



DESENVOLVIMENTO AVANÇADO MOBILE

GOOGLE ANDROID



DESENVOLVIMENTO ANDROID

PERSISTÊNCIA

ÍNDICE

- SHARED PREFERENCES
- FILES
- SQLITE
 - ACTIVE ANDROID

SHARED PREFERENCES

É um formato simples de armazenamento de pares de valores contendo um nome e valor primitivo;

A classe **SharedPreferences** encapsula os dados armazenados e pode ser obtida por meio dos métodos:

getPreferences(int P1): obtém um único arquivo de preferências;

getSharedPreferences(String P1, int P2): obtém um arquivo de preferências com o nome especificado em P1;

O parâmetro P1 do **getPreferences** e P2 do **getSharedPreferences** especificam as permissões dos arquivos:

MODE_PRIVATE - somente a aplicação pode acessar;

MODE_WORLD_READABLE - público somente leitura;

MODE_WORLD_WRITEABLE - público com leitura / escrita;

Uma vez obtido um objeto **SharedPreferences** basta acionar o método **edit** para retornar a interface **Editor**;

Com o **Editor** pode-se criar pares de valores por meio de métodos:

- **putBoolean(String P1, boolean P2);**
- **putFloat(String P1, float P2);**
- **putInt(String P1, int P2);**
- **putLong(String P1, long P2);**
- **putString(String P1, String P2);**

Onde **P1** representa o nome do valor (campo) e **P2** o valor a ser armazenado no respectivo tipo de dado;

Para confirmar a persistência dos dados chamar o método `commit()` da classe `Editor`:

```
SharedPreferences sp = getPreferences(MODE_PRIVATE);  
Editor e = sp.edit();  
e.putString("username", "rubim");  
e.commit();
```

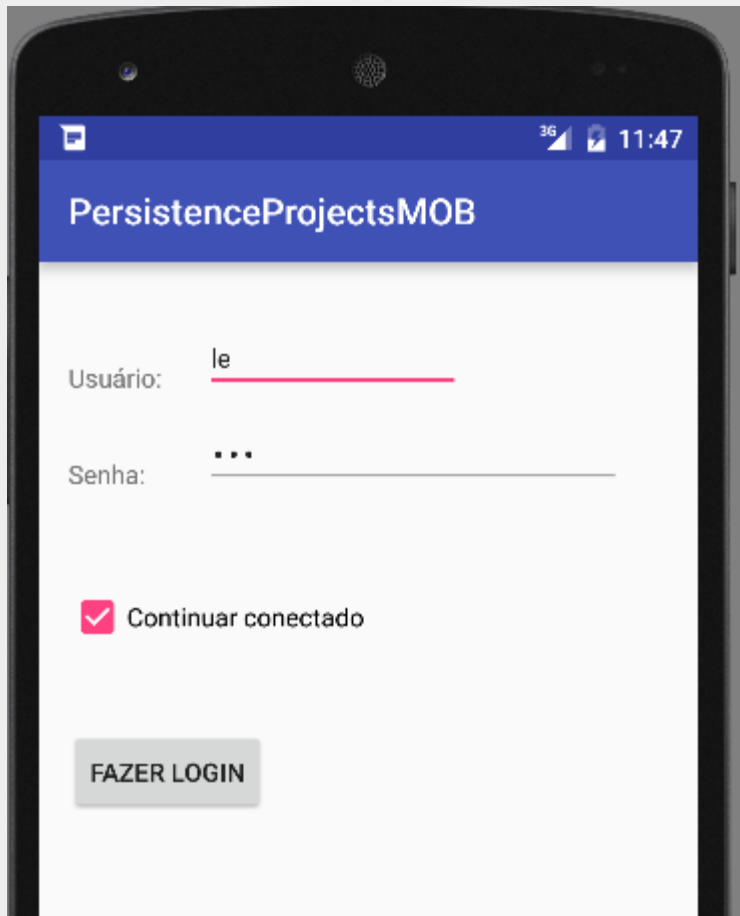
A leitura de uma determinada propriedade deve ser feita diretamente por meio de métodos da classe **SharedPreferences**:

- **getBoolean(String P1, boolean P2);**
- **getFloat(String P1, float P2);**
- **getInt(String P1, int P2);**
- **getLong(String P1, long P2);**
- **getString(String P1, String P2);**

Onde **P1** representa o nome do valor (campo) e **P2** o valor padrão (default) a ser retornado caso o nome do valor informado não tenha sido ainda persistido;

```
SharedPreferences sp = getPreferences(MODE_PRIVATE);  
  
// Caso o valor username não exista retorna null  
String username = sp.getString("username", null);  
Toast.makeText(this, username, Toast.LENGTH_SHORT).show();
```

ATIVIDADE 1 – SHARED PREFERENCES



Utilizando SharedPreferences escreva uma aplicação que:

1. Usuário informa o seu usuário e senha e opta por “continuar conectado”
2. Aplicação registra o usuário e senha em uma Shared Preference
3. Usuário executa novamente a aplicação
4. Aplicação lê o usuário e senha salvos anteriormente no SharedPreferences e os exibe nos respectivos campos

Também é possível gravar e ler arquivos nos diretórios do dispositivo. Para tanto, basta utilizar um **FileOutputStream** e o **FileInputStream** respectivamente;

Exemplo de gravação de um texto:

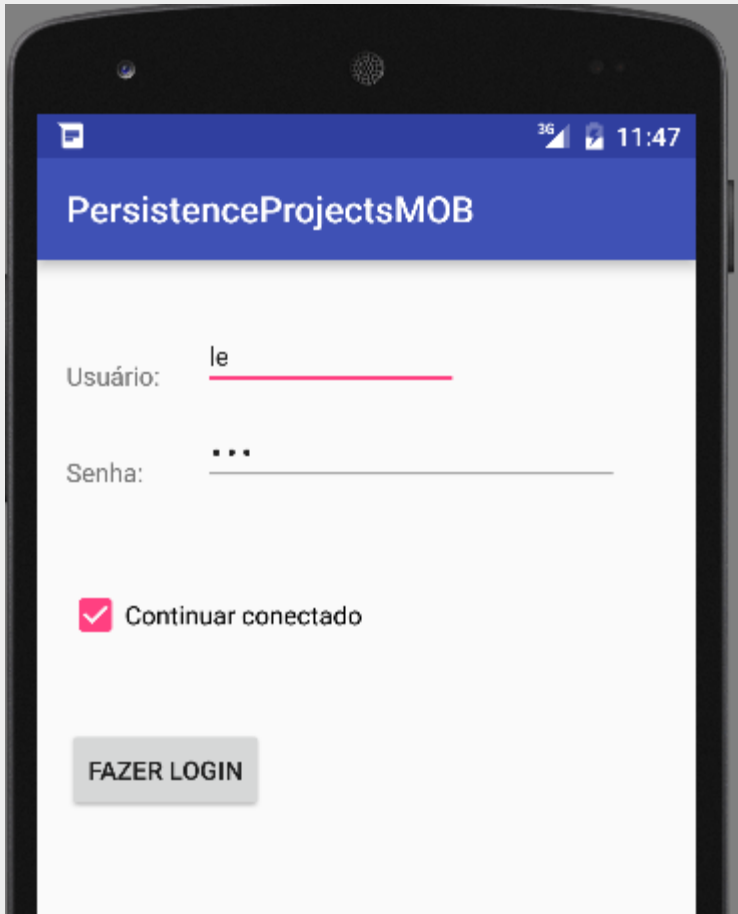
```
// Abre um stream de saída para gravação de arquivos
FileOutputStream fos = openFileOutput("teste.txt", MODE_PRIVATE);
String txt = "Boa noite!";
fos.write(txt.getBytes());
fos.close();
```

Exemplo de leitura de um texto:

```
// Abre um stream de entrada para leitura de arquivos
FileInputStream fis = openFileInput("teste.txt");
BufferedReader br = new BufferedReader(new InputStreamReader(fis));

// Lê uma linha do arquivo e a retorna uma String
String txt = br.readLine();
fis.close();
Toast.makeText(this, txt, Toast.LENGTH_SHORT).show();
```

ATIVIDADE 2 – FILES



Utilizando Files escreva uma aplicação similar a escrita com SP, onde:

1. Usuário informa o seu usuário e senha e opta por “continuar conectado”
2. Aplicação registra o usuário e senha em um Arquivo
3. Usuário executa novamente a aplicação
4. Aplicação lê o usuário e senha salvos anteriormente no Arquivo e os exibe nos respectivos campos



Copyright © 2016 - Profs. Me. Leandro Rubim, Prof. Me. Thiago T. I. Yamamoto e Prof. Me. Edson Sensato

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Autor.

