



android

Prof. Diogo Soares

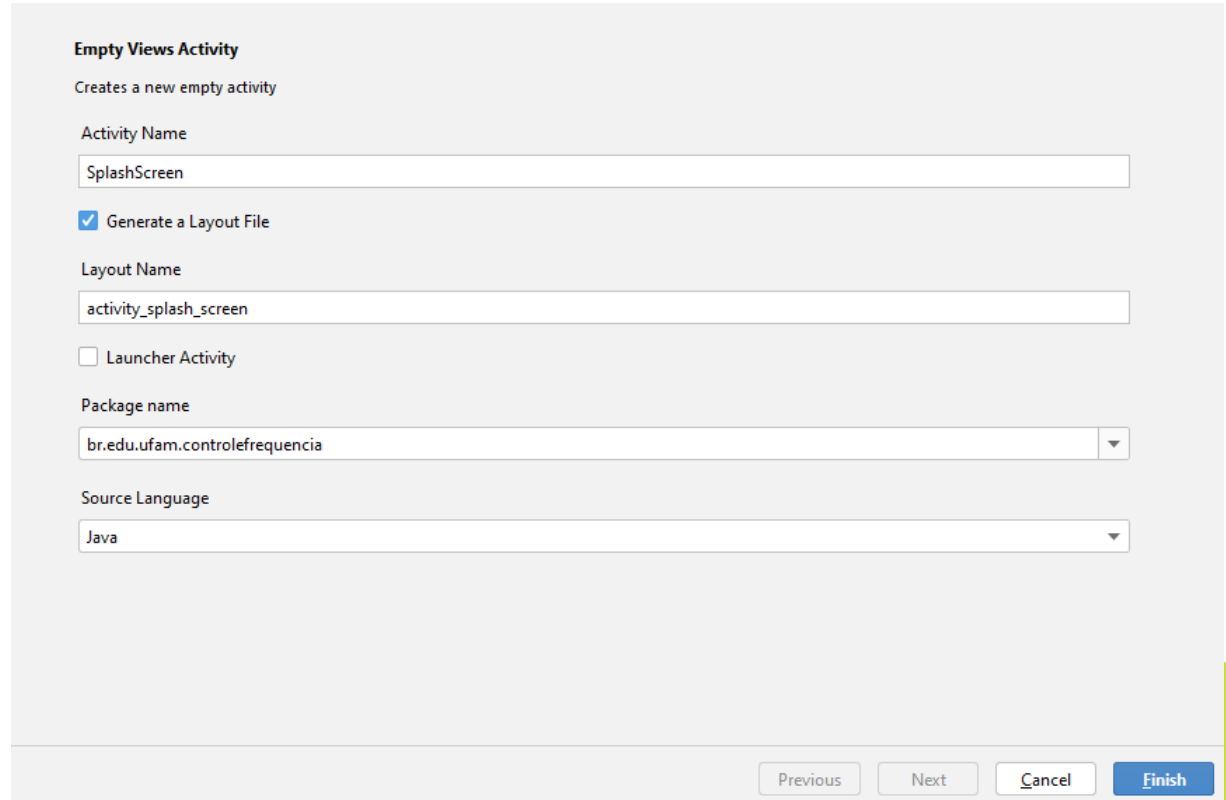
O que é splash screen?



- Tela de abertura exibida ao usuário como primeira tela do aplicativo
- Em geral fica aberta por alguns segundos antes de ir para a próxima tela do aplicativo, usando intents

Criando uma Splash Screen

- Crie uma nova activity (pasta java, Ctrl direito -> new -> Activity -> Empty view activity)



The screenshot shows the 'Empty Views Activity' dialog box. It has a title bar 'Empty Views Activity' and a subtitle 'Creates a new empty activity'. The form contains the following fields and options:

- Activity Name:** A text input field containing 'SplashScreen'.
- Generate a Layout File:** A checked checkbox.
- Layout Name:** A text input field containing 'activity_splash_screen'.
- Launcher Activity:** An unchecked checkbox.
- Package name:** A dropdown menu showing 'br.edu.ufam.controlefrequencia'.
- Source Language:** A dropdown menu showing 'Java'.

At the bottom right, there are four buttons: 'Previous', 'Next', 'Cancel', and 'Finish'.

Criando uma Splash Screen



- Adicione elementos a sua tela
- Tente adicionar uma logo e uma barra para indicar que está iniciando
- Neste exemplo iremos adicionar uma imagem e uma barra de progress
 - ImageView & ProgressBar

Splash Screen Controller

- No controller, adicione uma função para rodar por um breve período aquela tela (activity)
- Após isso, chame a próxima activity usando intents

Requisita o carregamento da tela para usuário (onStart)

```
public class SplashScreen extends AppCompatActivity {  
  
    static int TEMPO_SPLASH_SCREEN = 2000; //ms  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_splash_screen);  
    }  
}
```

Seta um tempo de duração

Splash Screen Controller

- No controller, adicione uma função para rodar por um breve período aquela tela (activity)
- Após isso, chame a próxima activity usando intents

Após isso, chame um método para adicionar um atraso (função Handler com postDelayed())

```
// Adicione um atraso de 2 segundos antes de iniciar a próxima activity
new Handler().postDelayed(new Runnable() {
    @Override
    public void run() {
        irParaProximoActivity();
    }

    private void irParaProximoActivity() {
        Intent intent = new Intent(SplashScreen.this, MainActivity.class);
        startActivity(intent);
        finish(); //finaliza o activity
    }
}, TEMPO_SPLASH_SCREEN);
```

Define uma função para fazer chamada de outra activity

Finish() finaliza a activity e remove-a da pilha de activities (lembra do onDestroy()? :p)

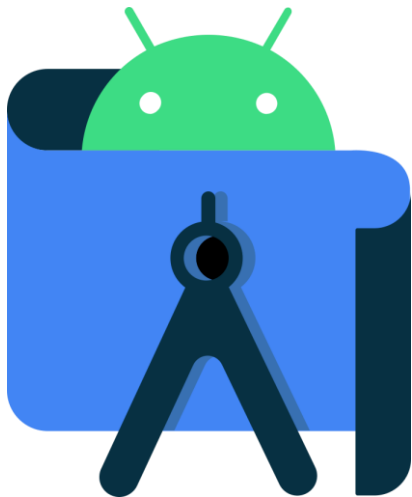
Splash Screen Controller

- Por fim, mude o manifesto para representar quem é a tela primária nas categorias MAIN e LAUNCHER
- Edite o AndroidManifest.xml. Perceberão que a nova activity já está lá 😊

```
<activity  
  android:name=".SplashScreen"  
  android:exported="true" >  
  <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
  
    <category android:name="android.intent.category.LAUNCHER" />  
  </intent-filter>  
</activity>
```

Define nome da
activity de splash

Adiciona filtro de tela
primária e launcher



Activities e Intents

Prof. Diogo Soares



Abrindo outras Activities

- Um `Intent` provê uma ligação entre dois componentes que, em geral, são duas activities
 - Indica uma intenção de fazer alguma coisa

Intenção

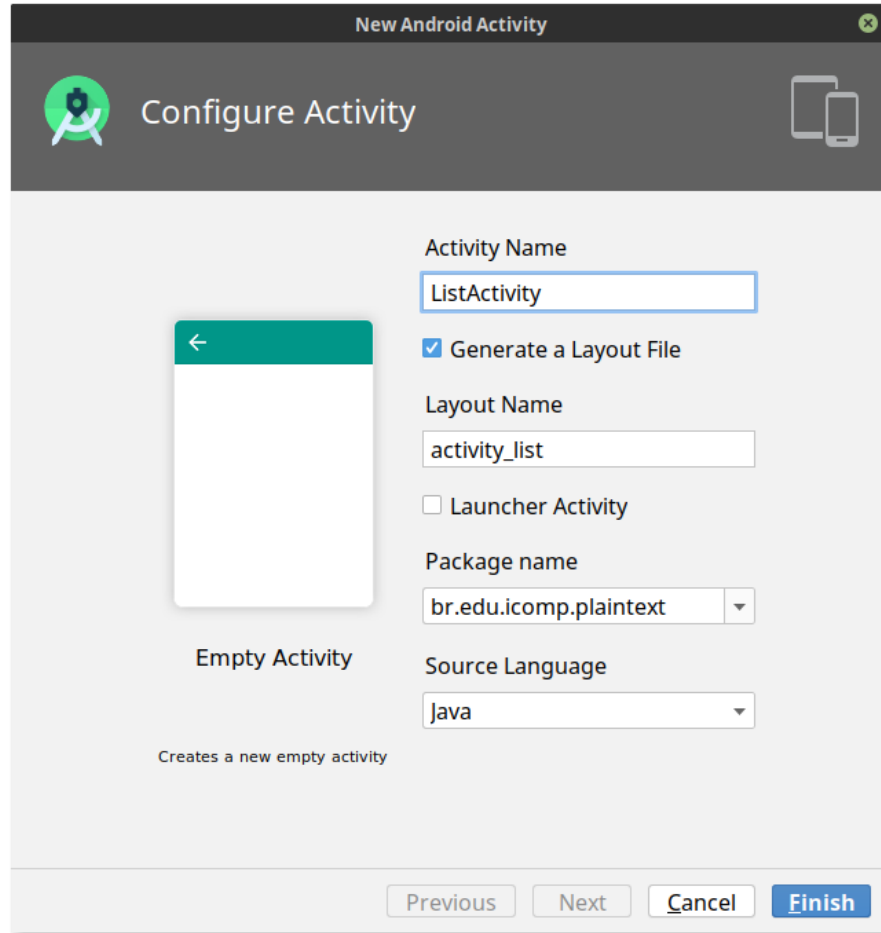
```
public void entrarClicado(View view) {  
    Intent intent = new Intent(this, ListaActivity.class);  
    startActivity(intent);  
}
```

Execução

- A partir de agora, o botão “Entrar”, irá abrir uma nova activity
 - Entretanto, precisamos criar essa nova activity (`ListaActivity`)

Criando a Nova Activity

- No Android Studio
 - File
 - New
 - Activity
 - Empty Activity

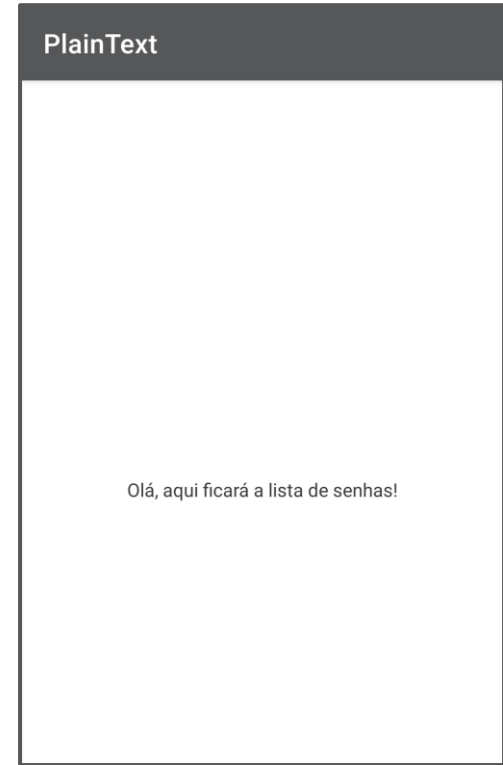


Testando a Nova Activity

- Ao clicar no botão “Entrar”, a nova activity é aberta



The screenshot shows the login interface for an application named 'PlainText'. At the top, there's a dark header with the title 'PlainText' and a menu icon. Below this is a green banner featuring a white icon of a key on a document and the text: "The most secure password manager" and "Bob and Alice". The main content area is white and contains the instruction "Digite suas credenciais para continuar". There are two input fields: "Login:" with the text "dovahkiin" and "Senha:" (Password). Below the password field is a checkbox labeled "Salvar informação de login". At the bottom, there is a dark button with the text "ENTRAR" in green.



The screenshot shows the main interface of the 'PlainText' application. It has a dark header with the title 'PlainText'. The main content area is white and contains the text: "Olá, aqui ficará a lista de senhas!" (Hello, here will be the list of passwords!).

Passando Dados entre Activities

- Para passar dados para a activity que será aberta, usa-se o método `putExtra` da classe `Intent`:

Acessa o login do usuário

Passando o login para a próxima activity

```
public void entrarClicado(View view) {  
    Intent intent = new Intent(this, LoginActivity.class);  
  
    EditText inputLogin = findViewById(R.id.editLogin);  
    intent.putExtra("login", inputLogin.getText().toString());  
  
    startActivity(intent);  
}
```

Inicia a nova activity

Passando Dados entre Activities

- Para acessar os dados na nova activity, usa-se o método `getStringExtra`:

```
public class LoginActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_list);  
  
        TextView textBemVindo =  
            findViewById(R.id.textBemVindo);  
  
        Intent intent = getIntent();  
        String login = intent.getStringExtra("login");  
  
        textBemVindo.setText("Olá " + login + "!");  
    }  
}
```

The diagram illustrates the process of passing data between activities. It shows the `onCreate` method of `LoginActivity` where data is retrieved from an intent and used to update a `TextView`. Blue boxes with arrows point to specific lines of code to explain each step:

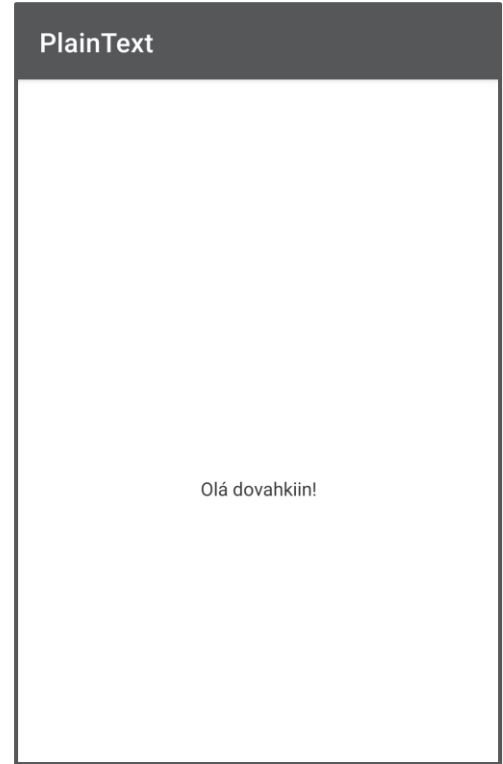
- onCreate da nova Activity**: Points to the `onCreate` method signature.
- Acessa o texto de bem-vindo**: Points to the `findViewById` call for `textBemVindo`.
- Acessa o intent que abriu a activity**: Points to the `getIntent()` call.
- Acessa o valor do login passado**: Points to the `getStringExtra("login")` call.
- Muda a mensagem de bem-vindo**: Points to the `setText` call on `textBemVindo`.

Passando Dados entre Activities

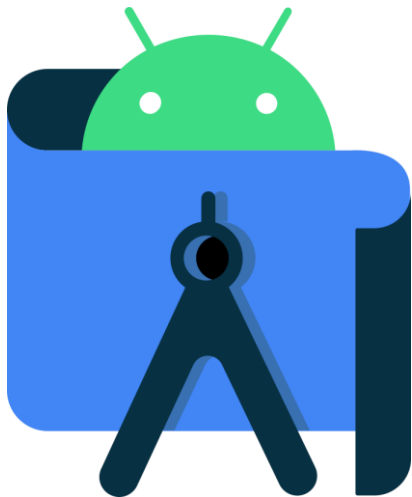
- Testando novamente o botão “Entrar”:



The screenshot shows the login interface of an application named 'PlainText'. At the top, there's a dark header with the title 'PlainText' and a menu icon. Below the header is a green banner featuring a white key icon and the text: "The most secure password manager" and "Bob and Alice". The main content area is white and contains the instruction "Digite suas credenciais para continuar". There are two input fields: "Login:" with the text "dovahkiin" and "Senha" (Password). Below the password field is a checkbox labeled "Salvar informação de login". At the bottom, there is a dark button with the text "ENTRAR" in green.



The screenshot shows the home screen of the 'PlainText' application. It has a dark header with the title 'PlainText'. The main content area is white and displays the text "Olá dovahkiin!" in the center.



Activity de Preferências

Prof. Diogo Soares



Activity de Preferências

- As duas principais formas de armazenar dados em um app são:
 - Usando preferências
 - Usando banco de dados
- Por enquanto, iremos mostrar como criar preferências
 - Futuramente, veremos banco de dados
- Em nosso app de exemplo:
 - As informações de login para acessar o app serão salvas nas preferências
 - Os dados (itens) das senhas serão salvas no banco de dados

Criando Preferências



- As preferências permitem armazenar dados de configurações da aplicação atual
- A forma mais prática de criar uma tela de configurações simples (com apenas uma tela), é
 - Criando um arquivo XML com as preferências
 - Criando uma activity que, ao ser criada, inicia um Fragmento de Preferências (classe PreferenceFragmentCompat), que provê todo o código necessário para:
 - *Mostrar as preferências de forma padronizada entre os diversos dispositivos*
 - *Salvar as configurações*

Biblioteca de Preferências

- A parte de preferências do Android foi reimplementada
 - Devido a isso, é necessário incluir uma biblioteca no seu app:
 - Botão direito no App → Open Module Settings → Dependencies

The screenshot illustrates the process of adding a dependency to an Android app. On the left, the 'Project Structure' dialog is open, showing the 'Dependencies' tab for the 'app' module. A blue box labeled 'App' points to the 'app' module in the 'Modules' list. Another blue box labeled '+' points to the '+' icon next to the 'app' module. A third blue box labeled 'Java Dependency' points to the 'Library Dependency' option in the 'Declared Dependencies' list. Below this, the 'Details' section shows the dependency information for 'androidx.preference:preference:1.1.1'.

On the right, the 'Add Library Dependency' dialog is open. A blue box labeled 'androidx.preference:preference' points to the search input field. The dialog shows the search results for 'androidx.preference:preference' and the 'Library' field is populated with 'androidx.preference:preference:1.1.1'.

Project Structure - Dependencies

Declared Dependencies
1 Library Dependency
2 Jar Dependency

Details

Group ID: androidx.preference
Artifact Name: preference
Requested Version: 1.1.1

Add Library Dependency

Module 'app'

Step 1. Use the form below to find the library to add. This form uses the repositories specified in the project's build files (Google, JCenter)

Search: androidx.preference:preference

Enter a search query or fully-qualified coordinates (e.g. guava* or com.google.*:guava* or com.google.guava:guava:26.0)

Group ID	Artifact Name	Repository	Versions
androidx.preference	preference	Google	1.1.1

Library: androidx.preference:preference:1.1.1

Step 2. Assign your dependency to a configuration by selecting one of the configurations below.

Open Documentation

implementation

OK Cancel Apply

Biblioteca de Preferências

- Ou pelo arquivo de configuração do gradle:
 - Devido a isso, é necessário incluir uma biblioteca no seu app:
 - Botão direito no App → Open Module Settings → Dependencies

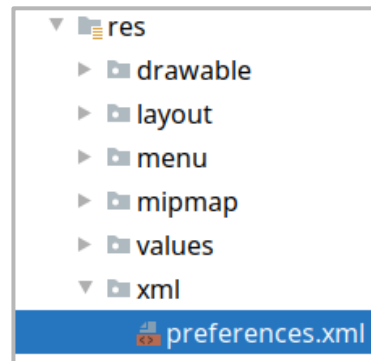
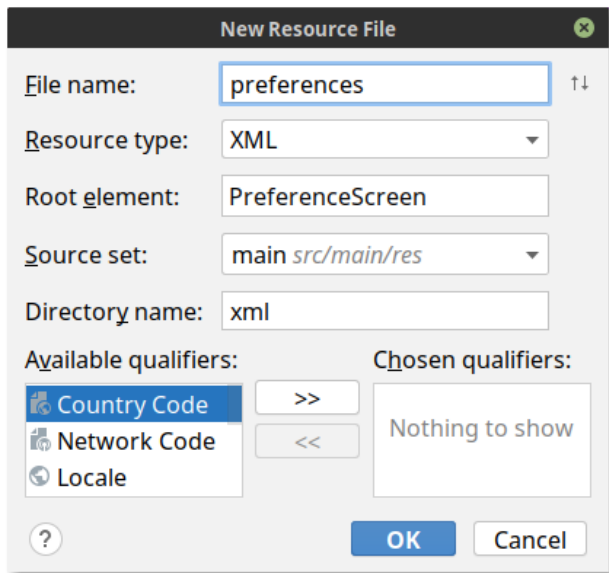
The image shows a snippet of a Gradle file in an IDE. Two blue callout boxes are present: one labeled 'versão' pointing to the variable `preference_version` on line 35, and another labeled `androidx.preference:preference` pointing to the implementation line on line 40. The code is as follows:

```
33
34 dependencies {
35     def preference_version :String = "1.2.0"
36
37     implementation 'androidx.appcompat:appcompat:1.6.1'
38     implementation 'com.google.android.material:material:1.9.0'
39     implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
40     implementation "androidx.preference:preference:$preference_version"
41     testImplementation 'junit:junit:4.13.2'
42     androidTestImplementation 'androidx.test.ext:junit:1.1.5'
43     androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
44 }
```

XML de Preferências

■ Para criar o arquivo XML de preferências:

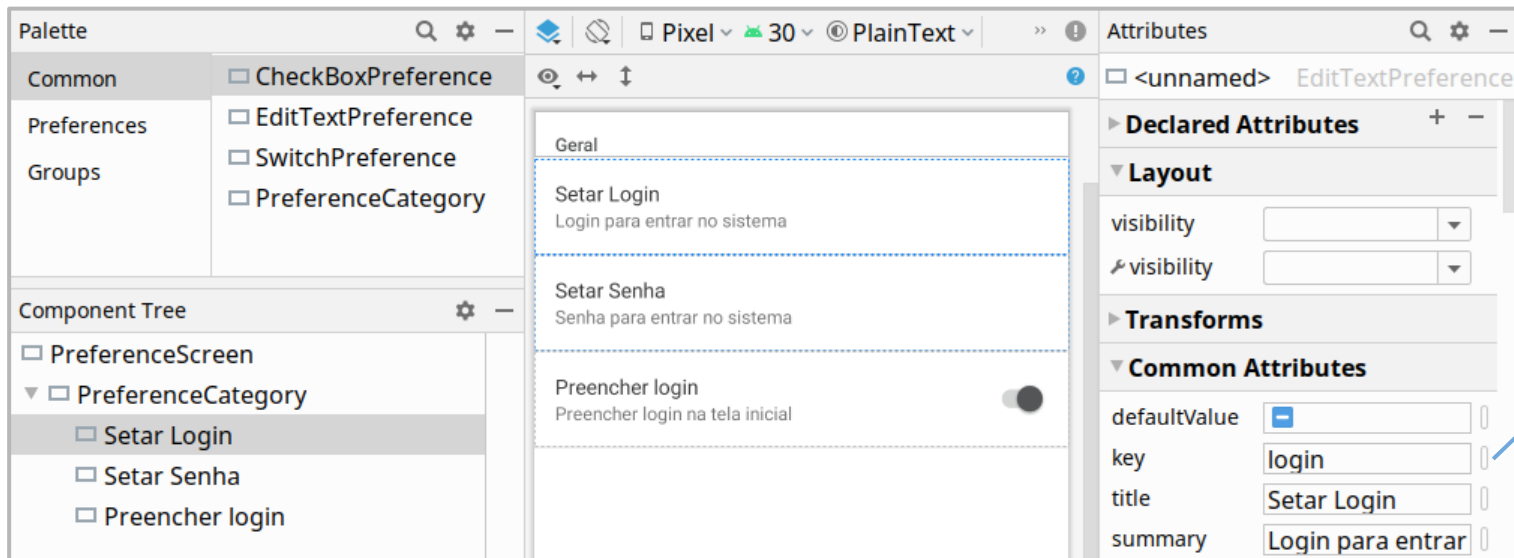
□ File → New → Android Resource File



Abra o
arquivo XML
de
preferências

XML de Preferências

- Assim como no Menu, você pode editar as preferências na tela de design ou editando o XML diretamente (próximo slide)
 - Editando na tela de design:



XML de Preferências

XML resultante do slide anterior

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
  <PreferenceCategory android:title="Geral">
    <EditTextPreference
      android:dialogTitle="Setar Login"
      android:key="login"
      android:summary="Login para entrar no sistema"
      android:title="Setar Login" />
    <EditTextPreference
      android:dialogTitle="Setar Senha"
      android:key="password"
      android:dialogMessage="Digite a senha"
      android:inputType="textPassword"
      android:summary="Senha para entrar no sistema"
      android:title="Setar Senha" />
    <SwitchPreference
      android:key="showLogin"
      android:title="Preencher login"
      android:summary="Preencher login na tela inicial"
      android:defaultValue="true"/>
  </PreferenceCategory>
</PreferenceScreen>
```

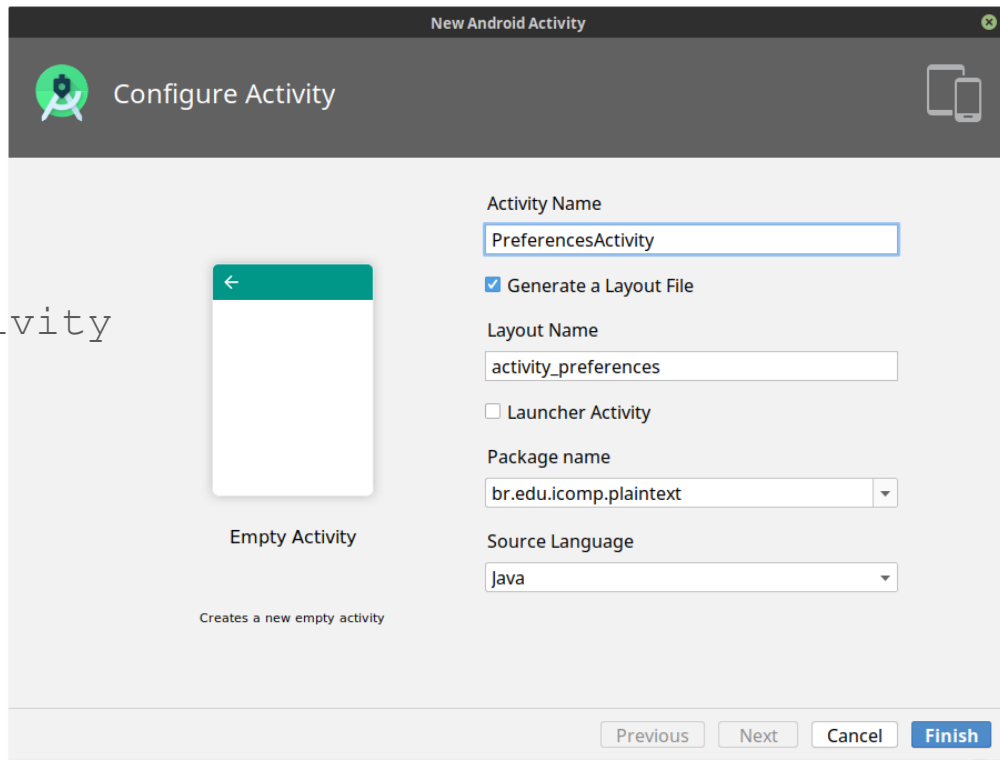
Note a chave
dos itens de
preferência

Activity de Preferências

■ Por fim, vamos criar a activity de preferências:

- File
 - New
 - Activity
 - Empty Activity

□ Nome:
PreferencesActivity



Activity de Preferências

▣ PreferencesActivity.java

```
public class PreferencesActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        getSupportFragmentManager()
            .beginTransaction()
            .replace(android.R.id.content, new PreferencesFragment())
            .commit();
    }

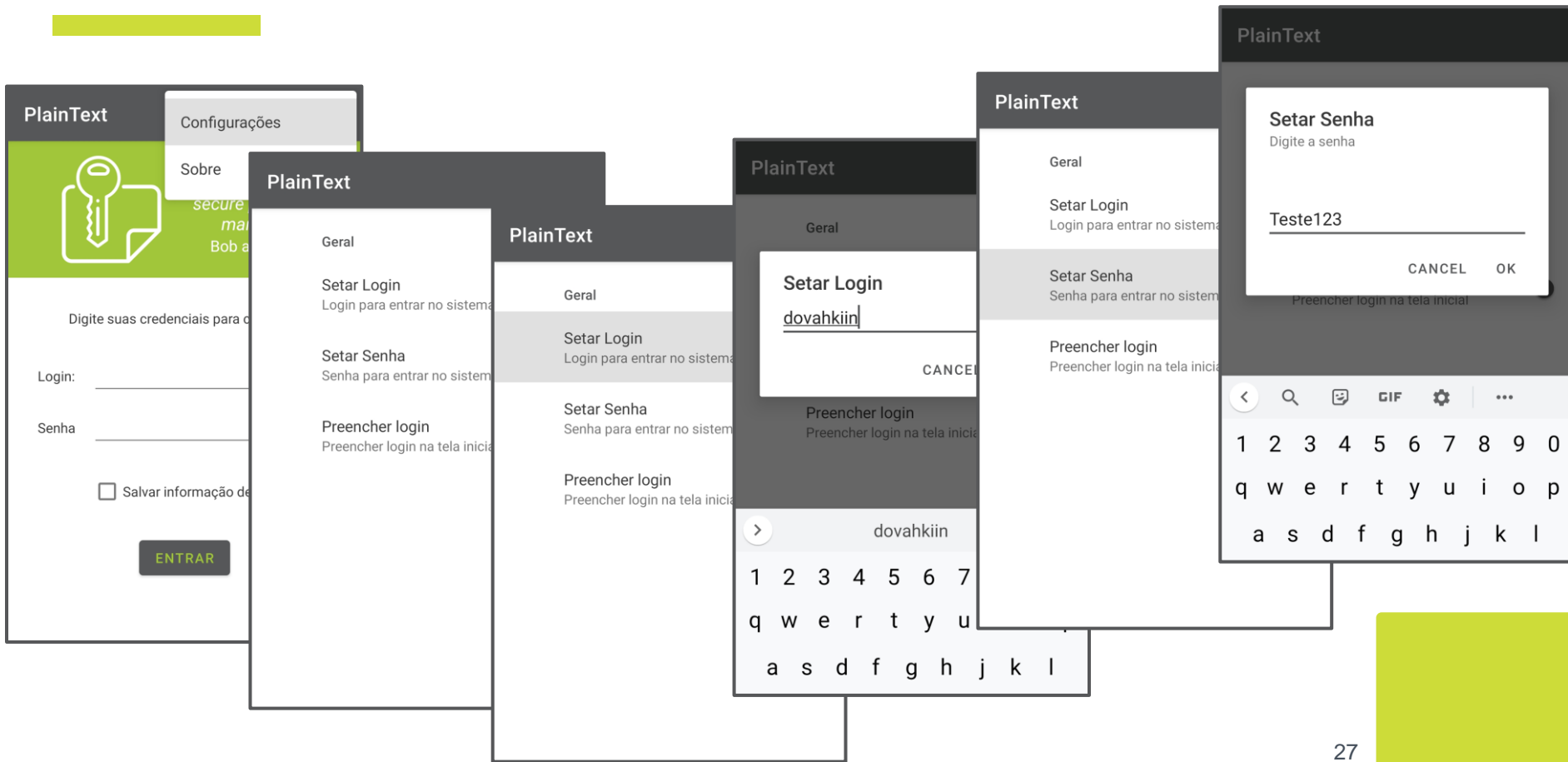
    public static class PreferencesFragment extends PreferenceFragmentCompat {
        @Override
        public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
            setPreferencesFromResource(R.xml.preferences, rootKey);
        }
    }
}
```

Iniciando a Activity de Preferências

- Para iniciar as configurações a partir da tela inicial, precisamos modificar o método `onOptionsItemSelected`:

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.about:  
            AlertDialog.Builder alert = new AlertDialog.Builder(this);  
            alert.setMessage("PlainText Password Manager v1.0")  
                .setNeutralButton("Ok", null).show();  
            return true;  
  
        case R.id.configs:  
            Intent intentConfig = new Intent(this, PreferencesActivity.class);  
            startActivity(intentConfig);  
            return true;  
  
        default:  
            return super.onOptionsItemSelected(item);  
    }  
}
```

Testando as Preferências



Acessando as Preferências

Acessa as
preferências

Acessa o
login salvo

Acessa a
senha salva


```
public void entrarClicado(View view) {  
    SharedPreferences sharedPreferences =  
    PreferenceManager.getDefaultSharedPreferences(this);  
    String prefLogin = sharedPreferences.getString("login", "");  
    String prefPass = sharedPreferences.getString("password", "");  
  
    String editLogin = ((EditText) findViewById(R.id.editLogin)).getText().toString();  
    String editPass = ((EditText) findViewById(R.id.editPassword)).getText().toString();  
  
    if (editLogin.equals(prefLogin) && editPass.equals(prefPass)) {  
        Intent intent = new Intent(this, ListActivity.class);  
        EditText inputLogin = findViewById(R.id.editLogin);  
        intent.putExtra("login", inputLogin.getText().toString());  
        startActivity(intent);  
    }  
    else  
        Toast.makeText(this, "Login/senha inválidos!", Toast.LENGTH_SHORT).show();  
}
```

Acessa os
campos de texto
de login e senha

Verifica o
login/senh
a

Acessando as Preferências

PlainText



"The most secure password manager"
Bob and Alice

Digite suas credenciais para continuar

Login:


Senha:

☐ Salvar informação de login

ENTRAR

Login/senha inválidos!

PlainText



"The most secure password manager"
Bob and Alice

Digite suas credenciais para continuar

Login:

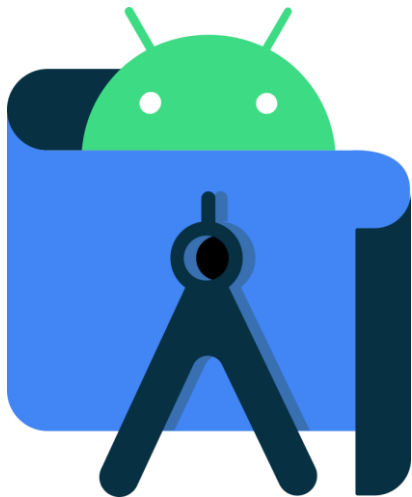
Senha:

☐ Salvar informação de login

ENTRAR

PlainText

Olá dovahkiin!



Challenging

Prof. Diogo Soares



Tela de login com salvar informacoes de login

- Tentem implementar uma rotina que, dado o checkbox de salvar dados já esteja marcado, faça:
 - Salve o login e senha usando XML de preferências
 - Crie uma função que retorne se o login e senha já estão salvos no XML de preferências
 - Faça o app logar automaticamente caso já esteja salvo no XML de preferências, ou seja, pule de splash screen para a tela de logado sem passar pela tela de login