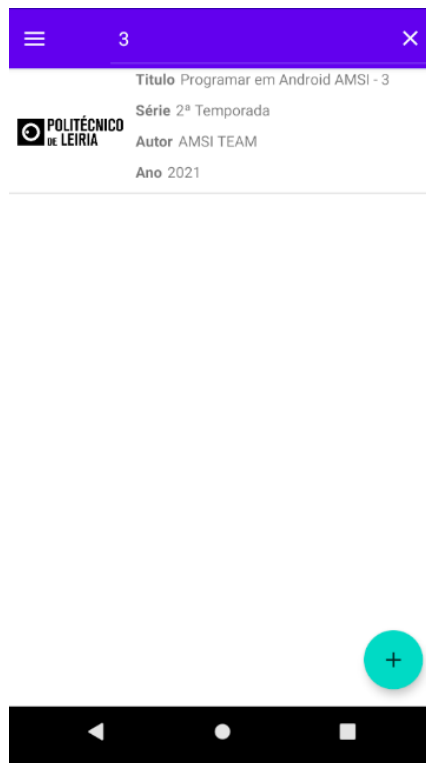
```
setHasOptionsMenu(true);
```

5.2. Deve reimplementar o método **onCreateOptionsMenu()** para apresentar o recurso de menu **menu_pesquisa.xml**;

5.3. Após a implementação do método crie um atributo do tipo **SearchView**;

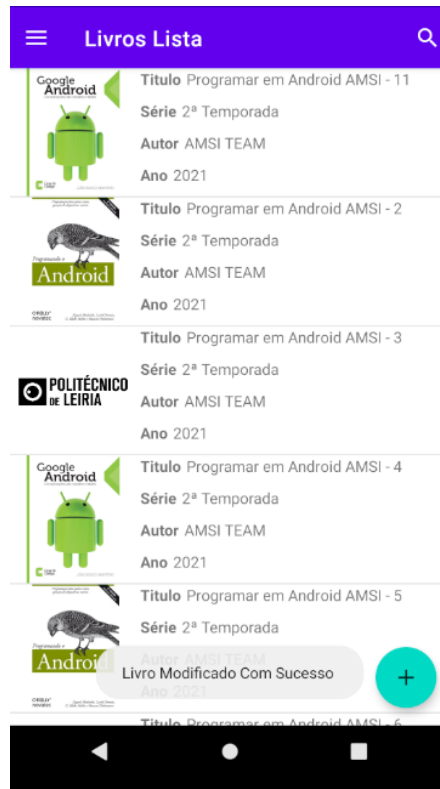
5.4. No método **onCreateOptionsMenu()** vamos criar uma pesquisa em *runtime* recorrendo ao *listener* **OnQueryTextListener()** através da implementação do método **setOnQueryTextListener()**;

5.5. Agora é necessário implementar código no método **onQueryTextChanged()**, de forma a filtrar os livros. O resultado esperado será semelhante ao apresentado na figura seguinte.

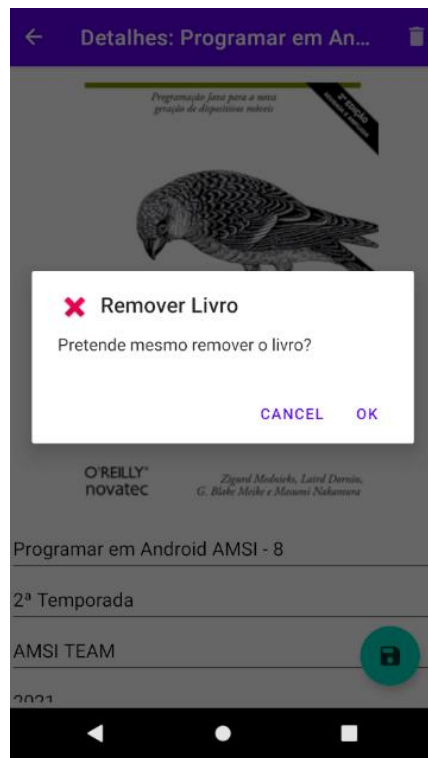


6. Deve criar um menu de opções **menu_detalhes_livro.xml** que irá conter o item **itemRemover**;
 - 6.1. Este menu deve estar associado à atividade **DetalhesLivroActivity** para permitir remover um livro existente.
7. Na atividade **DetalhesLivroActivity** deve adicionar um novo **FloatingActionButton** semelhante ao adicionado anteriormente;
 - 7.1. Altere o layout da atividade para **CoordinatorLayout**;
 - 7.2. Adicione o componente **ScrollView**;
 - 7.2.1. Deve arrastar para dentro do **LinearLayout** todos os componentes existentes;
 - 7.3. Para finalizar adicione um **FloatingActionButton**.
8. Na classe **Livro**, altere o construtor de modo que o **id** do livro seja único e incrementado de forma automática e sequencial.
9. Na classe **SingletonGestorLivros** deve implementar os métodos **adicionarLivro()**, **removerLivro()** e **editarLivro()**.
10. Na atividade **DetalhesLivroActivity**, quando não recebe nenhum **id de livro**, através do *bundle Extra* do *intent*, deve implementar a funcionalidade para adicionar um novo livro, caso contrário a atividade deve permitir editar os detalhes do livro apresentado;

- 10.1. O **FloatingActionButton** existente deve adequar o seu **icon** à tarefa de adicionar um livro ou gravar os dados editados e além disso deve implementar as respetivas funcionalidades.
- 10.2. Alterar os **Intents** existentes em **ListaLivroFragment** e **GrelhaLivrosFragment**, de forma que quando finalizamos a inserção ou edição de um livro, ser apresentado num **Toast** a informação sobre a ação executada, tal como na imagem seguinte:



- 10.2.1. Para implementar a funcionalidade, deve substituir o **startActivity** pelo **startActivityForResult** onde vai colocar um código da ação e depois efetuar uma reimplementação do método **onActivityResult()**.
11. Na atividade **DetalhesLivroActivity** deve implementar a funcionalidade “remover livro”;
- 11.1. Quando é recebido um **id de livro**, através do *bundle Extra* do *intent*, o menu **menu_detalhes_livro.xml** deve estar disponível;
- 11.2. Deve reimplementar o método **onOptionsItemSelected()**, que permite detetar qual o item do menu selecionado;
- 11.3. Para a implementação da funcionalidade remover, deve usar o componente **AlertDialog**, que permite confirmar junto do utilizador a sua intenção, tal como apresentado na imagem seguinte:



12. Para melhorar a forma de interação com a aplicação deve investigar:

- 12.1. O uso do **widget SwipeRefreshLayout** para sincronizar os dados a apresentar na lista e na grelha;
- 12.2. A substituição dos componentes **Toast** por componentes do tipo **Snackbar** para apresentação de mensagens, tal como podem ver na imagem seguinte;

