


Open   Opened 2 days ago by  **Rubén Montero**

# Pantalla de inicio de sesión

## Resumen

- Crearemos una nueva actividad `LoginActivity.java`
- La escogeremos como `LAUNCHER` y `MAIN` en el manifiesto, para que sea la actividad inicial
- Añadiremos *dos* `<EditText>` y *dos* `<Button>`
- Conectaremos `LoginActivity` con `RegisterActivity`

## Descripción

Ya tenemos una pantalla de registro. pero, ¿acaso es lo que ves en las *apps* que usas habitualmente?

Normalmente la pantalla inicial te permite hacer *Login*, ¿verdad? Tiene lógica. Si un usuario se descarga tu *app* es posible que ya tenga una cuenta creada en tu plataforma. Por otro lado, el proceso de registro sólo se lleva a cabo una vez.

¡Vamos a empezar a trabajar en una pantalla de *Login*! De momento, sólo en la parte visual.

## La tarea

**Añade** un nuevo archivo de *layout* `activity_login.xml` . Tendrá un `<ConstraintLayout>` como elemento raíz, y dentro:

- **Añade** un `<EditText>` (será para la *contraseña*). Debe tener:
  - Ancho proporcional al 60% del contenedor.
  - Alto ajustado al contenido ( `wrap_content` ).
  - Posición centrada, vertical y horizontalmente.
  - `android:inputType="textPassword"` .
  - `android:hint="Contraseña"` .
  - Un `android:id` escogido por ti.
- **Añade** otro `<EditText>` (será para el *nombre de usuario*). Debe tener:
  - Ancho proporcional al 60% del contenedor.
  - Alto ajustado al contenido ( `wrap_content` ).
  - Posicionado *sobre* (encima) el `<EditText>` anterior.
  - `android:hint="alice.doe"` .
  - Un `id` escogido por ti.
- **Añade** un `<Button>` , que servirá para *iniciar sesión*. Debe tener:
  - *Constraints* que lo peguen abajo, a la izquierda y a la derecha del contenedor.
  - Ancho ajustado a las *constraints* ( `0dp` ).
  - Alto ajustado al contenido ( `wrap_content` ).
  - Indicará el texto `"Loguearse"` .
  - Un `id` escogido por ti.
- **Añade** un `<Button>` que servirá para *ir a la pantalla de registro*. Debe tener:
  - *Constraints* que lo peguen, a la izquierda y a la derecha del contenedor. También, estará posicionado *encima* del `<Button>` anterior.
  - Ancho ajustado a las *constraints* ( `0dp` ).
  - Alto ajustado al contenido ( `wrap_content` ).
  - Indicará el texto `"¿No tienes cuenta?"` ,

Con esto, habrás terminado el trabajo en el archivo de interfaz `activity_login.xml` .

Luego, **crea** una nueva clase Java `LoginActivity.java` . **Escribe** la cláusula para que *herede* de `AppCompatActivity` , e **implementa** el método `onCreate` así:

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
}
```

Después, **edita** el archivo `AndroidManifest.xml`. **Añade** una nueva declaración para la actividad recién creada, indicando los `<intent-filter>` `LAUNCHER` y `MAIN`. **Así**:

```
<activity
    android:name=".LoginActivity"
    android:exported="true"> <!-- exported debe indicarse como "true" para actividades LAUNCHER desde Android 12 -->
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

- `"android.intent.action.MAIN"` significa que esta actividad es válida como *top-level* para acceder al resto de actividades y no recibe ningún 'Extra'
- `"android.intent.category.LAUNCHER"` significa que es la actividad usada como punto de entrada cuando el usuario abre la aplicación

También, **elimina** dichas declaraciones de `RegisterActivity`:

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

## Conectando la pantalla de *Login* y la de *Registro*

Ahora que ya está creada la actividad de *Login* puedes lanzar la *app* y probarla.

Como últimos pasos de esta tarea, vamos a conectar mejor las pantallas que ya tenemos.

Primero, en `LoginActivity.java`:

- **Añade** un atributo privado de tipo `Button` e **inicialízalo** en `onCreate`, conectado al botón de `"¿No tienes cuenta?"` (debes recordar *qué id* le diste).
- Después, en `onCreate`, **establece** un `OnClickListener` para dicho botón.
- Dentro del `onClick`, **lanza** la `RegisterActivity` usando un `Intent` y `startActivity`, como ya hiciste anteriormente.

¡Si pruebas la *app* deberás poder ir a la pantalla de *Registro* pulsando el botón!

Por último, **abre** `RegisterActivity.java`. Ahí, **invoca** `finish()`; **cuando la petición de registro se completa satisfactoriamente**. No elimines el `Toast` que implementamos anteriormente. Invoca `finish()` en la línea *posterior* al `Toast`.

La idea es que, tras registrarse con éxito, el usuario regrese *automáticamente* a la pantalla de *Login* para iniciar sesión.

Puedes lanzar la *app* y registrarte para verificarlo. ¿Regresas automáticamente a la pantalla de *Login*?

## Por último

Sube tus cambios al repositorio en un nuevo *commit*.



[Rubén Montero @ruben.montero](#) changed milestone to [%Sprint 3](#) 2 days ago