

Docentes:

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

Ficha Prática 3

Books – Atividades e Intents

Objetivos da Ficha

- Criar um novo projeto “Books”
- Criar uma atividade inicial de Login
- Criar atividades a partir de outras atividades
 - Recorrendo a conteúdo estático e/ou dinâmico
- Enviar um email
- Utilização de uma classe do tipo Singleton

Introdução

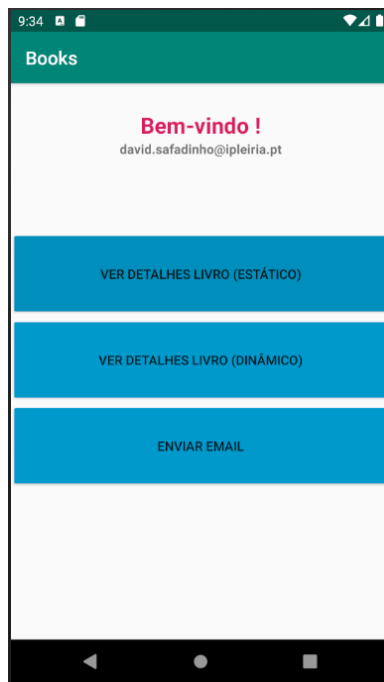
Nesta ficha vamos criar uma aplicação que permita ao utilizador implementar um acesso por login a partir de um email e password.

A janela inicial da aplicação “Books” deve ter um aspeto semelhante ao apresentado na figura seguinte.



Criar um novo projeto (Books)

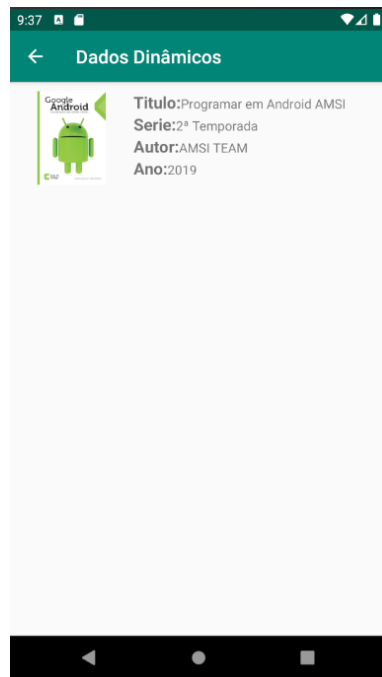
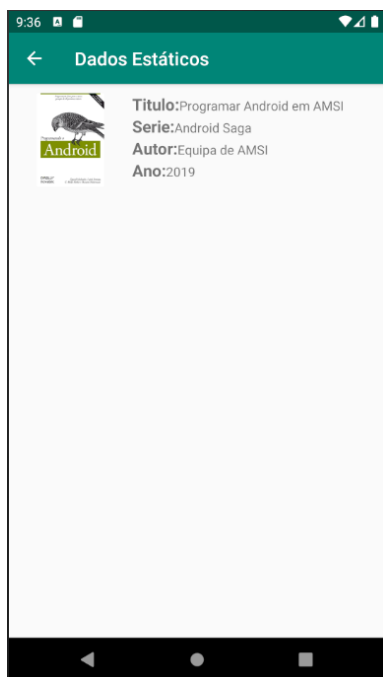
1. Deve efetuar o download da pasta **recursos**, disponibilizada para resolução da ficha.
2. Crie um novo projeto denominado “Books”, que irá ser desenvolvido ao longo do semestre. Deve criar do tipo “**No Activity**”.
3. Como atividade inicial (Launch Activity) crie uma atividade do tipo “**EmptyActivity**” com o nome “**LoginActivity**”.
4. Depois de criar o projeto, para adicionar as imagens disponibilizadas, deve copiar e colá-las na pasta **drawable**.
5. No layout “**login_activity**”, deve adicionar:
 - 5.1. Uma imagem através de uma **ImageView**, usando a imagem do IPL disponibilizada;
 - 5.2. Duas **EditText**, uma para email outra para a password e um **Button** para efetuar o respetivo Login.
 - 5.3. Criar um novo método **isEmailValido()** para validar o email introduzido.
 - 5.4. Criar um novo método **isPasswordValida()** para garantir que a password terá no mínimo 4 caracteres.
 - 5.5. Crie o método **onClickLogin()**, associado ao click do botão, que tem como intuito validar o email e o login e apresentar o resultado da validação num Toast.
6. Criar uma nova Atividade do tipo **empty activity** com o nome “**MainActivity**” que irá apresentar um ecrã, de acordo com o layout da imagem seguinte:



6.1. Volte novamente à **LoginActivity** e efetue as alterações necessárias para que em vez de apresenta um Toast consiga iniciar outra atividade (**MainActivity**).

6.1.1. Esta alteração será efetuada no método **onClickLogin()**, que foi atribuído ao botão **Login**, recorrendo à criação de um novo Intent, que permite passar o email como parâmetro para a **MainActivity**.

7. De seguida deve criar as atividades **DetalhesEstaticoActivity** e **DetalhesDinamicoActivity**, de acordo com o layout apresentado na figura seguinte:



- 7.1. Volte novamente à **MainActivity** e efetue as alterações necessárias para que ao clicar nos botões dos “Detalhes livro” inicie uma outra atividade (**DetalhesEstaticoActivity** ou **DetalhesDinamicoActivity**) consoante a opção clicada.
8. Considerando a atividade **DetalhesEstaticoActivity**, implemente o código necessário de modo a conseguir manipular os componentes.
- 8.1. Para adicionar o botão **back**, na **actionBar**, deve colocar o seguinte código no método **onCreate**. No entanto nas versões mais recentes do Android, este código já não é obrigatório.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    getSupportActionBar().setDisplayHomeAsUpEnabled(true);
}
```

- 8.2. Para que seja possível clicar no botão **back** e voltar à **MainActivity** é necessário, acrescentar no **AndroidManifest.xml** a indicação do respetivo **parent**.

```
<activity android:name=".DetalhesEstatico"
    android:parentActivityName=".MainActivity"/>
```

- 8.3. Teste a aplicação e verifique o que acontece quando voltamos à **MainActivity**;
- 8.4. Volte ao **AndroidManifest.xml** e defina que a **MainActivity** deve ter um **launchMode** definido como **singleTop**. Agora quando fizer **back** o endereço de email continua a ser apresentado.

```
<activity android:name=".MainActivity"
    android:launchMode="singleTop"/>
```

9. De forma a organizar as classes e a separar conceitos, deve-se criar um **package** com o nome **modelo**, e outro com o nome **vistas** onde vai colocar as Activities criadas.
10. No package **modelo** deve criar a classe **Livro**:
- 10.1. Os atributos necessários são: **id**, **titulo**, **serie**, **autor**, **ano**, **capa**.
- 10.1.1. O **id** e o **ano** devem ser armazenados com o tipo de dados inteiro, assim como a **capa**, apesar de corresponder a uma imagem.
- 10.1.2. Implemente o construtor e os respetivos métodos “*Getters and Setters*” exceto o *setter* do **id**.

11. De seguida, no mesmo package deve criar a classe do tipo *Singleton* denominada **SingletonGestorLivros**, para conter a simulação dos dados dinâmicos.
 - 11.1. Esta classe deve conter uma lista de livros, que será iniciada no respetivo construtor, a partir do método **gerarDadosDinamico()**.
 - 11.2. Deve também ter um método que devolva os respetivos livros da lista.
12. Após a criação destas classes pode proceder à implementação do código da atividade **DetalhesDinamicoActivity**.
13. Adicione os métodos de ciclo de vida de uma atividade (mencionados na aula teórica), inclua neles instruções para imprimir textos para a consola e execute a aplicação verificando a ordem das chamadas. Use o debugger para poder parar em cada método e ver as informações de contexto sobre o estado da aplicação e respetiva atividade.
14. Na **MainActivity** deve implementar a funcionalidade do botão **Enviar Email**, recorrendo ao respetivo **Intent** implícito.