

2º DAM

*Programación Multimedia y
Dispositivos Móviles*

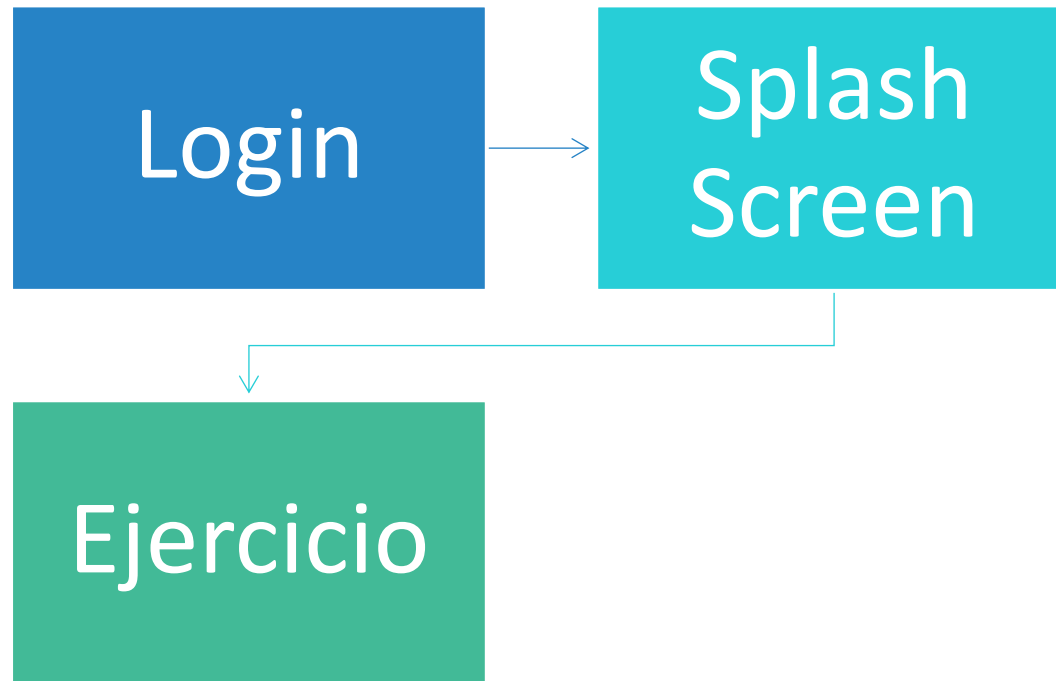
Programación Android



android

José A. Lara

Contenido





Login

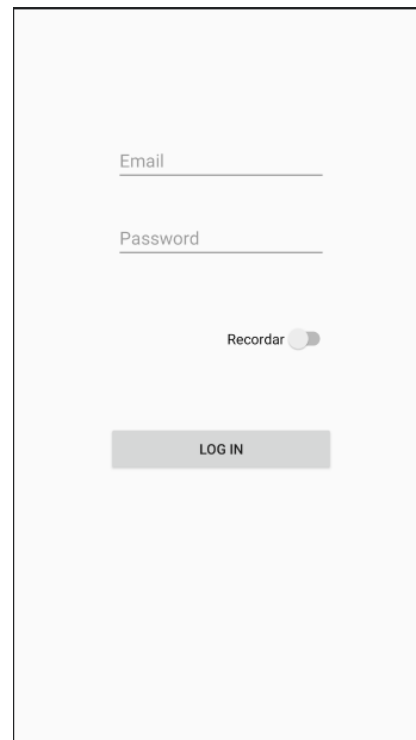


android

Programación Android

Login

Vamos a crear una app para ilustrar los conceptos de esta unidad. Creamos un nuevo proyecto y además de MainActivity creamos una nueva Activity (ActivityLogin). Diseñamos el layout activity_login.

A vertical login form layout on a light gray background. It features two text input fields: the first is labeled 'Email' and the second is labeled 'Password'. Below the password field is a toggle switch labeled 'Recordar'. At the bottom of the form is a gray rectangular button with the text 'LOG IN' in white capital letters.

Email

Password

Recordar ☐

LOG IN



Programación Android

Login

En cuanto al código Kotlin o Java, tenemos que identificar las vistas del layout y realizar algunas comprobaciones. Para esta tarea, contamos con clases como TextUtils que nos proporciona métodos estáticos como isEmpty(), y la clase Patterns que nos ofrece la posibilidad de validar las reglas básicas de una dirección de email correcta.

```
private fun eMailValido(email: String): Boolean {  
    return !TextUtils.isEmpty(email) && Patterns.EMAIL_ADDRESS.matcher(email).matches()  
}
```

```
private boolean eMailValido(String email){  
    return !TextUtils.isEmpty(email) && Patterns.EMAIL_ADDRESS.matcher(email).matches();  
}
```



Programación Android

Login

A continuación, configuramos las Shared Preferences, para guardar los datos de acceso, de forma que cuando entremos en la app no tengamos que especificar el email y la password.

```
private fun guardarPreferencias(email: String, password: String) {  
    val editor = prefs!!.edit()  
    if (swRecordar!!.isChecked) {  
        editor.putString("email", email)  
        editor.putString("password", password)  
        editor.putBoolean("recordar", true)  
        editor.apply()  
    } else {  
        editor.clear()  
        editor.putBoolean("recordar", false)  
        editor.apply()  
    }  
}
```

```
private void guardarPreferencias(String email, String password){  
    SharedPreferences.Editor editor=prefs.edit();  
    if (swRecordar.isChecked()){  
        editor.putString( s: "email",email);  
        editor.putString( s: "password",password);  
        editor.putBoolean( s: "recordar", b: true);  
        editor.apply();  
    } else {  
        editor.clear();  
        editor.putBoolean( s: "recordar", b: false);  
        editor.apply();  
    }  
}
```



Programación Android

Login

```
class LoginActivity : AppCompatActivity() {
    private var etEmail: EditText? = null
    private var etPassword: EditText? = null
    private var swRecordar: Switch? = null
    private var prefs: SharedPreferences? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)
        etEmail = findViewById(R.id.etEmail)
        etPassword = findViewById(R.id.etPassword)
        val btLogin: Button = findViewById(R.id.btLogin)
        swRecordar = findViewById(R.id.swRecordar)
        prefs = getSharedPreferences( name: "app", MODE_PRIVATE)
        establecerValoresSiExisten()
        btLogin.setOnClickListener { it: View!
            val email=etEmail!!.text.toString()
            val password=etPassword!!.text.toString()
            if (login(email, password)) goToMain()
            guardarPreferencias(email,password)
        }
    }

    private fun eMailValido(email: String): Boolean {
        return !TextUtils.isEmpty(email) && Patterns.EMAIL_ADDRESS.matcher(email).matches()
    }

    private fun passwordValida(password: String): Boolean {
        return !TextUtils.isEmpty(password) && password.length > 7
    }
}
```

```
private fun login(email: String, password: String): Boolean {
    var valido = false
    if (!eMailValido(email)) {
        Toast.makeText( context: this,
            text: "e-mail no válido. Introduzca un e-mail correcto",
            Toast.LENGTH_SHORT).show()
    } else if (!passwordValida(password)) {
        Toast.makeText( context: this,
            text: "Password no válida. Debe tener al menos 8 caracteres",
            Toast.LENGTH_SHORT).show()
    } else {
        valido = true
    }
    return valido
}

private fun goToMain() {
    val intent = Intent( packageContext: this, MainActivity::class.java)
    //Evita que pasemos de nuevo a la activity login
    intent.flags = Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK
    startActivity(intent)
}
```



Programación Android

Login

```
private fun guardarPreferencias(email: String, password: String) {
    val editor = prefs!!.edit()
    if (swRecordar!!.isChecked) {
        editor.putString("email", email)
        editor.putString("password", password)
        editor.putBoolean("recordar", true)
        editor.apply()
    } else {
        editor.clear()
        editor.putBoolean("recordar", false)
        editor.apply()
    }
}

private fun establecerValoresSiExisten() {
    val email = prefs!!.getString("email", "")
    val password = prefs!!.getString("password", "")
    val recordar = prefs!!.getBoolean("recordar", false)
    if (!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password)) {
        etEmail!!.setText(email)
        etPassword!!.setText(password)
        swRecordar!!.isChecked = recordar
    }
}
```

En MainActivity:

```
override fun onBackPressed() {
    moveTaskToBack( nonRoot: true)
}
```



Programación Android

Login

```
public class LoginActivity extends AppCompatActivity {

    private EditText etEmail;
    private EditText etPassword;
    private Switch swRecordar;
    private SharedPreferences prefs;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        etEmail=findViewById(R.id.etEmail);
        etPassword=findViewById(R.id.etPassword);
        Button btLogin=findViewById(R.id.btLogin);
        swRecordar=findViewById(R.id.swRecordar);
        prefs=getSharedPreferences( name: "app", Context.MODE_PRIVATE);
        establecerValoresSiExisten();
        btLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String email=etEmail.getText().toString();
                String password=etPassword.getText().toString();
                if (login(email,password))
                    goToMain();
                guardarPreferencias(email,password);
            }
        });
    }

    private boolean eMailValido(String email){
        return !TextUtils.isEmpty(email) && Patterns.EMAIL_ADDRESS.matcher(email).matches();
    }
}
```

```
private boolean passwordValida(String password){
    return !TextUtils.isEmpty(password) && password.length()>7;
}

private boolean login(String email, String password){
    boolean valido=false;
    if (!eMailValido(email)){
        Toast.makeText( context: this,
            text: "e-mail no válido. Introduzca un e-mail correcto",
            Toast.LENGTH_SHORT).show();
    } else if (!passwordValida(password)){
        Toast.makeText( context: this,
            text: "Password no válida. Debe tener al menos 8 caracteres",
            Toast.LENGTH_SHORT).show();
    } else {
        valido=true;
    }
    return valido;
}

private void goToMain(){
    Intent intent=new Intent( packageContext: this,MainActivity.class);
    //Evita que pasemos de nuevo a la actividad login
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
    startActivity(intent);
}
```



Programación Android

Login

```
private void guardarPreferencias(String email, String password){
    SharedPreferences.Editor editor=prefs.edit();
    if (swRecordar.isChecked()){
        editor.putString( s: "email",email);
        editor.putString( s: "password",password);
        editor.putBoolean( s: "recordar", b: true);
        editor.apply();
    } else {
        editor.clear();
        editor.putBoolean( s: "recordar", b: false);
        editor.apply();
    }
}

private void establecerValoresSiExisten(){
    String email=prefs.getString( s: "email", s1: "");
    String password=prefs.getString( s: "password", s1: "");
    boolean recordar=prefs.getBoolean( s: "recordar", b: false);
    if (!TextUtils.isEmpty(email)&&(!TextUtils.isEmpty(password))){
        etEmail.setText(email);
        etPassword.setText(password);
        swRecordar.setChecked(recordar);
    }
}
}
```

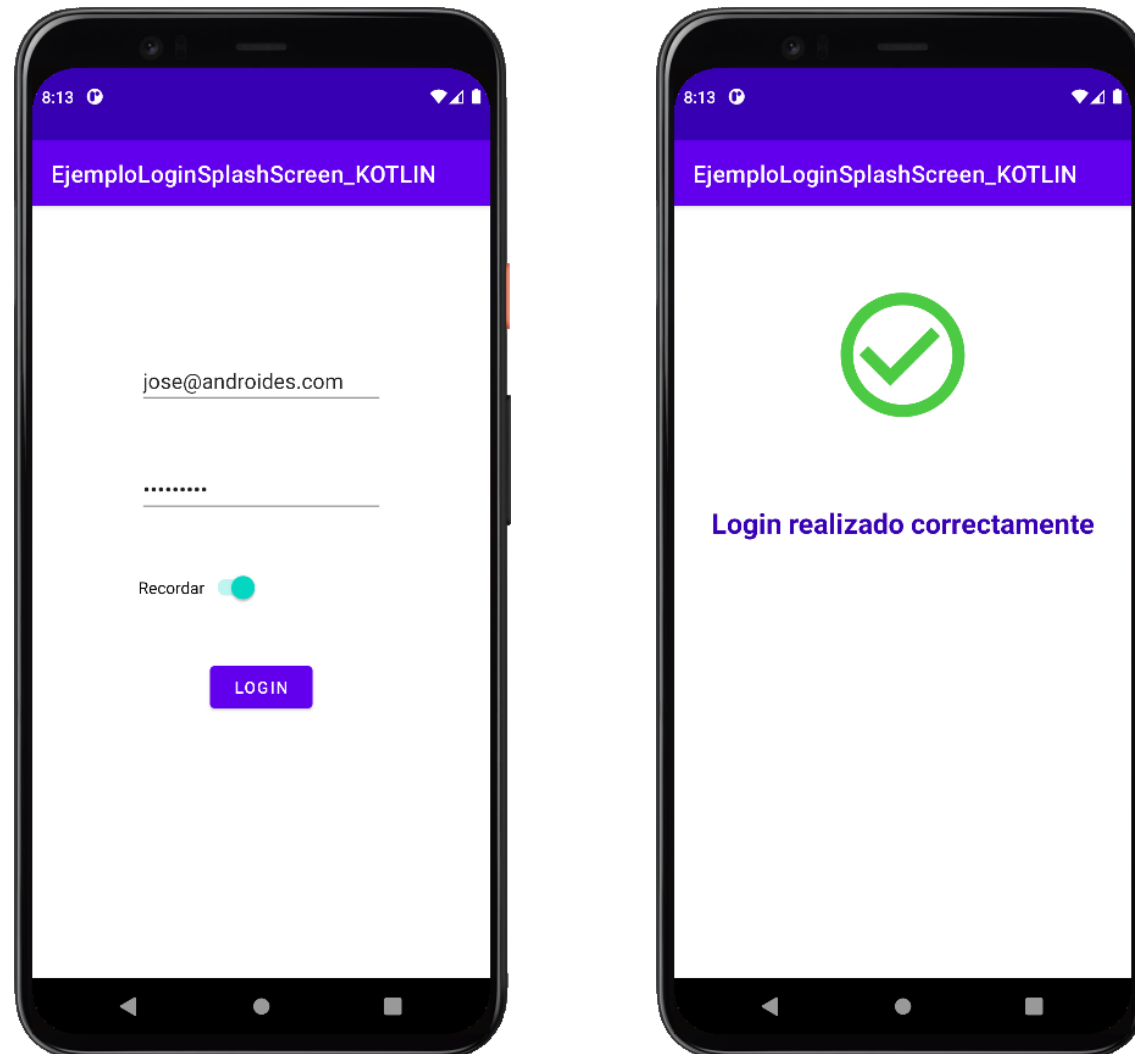
En MainActivity:

```
@Override
public void onBackPressed() {
    moveTaskToBack( nonRoot: true);
}
```



Programación Android

Login



Splash Screen



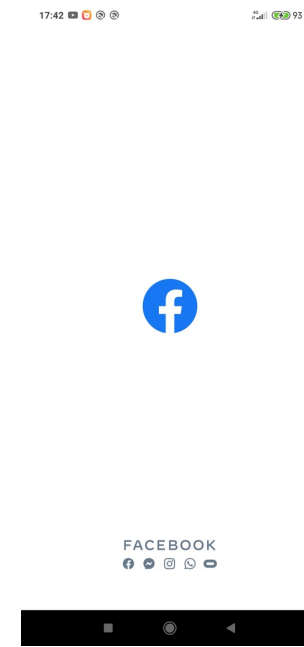
android



Programación Android

Splash Screen

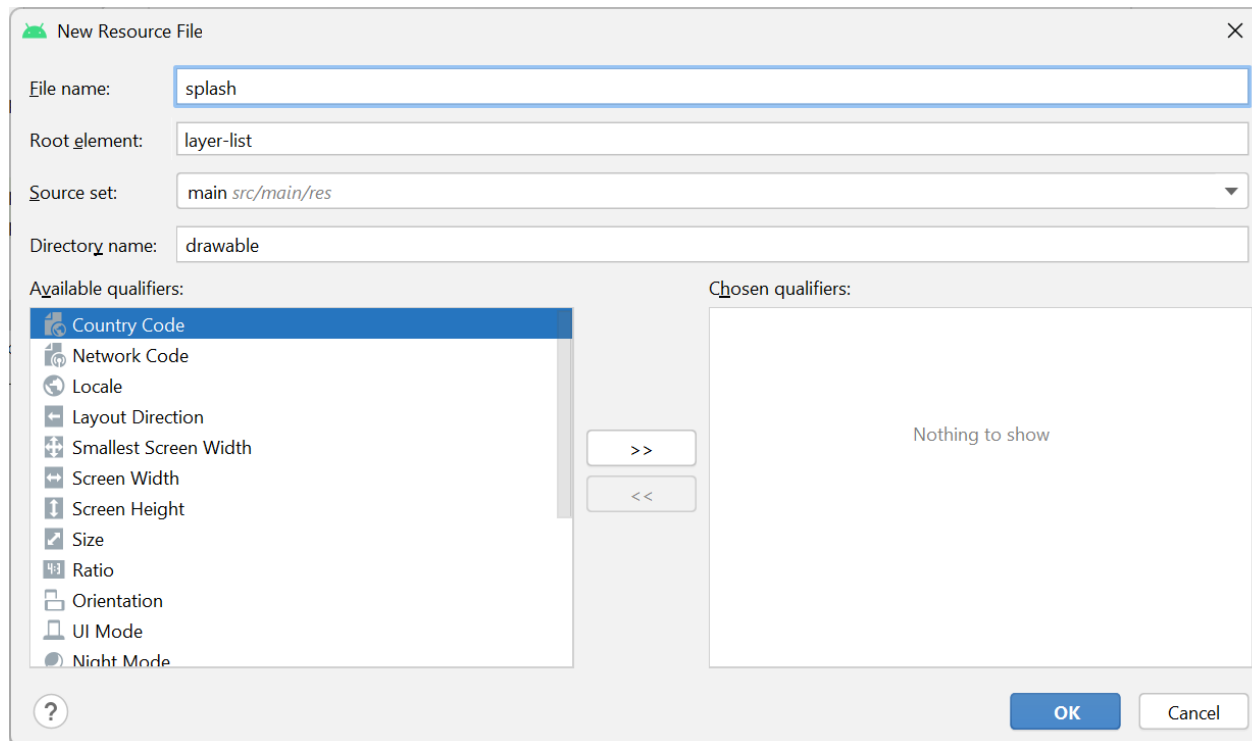
Splash Screen es la primera pantalla visible para el usuario cuando se inicia la aplicación. La pantalla de bienvenida es una de las pantallas más vitales de la aplicación, ya que es la primera experiencia del usuario con la aplicación.



Programación Android

Splash Screen

Crearemos un nuevo Drawable Resource File, y lo llamamos splash.



```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@color/white"/>
    <item>
        <bitmap
            android:gravity="center"
            android:src="@drawable/android_s"/>
        </item>
    </layer-list>
```



Programación Android

Splash Screen

Creamos un nuevo estilo para esta pantalla.

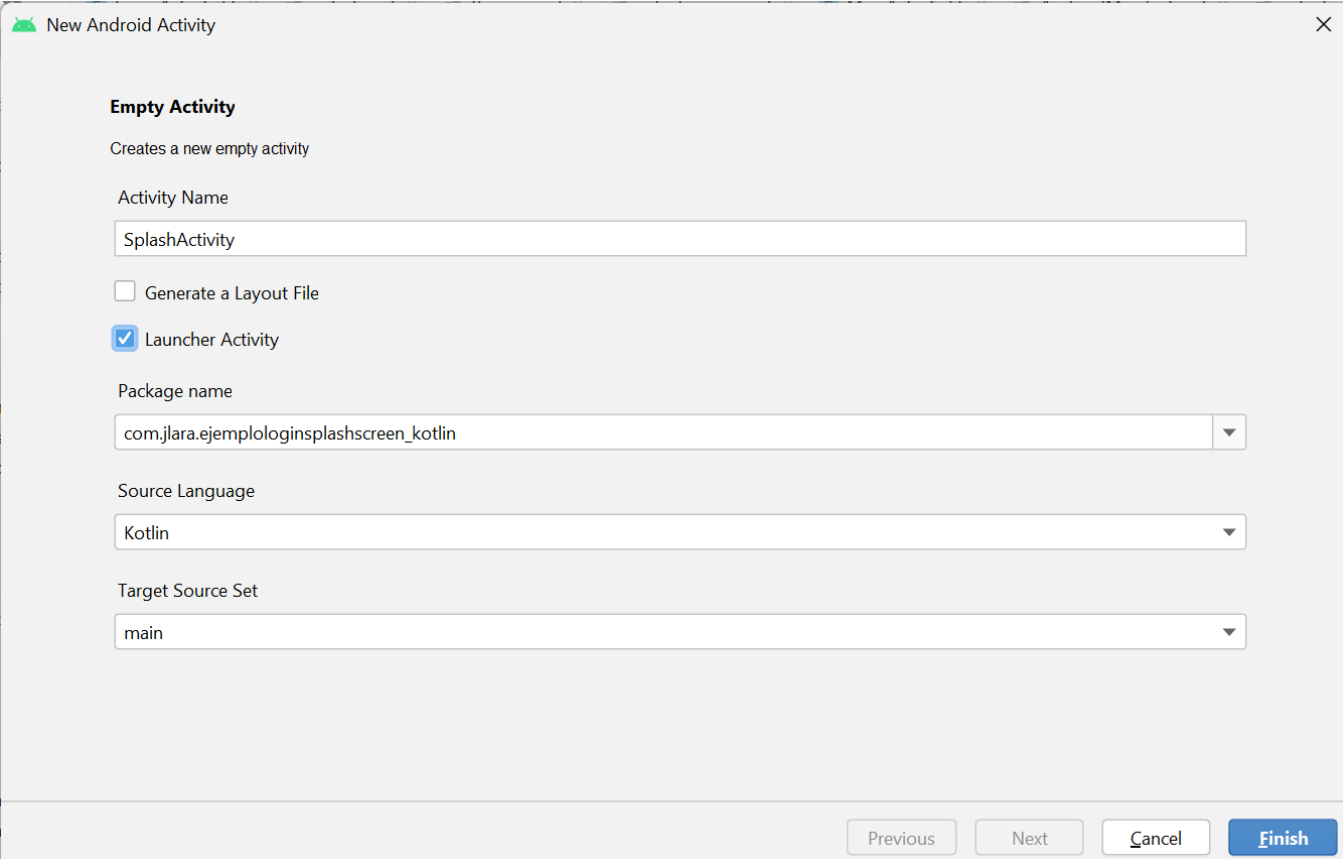
```
<resources xmlns:tools="http://schemas.android.com/tools">
  <!-- Base application theme. -->
  <style name="Theme.EjemploLoginSplashScreen_KOTLIN" parent="Theme.MaterialComponents.DayNight.DarkActionBar">
    <!-- Primary brand color. -->
    <item name="colorPrimary">@color/purple_500</item>
    <item name="colorPrimaryVariant">@color/purple_700</item>
    <item name="colorOnPrimary">@color/white</item>
    <!-- Secondary brand color. -->
    <item name="colorSecondary">@color/teal_200</item>
    <item name="colorSecondaryVariant">@color/teal_700</item>
    <item name="colorOnSecondary">@color/black</item>
    <!-- Status bar color. -->
    <item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>
    <!-- Customize your theme here. -->
  </style>
  <style name="SplashScreen" parent="Theme.MaterialComponents.Light.NoActionBar">
    <item name="android:windowBackground">@drawable/splash</item>
  </style>
</resources>
```



Programación Android

Splash Screen

Creamos una nueva Activity (sin layout).



The screenshot shows the 'New Android Activity' dialog box in Android Studio. The title bar says 'New Android Activity' with a close button. The main content area is titled 'Empty Activity' with a subtitle 'Creates a new empty activity'. Below this, there are several fields and checkboxes:

- Activity Name:** A text field containing 'SplashActivity'.
- Generate a Layout File:** An unchecked checkbox.
- Launcher Activity:** A checked checkbox.
- Package name:** A text field containing 'com.jlara.ejemplologinsplashscreen_kotlin'.
- Source Language:** A dropdown menu set to 'Kotlin'.
- Target Source Set:** A dropdown menu set to 'main'.

At the bottom of the dialog, there are four buttons: 'Previous' (disabled), 'Next' (disabled), 'Cancel', and 'Finish' (highlighted in blue).



Programación Android

Splash Screen

En AndroidManifest.xml quitamos como LAUNCHER a LoginActivity y ponemos y le asignamos el tema a SplashActivity.

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.EjemploLoginSplashScreen_KOTLIN">
    <activity
        android:name=".SplashActivity"
        android:exported="true"
        android:theme="@style/SplashScreen">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

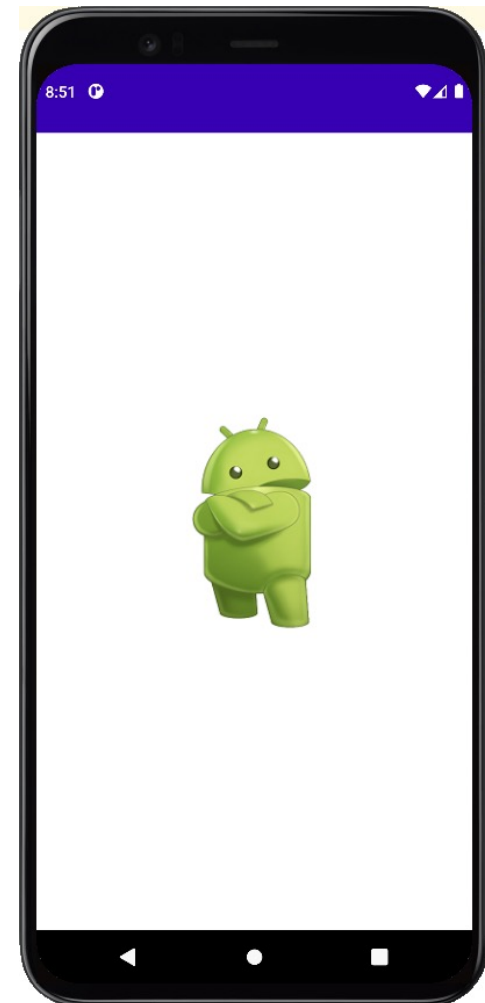
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".MainActivity"
        android:exported="false" />
    <activity
        android:name=".LoginActivity"
        android:exported="true"/>
</application>
```



Programación Android

Splash Screen

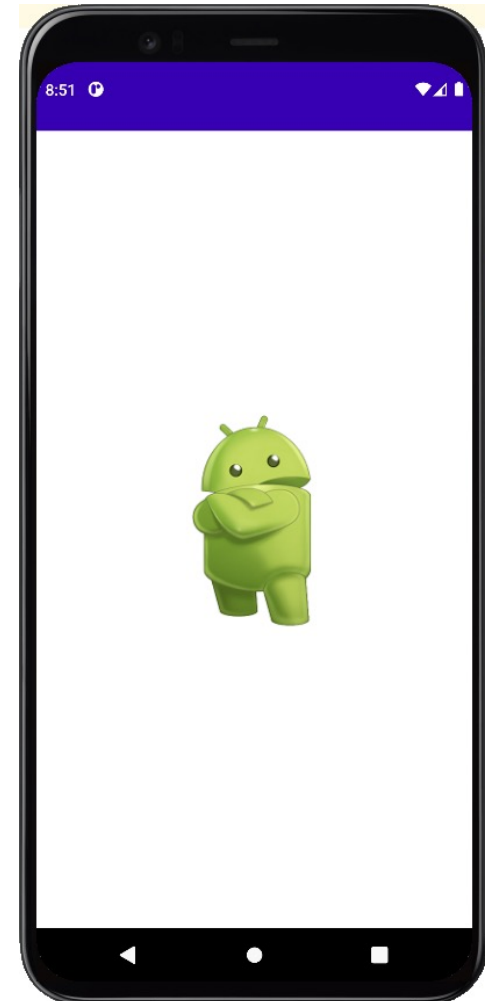
```
class SplashActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setTheme(R.style.Theme_EjemploLoginSplashScreen_KOTLIN)  
        val intent:Intent= Intent( packageContext: this, LoginActivity::class.java)  
        intent.flags=Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK  
        startActivity(intent)  
    }  
}
```



Programación Android

Splash Screen

```
public class SplashActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setTheme(R.style.Theme_EjemploLoginSplashScreen_KOTLIN);  
        Intent intent = new Intent( packageContext: this, LoginActivity.class);  
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_CLEAR_TASK);  
        startActivity(intent);  
    }  
}
```



Programación Android

Splash Screen

Con lo que hemos hecho hasta ahora, la Splash Screen aparecerá y desaparecerá muy rápido, casi no nos da tiempo a verla. Para "retrasar" algo el inicio y que se pueda ver, creamos una nueva clase MyApp en la que vamos a incluir un retraso, y lo añadimos en el AndroidManifest.xml

```
class MyApp: Application() {  
    override fun onCreate() {  
        super.onCreate()  
        SystemClock.sleep( ms: 3000)  
    }  
}
```

```
<application  
    android:name=".MyApp"  
    android:allowBackup="true"  
    android:icon="@mipmap/ic_launcher"  
    android:label="@string/app_name"  
    android:roundIcon="@mipmap/ic_launcher_round"  
    android:supportsRtl="true"  
    android:theme="@style/Theme.EjemploLoginSplashScreen_KOTLIN">  
    <activity  
        android:name=".SplashActivity"  
        android:exported="true"  
        android:theme="@style/SplashScreen">  
        <intent-filter>  
            <action android:name="android.intent.action.MAIN" />  
  
            <category android:name="android.intent.category.LAUNCHER" />  
        </intent-filter>  
    </activity>  
    <activity  
        android:name=".MainActivity"  
        android:exported="false" />  
    <activity  
        android:name=".LoginActivity"  
        android:exported="true"/>  
</application>
```



Programación Android

Splash Screen

Con lo que hemos hecho hasta ahora, la Splash Screen aparecerá y desaparecerá muy rápido, casi no nos da tiempo a verla. Para "retrasar" algo el inicio y que se pueda ver, creamos una nueva clase MyApp en la que vamos a incluir un retraso, y lo añadimos en el AndroidManifest.xml

```
public class MyApp extends Application {  
    @Override  
    public void onCreate() {  
        super.onCreate();  
        SystemClock.sleep( ms: 3000);  
    }  
}
```

```
<application  
    android:name=".MyApp"  
    android:allowBackup="true"  
    android:icon="@mipmap/ic_launcher"  
    android:label="@string/app_name"  
    android:roundIcon="@mipmap/ic_launcher_round"  
    android:supportsRtl="true"  
    android:theme="@style/Theme.EjemploLoginSplashScreen_KOTLIN">  
    <activity  
        android:name=".SplashActivity"  
        android:exported="true"  
        android:theme="@style/SplashScreen">  
        <intent-filter>  
            <action android:name="android.intent.action.MAIN" />  
  
            <category android:name="android.intent.category.LAUNCHER" />  
        </intent-filter>  
    </activity>  
    <activity  
        android:name=".MainActivity"  
        android:exported="false" />  
    <activity  
        android:name=".LoginActivity"  
        android:exported="true"/>  
</application>
```



Programación Android

Splash Screen

Ahora vamos a modificar el inicio de la app de forma que, si tenemos el email y la password en un archivo SharedPreferences, directamente lo rescatamos y pasamos a MainActivity, mientras que si no tenemos los datos pasaremos a LoginActivity.

```
class SplashActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setTheme(R.style.Theme_EjemploLoginSplashScreen_KOTLIN)  
        val intentLogin:Intent= Intent( packageContext: this,LoginActivity::class.java)  
        intentLogin.flags=Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK  
        val intentMain:Intent= Intent( packageContext: this,MainActivity::class.java)  
        intentMain.flags=Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK  
        val prefs:SharedPreferences=getSharedPreferences( name: "app", Context.MODE_PRIVATE)  
        if ((!TextUtils.isEmpty(prefs.getString("email",""))) &&  
            (!TextUtils.isEmpty(prefs.getString("password",""))))  
            startActivity(intentMain)  
        else  
            startActivity(intentLogin)  
        finish()  
    }  
}
```



Programación Android

Splash Screen

Ahora vamos a modificar el inicio de la app de forma que, si tenemos el email y la password en un archivo SharedPreferences, directamente lo rescatamos y pasamos a MainActivity, mientras que si no tenemos los datos pasaremos a LoginActivity.

```
public class SplashActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setTheme(R.style.Theme_EjemploLoginSplashScreen_KOTLIN);  
        Intent intentLogin = new Intent( packageContext: this, LoginActivity.class);  
        intentLogin.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_CLEAR_TASK);  
        Intent intentMain = new Intent( packageContext: this, MainActivity.class);  
        intentMain.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_CLEAR_TASK);  
        SharedPreferences prefs=getSharedPreferences( name: "app", Context.MODE_PRIVATE);  
        if ((!TextUtils.isEmpty(prefs.getString( s: "email", s1: "")))&&  
            (!TextUtils.isEmpty(prefs.getString( s: "password", s1: ""))))  
            startActivity(intentMain);  
        else  
            startActivity(intentLogin);  
        finish();  
    }  
}
```



Ejercicio



android



Programación Android

Ejercicio

Añadir a la aplicación desarrollada para las Comunidades una activity de login en la que vamos a hacer uso de las shared preferences y una Activity SplashScreen.

La idea es mejorar la app que se ha ido desarrollando de la siguiente forma:

- Si cerramos sesión y las credenciales del login han sido guardadas, activaremos el switch button para que esté marcado como activo.
- El fondo del SplashScreen será blanco, para una mejor transacción visual hacia la siguiente pantalla. (Debemos añadir ese color al archivo de colores pertinente).
- Cuando accedemos a la app, si nos hemos registrado previamente y tenemos los datos guardados en las Shared Preferences, accederemos directamente al listado de comunidades.



Programación Android

Ejercicio

