

DESENVOLVIMENTO MOBILE - ANDROID

Persistência I

Fernando Lima

Proffernando.lima@fiap.com.br

Shared Preferences

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

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

getSharedPreferences(String P1, int P2) → obtém um arquivo de preferências com o nome especificado em P1 e o modo de acesso em P2:

MODE_PRIVATE → somente a aplicação pode acessar;

MODE_WORLD_READABLE → público somente leitura;

MODE_WORLD_WRITEABLE → público com leitura / escrita;

Shared Preferences

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 = getSharedPreferences("teste", MODE_PRIVATE);  
Editor e = sp.edit();  
e.putString("username", "jpereira");  
e.commit();
```

Shared Preferences

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;

Exemplo:

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

Shared Preferences

```
public class MainActivity extends AppCompatActivity {

    private EditText edtUsu;
    private EditText edtSen;
    private SharedPreferences pref;

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

        edtUsu = (EditText) findViewById(R.id.edtUsuario);
        edtSen = (EditText) findViewById(R.id.edtSenha);

        pref = getSharedPreferences("login", MODE_PRIVATE);

        /*
        String u = pref.getString("usuario", null);
        String s = pref.getString("senha", null);

        if (u != null)
            edtUsu.setText(u);
        if (s != null)
            edtSen.setText(s);
        */
    }
}
```



Shared Preferences

```
public void gravar(View v) {

    SharedPreferences.Editor e = pref.edit();
    e.putString("usuario", edtUsu.getText().toString().trim());
    e.putString("senha", edtSen.getText().toString().trim());
    e.commit();

}

public void login(View v){

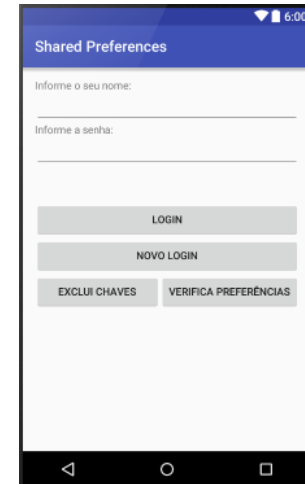
    String u = pref.getString("usuario", "");
    String s = pref.getString("senha", "");

    if ((u.equals(edtUsu.getText().toString())) && (s.equals(edtSen.getText().toString())))
        Toast.makeText(this, "Usuário válido.", Toast.LENGTH_LONG).show();
    else
        Toast.makeText(this, "Usuário inválido.", Toast.LENGTH_LONG).show();

}

public void excluirChave(View v){
    SharedPreferences.Editor e = pref.edit();
    e.remove("usuario");
    e.remove("senha");
    e.commit();
}

public void verificaPreferencia(View v) {
    SharedPreferences sharedPrefs = getSharedPreferences("login", MODE_PRIVATE);
    if (!sharedPrefs.contains("initialized")) {
        Toast.makeText(this, "Preferência inicializada.", Toast.LENGTH_LONG).show();
    }
}
```



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 = "Valor a ser gravado...";
```

```
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();
```

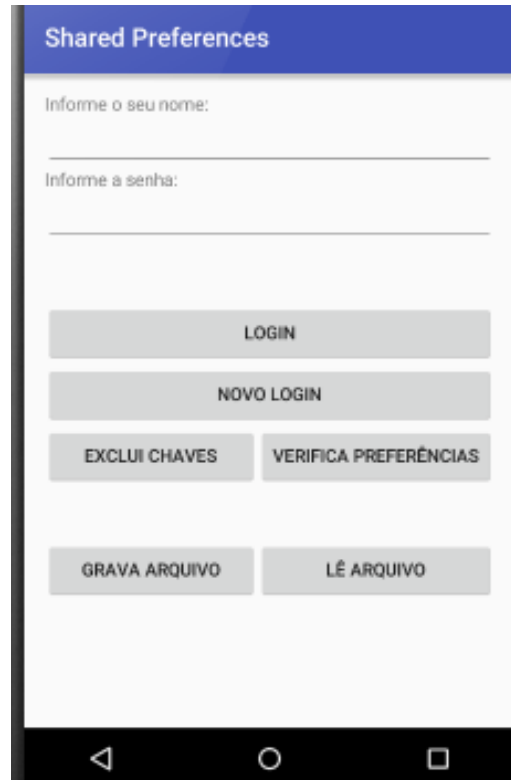
```
fis.close();
```

```
Toast.makeText(this, txt, Toast.LENGTH_SHORT).show();
```

Tratar Exceções.

Exercício

Desenvolva os botões grava arquivo e lê arquivo, salvando o conteúdo digitado.



The image shows a mobile application interface titled "Shared Preferences". It features two text input fields: "Informe o seu nome:" and "Informe a senha:". Below these fields are five buttons arranged vertically: "LOGIN", "NOVO LOGIN", "EXCLUI CHAVES", "VERIFICA preferências", and "GRAVA ARQUIVO". The "GRAVA ARQUIVO" button is highlighted with a blue background, while the others have a light gray background. At the bottom of the screen, there is a black navigation bar with three white icons: a triangle, a circle, and a square.

Copyright © 2015 Prof. Fernando Lima e Edson Alexandre Sensato

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