

Docentes:

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

Ficha Prática 4

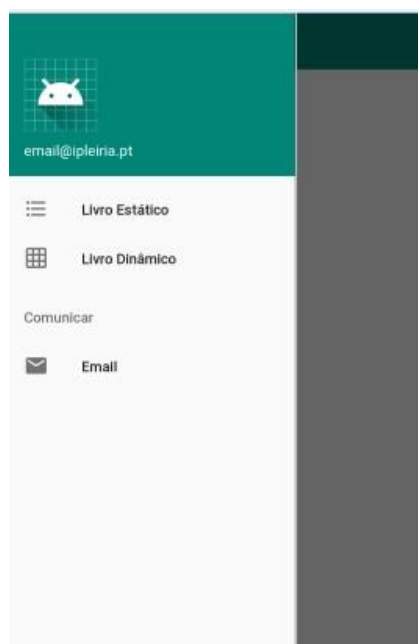
Books – Navegação e Fragmentos

Objetivos da Ficha

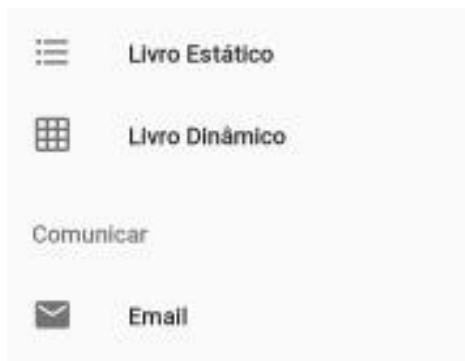
- Alterar o projeto da aula anterior “Books” de forma a ter um menu de navegação
- Inserir e manipular componentes visuais: NavigationView e AppBar
- Criar fragmentos

Introdução

Nesta ficha vamos criar uma aplicação que permita ao utilizador navegar pelos vários itens de um menu lateral. A janela principal da aplicação “**Books**” terá um aspeto semelhante ao apresentado na figura seguinte:



1. Dando continuidade ao projeto da aula anterior - Books, crie um nova atividade denominada **MenuMainActivity**, que irá substituir futuramente a *MainActivity*.
2. A próxima etapa será definir os itens do menu de navegação, para isso é necessário criar uma pasta que irá conter os itens existentes no menu da aplicação. Ao selecionar a pasta *res*, faça *New→Android Resource Directory*, escolha um recurso do tipo **menu** e posteriormente crie um ficheiro, *New→ Menu Resource File*, com nome **activity_main_drawer**.



Os 2 primeiros itens do menu fazem parte de um **grupo**, podendo partilhar estado e visibilidade mas não podem ser clicáveis em simultâneo, para isso é necessário colocar o atributo:

```
android:checkableBehavior="single".
```

Para cada um dos itens vamos atribuir o id: **navEstatico**, **navDinamico**, **navEmail**, que posteriormente serão utilizados para identificar a opção selecionada.

- 2.1. Ao analisar o menu anterior vai verificar que não aparece posicionado corretamente, o objetivo é colocá-lo no **navigationView**. Para que isso aconteça altere a tag do menu principal:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:tools="http://schemas.android.com/tools"
      tools:showIn="navigation_view">
```

- 2.2. O menu anterior ainda não tem os ícones atrás de cada item, por isso deve criar as 3 imagens através da opção: *New→Image Asset*, e escolher um ícone do tipo *"Action Bar and Tab Icons"*. Posteriormente selecione uma imagem do *clipart* para associar ao item.

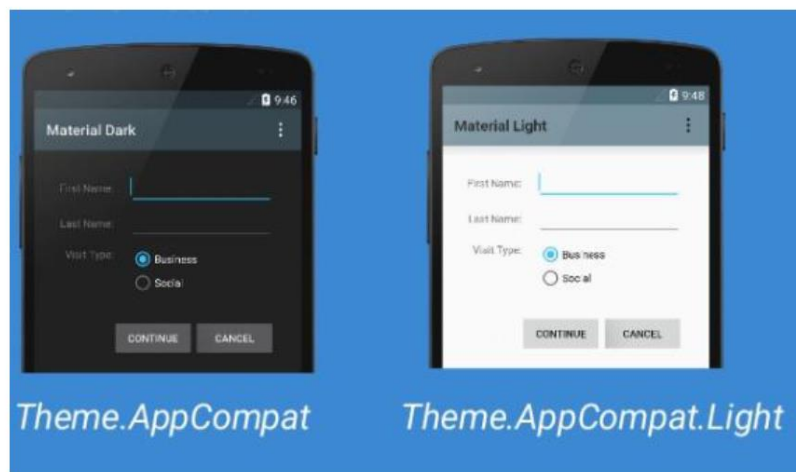
3. Para colocar a barra de ferramentas no layout da atividade *MenuMainActivity* é necessário um novo *widget* - **AppBarLayout** que pode ser incorporado na hierarquia do conteúdo principal.

- 3.1. Crie um novo layout denominado **app_bar_main** e altere de *LinearLayout* para *androidx.coordinatorlayout.widget.CoordinatorLayout*. Este layout deve ser composto por um *AppBarLayout* e um *FrameLayout* com o id **contentFragment**, onde será apresentado cada fragmento, de acordo com a seleção do menu.

- 3.2. Para que o *FrameLayout* não fique sobreposto é necessário adicionar a propriedade:

```
app:layout_behavior="@string/appbar_scrolling_view_behavior"
```

- 3.3. Para o *AppBarLayout* iremos usar o tema **ThemeOverlay.AppCompat.Dark.ActionBar**, que irá corresponder ao fundo escuro (imagem esquerda) e no *Toolbar* o tema **ThemeOverlay.AppCompat.Light**.



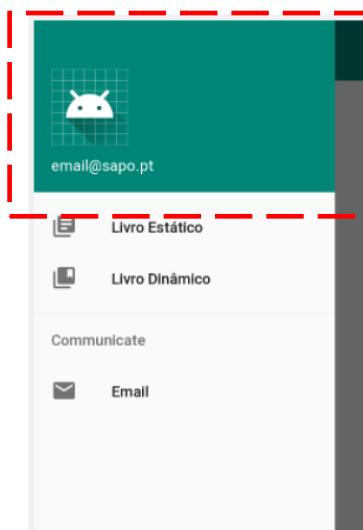
3.4. Para impedir que a aplicação use a classe *ActionBar* nativa, que fornece a barra de ferramentas, coloque na *MainMenuActivity*, declarada no *AndroidManifest*, o tema **NoActionBar** que deve herdar de **Theme.Books**.

```
<activity
    android:name=".MainMenuActivity"
    android:label="@string/app_name"
    android:launchMode="singleTop"
    android:screenOrientation="portrait"
    android:theme="@style/Theme.Books.NoActionBar" />
```

3.5. O tema utilizado na alínea anterior necessita de ser criado, para isso deve aceder à pasta *res* e no ficheiro *themes.xml* criar o tema **NoActionBar**:

```
<style name=" Theme.Books.NoActionBar">
    <item name="windowActionBar">false</item>
    <item name="windowNoTitle">true</item>
</style>
```

4. O *NavigationView* que irá ser criado posteriormente, é composto pelo menu (ponto 3) e por um cabeçalho. Para construir o cabeçalho crie um layout denominado **nav_header_main**, com aspeto similar ao da figura abaixo:



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="160dp"
    android:background="@color/purple_200"
    android:gravity="bottom"
    android:orientation="vertical"
    android:padding="16dp"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <!-- definir componentes em falta: ImageView e TextView(para o email) -->

</LinearLayout>
```

5. A etapa final do layout consiste em configurar o *Navigation Drawer* no ficheiro **activity_menu_main.xml**.

O *ConstraintLayout* deverá ser substituído pelo *androidx.drawerlayout.widget.DrawerLayout* com o id **drawerLayout** e na hierarquia será feita a inclusão do layout desenvolvido no ponto 4.1. usando a tag **include**.

Além disso também irá conter um **NavigationView**, com o id **navView**, que é composto pelo menu:

app:headerLayout="@layout/nav_header_main"

e um headerLayout: **app:menu="@menu/activity_main_drawer"**

O *NavigationView* ao ser clicado irá deslizar sobre o *ActionBar*.

```
<androidx.drawerlayout.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawerLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <!-- definir os componentes em falta -->

</androidx.drawerlayout.widget.DrawerLayout>
```

6. Já temos todos os layouts necessários para o menu de navegação, falta a codificação de forma a permitir que o utilizador interaja com os componentes.

6.1. Na *LoginActivity* altere o código do click do botão **Login**, para que inicie a atividade **MenuMainActivity**.

6.2. Na atividade *MenuMainActivity* vai ser necessário carregar o conteúdo do cabeçalho do menu de navegação, com o email recebido pela atividade anterior.

Coloque no método *onCreate* o código abaixo e crie o método **carregarCabecalho()**.

```
private NavigationView navigationView;
private DrawerLayout drawer;

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

    Toolbar toolbar = findViewById(R.id.toolbar);
```

```

setSupportActionBar(toolbar);
drawer = findViewById(R.id.drawerLayout);
navigationView = findViewById(R.id.navView);

ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(this, drawer,
toolbar, R.string.ndOpen, R.string.ndClose);
toggle.syncState();
drawer.addDrawerListener(toggle);

carregarCabecalho(); //TODO:criar método
}

```

6.3. Quando um item do menu é selecionado é invocado o método, **onNavigationItemSelectedListener**. Adicione um listener que tem como função “escutar” o que acontece a um objeto e “informar”. Nesta 1ª fase quando um item é selecionado iremos apresentar uma mensagem na consola e o *DrawerLayout* é fechado.

```

public class MainMenuActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
        //acrescentar a linha
        navigationView.setNavigationItemSelectedListener(this);
    }

    @Override
    public boolean onNavigationItemSelectedListener(MenuItem item) {
        switch (item.getItemId()) {
            case R.id.navEstatico:
                System.out.println("-->Nav Estatico");
                break;
            case R.id.navDinamico:
                System.out.println("-->Nav Dinamico");
                break;
            case R.id.navEmail:
                System.out.println("-->Nav Email");
                break;
        }

        drawer.closeDrawer(GravityCompat.START);
        return true;
    }
}

```

6.4. Teste o código anterior e verifique se está a funcionar corretamente.

7. Altere o código para que após a seleção de um item seja apresentado o ecrã pretendido. Para isso vai precisar de criar dois fragmentos: **EstaticoFragment** e **DinamicoFragment**.

7.1. Crie para cada um dos fragmentos um “blank fragment” e copie o conteúdo dos layouts *activity_detalhes_estatico* e *activity_detalhes_dinamico*.

7.2. Na classe de cada fragmento implemente o código necessário, podendo basear-se na ficha anterior. De referir que só iremos precisar do *construtor* e do método **onCreateView**, o restante código pode ser eliminado.

Para aceder aos componentes gráficos, no método *onCreateView* armazene a vista numa variável e a partir dela já conseguirá aceder (ver exemplo abaixo).

```
View view= inflater.inflate(R.layout.fragment_dinamico, container, false);  
tvTitulo = view.findViewById(R.id.tvTitulo);
```

8. Para que possa ser apresentada a informação de cada fragmento de acordo com a opção selecionada, no *MenuMainActivity* vai ser necessário definir um objeto **FragmentManager** para gerir esses fragmentos.

Para o item email não será necessário criar nenhum fragmento, bastará utilizar o intent implícito, elaborado na aula anterior.

O método **onNavigationItemSelectedListener** terá de ser alterado, para incorporar as novas alterações:

```
private FragmentManager fragmentManager;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    //adicionar ao código anterior a seguinte linha  
    fragmentManager = getSupportFragmentManager();  
}  
  
@Override  
public boolean onNavigationItemSelectedListener(MenuItem menuItem) {  
    Fragment fragment = null;  
    switch (menuItem.getItemId()) {  
        case R.id.navEstatico:  
            fragment = new EstaticoFragment();  
            setTitle(menuItem.getTitle());  
            break;  
        // restantes cases  
    }  
  
    if (fragment != null)  
        fragmentManager.beginTransaction().replace(R.id.contentFragment,  
fragment).commit();  
        // ...  
}
```

9. Volte a testar e verifique se ao clicar no item do menu é apresentado o respetivo fragmento.
10. Como deve ter reparado, inicialmente nenhum item está selecionado por isso nenhum fragmento é apresentado. Assim sendo, é necessário efetuar a alterações para que quando é apresentado o *MenuMainActivity* a opção “Livro Estático” esteja selecionada.
 - 10.1 Crie o método **carregarFragmentoInicial** onde terá de selecionar a opção do menu pretendida e fazer o carregamento do fragmento (ver código *onNavigationItemSelectedListener*).
11. Volte a testar e se estiver tudo a funcionar corretamente apague as atividades antigas: **MainActivity**, **DetalhesEstaticoActivity** e **DetalhesDinamicoActivity** assim como todas as dependências e layouts correspondentes.