

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Programação Para Dispositivos Móveis I

FICHEIROS

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis Ricardo Barbosa , rmb@estg.ipp.pt
Carlos Aldeias, cfpa@estg.ipp.pt







Índice

- Ficheiros;
- Armazenamento Interno;
- Armazenamento Externo;
- Notas Finais;
- Leitura Adicional.









Em conjunto com SharedPreferences e Databases, Ficheiros são um de três métodos de **persistência de dados** em Android.

- A sua utilização **não é recomendada** (geralmente);
- O Android utiliza os mesmos mecanismos de manipulação de ficheiros do Java.

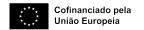
Os ficheiros podem ser armazenados em:

- Armazenamento interno (Ficam na mesma localização que os outros recursos como os ícones, imagens, música,...);
- Armazenamento externo (ex. cartão SD).









Armazenamento Interno

Os ficheiros gravados neste tipo de armazenamento são privados à aplicação (tal como os resources), e as outras aplicações (e o utilizador) não conseguem obter acesso aos mesmos.

 Quando o utilizador desinstala a aplicação estes ficheiros são também são eliminados;

ATENÇÃO: O espaço Interno é muito limitado comparativamente com o externo.









Armazenamento Interno -> Criação e Escrita

- Invocar o método openFileOutput()
 - Recebe comos parâmetros o nome do ficheiro e o modo de operação;
 - MODE_PRIVATE cria o ficheiro e torna-o privado à aplicação;
 - MODE_APPEND se o ficheiro já existir então adiciona os dados no final deste em vez de o apagar;
 - Retorna um objeto do tipo FileOutputStream;

- 2. Escrever o ficheiro utilizando o método write();
- 3. Fechar o stream de dados utilizando o método close().









Armazenamento Interno -> Criação e Escrita

```
String file_name = "test_file.txt";
String textContent = "Hello World!";
FileOutputStream fos = null;
try {
    fos = openFileOutput(file_name, this.MODE_PRIVATE);
    fos.write(textContent.getBytes());
} catch (FileNotFoundException fileNotFoundException) {
    Log.e("FILE_NOT_FOUND", fileNotFoundException.getMessage());
} catch (IOException ioException) {
   Log.e("INPUT", ioException.getMessage());
} finally {
   if (fos \neq null)
       fos.close();
```









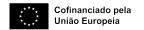
Armazenamento Interno -> Leitura

- Invocar o método openFileInput()
 - Recebe como parâmetro o nome do ficheiro a ler;
 - Retorna um objeto do tipo FileInputStream;
- 2. Ler os bytes do ficheiro utilizando o método read();
- 3. Fechar o stream de dados utilizando o método close();







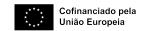


Armazenamento Interno -> Leitura

```
String file_name = "test_file.txt";
FileInputStream fileInputStream = null;
try {
    fileInputStream = openFileInput(file_name );
    int i;
    while((i=fileInputStream.read())\neq-1) {
} catch (FileNotFoundException fileNotFoundException) {
    Log.e("FILE_NOT_FOUND", fileNotFoundException.getMessage());
} catch (IOException ioException) {
    Log.e("INPUT", ioException.getMessage());
} finally {
   if (fileInputStream \neq null)
       fileInputStream.close();
```







Armazenamento Interno -> Outros métodos

```
getFilesDir()
```

Obtém o caminho absoluto para o diretório onde os ficheiros são gravados;

```
getDir()
```

Cria (ou abre caso exista) o diretório especificado;

```
deleteFile()
```

Apaga o ficheiro;

```
fileList()
```

Retorna um array com todos os ficheiros existentes no diretório.









Armazenamento Externo

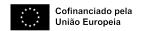
Este tipo de armazenamento pode ser do tipo **removível** (ex. SD Card) ou **embebido** no dispositivo. Neste tipo de armazenamento os ficheiros podem ser adicionados, editados, removidos e copiados pelo utilizador **não havendo controlo de segurança** sobre os mesmos.

 O armazenamento externo pode ficar indisponível caso o utilizador faça mount num computador ao armazenamento externo (ligação por USB)









Armazenamento Externo -> Disponibilidade

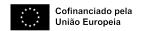
Antes de fazer qualquer operação sobre o armazenamento externo devese verificar qual o estado do mesmo, utilizando o método Environment.getExternalStorageState(), o qual pode retornar:

- Environment.MEDIA_MOUNTED o armazenamento está mounted a um computador e não pode ser utilizado;
- Environment.MEDIA_MOUNTED_READ_ONLY só temos acesso de leitura ao armazenamento;
- Environment.MEDIA_REMOVED armazenamento externo n\u00e3o est\u00e1
 dispon\u00edvel;
- Entre outros...









Armazenamento Externo -> Permissões [AndroidManifest.xml]

Permissões só de leitura

<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />









Armazenamento Externo -> Acesso

A path para este armazenamento pode ser obtida utilizando os métodos:

getExternalFilesDir()

- Recebe como parâmetro o tipo de diretório:
 - DIRECTORY_MUSIC;
 - DIRECTORY_RINGTONES;
 - DIRECTORY_PICTURES;
 - entre outros...
- Ou null caso seja de outro tipo;
- O ficheiro é apagado quando a aplicação é desinstalada.

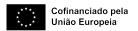
getExternalStoragePublicDirectory()

- Recebe como parâmetro o tipo de diretório:
 - DIRECTORY_MUSIC;
 - DIRECTORY_RINGTONES;
 - DIRECTORY_PICTURES;
 - entre outros...
- Ou null caso seja de outro tipo;
- O ficheiro não é apagado quando a aplicação é desinstalada.









Armazenamento Externo -> Exemplo









Notas

Armazenamento Interno	Armazenamento Externo
Está sempre disponível	Não está sempre disponível. (Ex: Ligado ao PC, retirado o cartão memória,)
Ficheiros apenas acessíveis pela aplicação	É visível a todas as aplicações android e pode ser acedido por fontes externas
Quando a aplicação é removida os ficheiros também são removidos	Quando a aplicação é desinstalada, o sistema android só elimina o ficheiro se a diretoria onde foi gravado foi obtida através de getExternalFilesDir()









Leitura Adicional

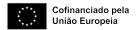
Data and File Storage:

https://developer.android.com/training/data-storage











ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Programação Para Dispositivos Móveis I

FICHEIROS

2024/_25 CTeSP – Desenvolvimento para a Web e Dispositivos Móveis Ricardo Barbosa , rmb@estg.ipp.pt
Carlos Aldeias, cfpa@estg.ipp.pt

Adaptação do conteúdo dos slides de João Ramos <u>irmr@estq.ipp.pt</u> e Fábio Silva <u>fas@estq.ipp.pt</u>





