

# FEUD

Fundación de egresados U. Distrital  
*Construyendo Profesionales*



***Somos el Centro de Entrenamiento Autorizado por marcas representativas en Gobierno TI y empresa, con el portafolio más amplio en Latinoamérica:***



Todas las marcas, nombres comerciales, marcas de servicios y logotipos a los que se hace referencia en el presente documento pertenecen a sus respectivas empresas. ITIL® es una marca registrada de AXELOS Limited. El Swirl logo™ es una marca de AXELOS Limited. PMP, PMI, PMI-RMP, PMI-ACP, PMI-SP, PgMP, CAPM, REP Logo son marcas registradas del Project Management Institute.

**FEUD**  
Fundación de egresados U. Distrital  
*Construyendo Profesionales*



# Desarrollo de aplicaciones utilizando SDK - Android

Diplomado Desarrollo de Software para móviles

# Temas

- Activity + Listeners + Intents
- Estilos en Formularios
- Manipulación de sistema de Archivos

# Practica - Objetivos

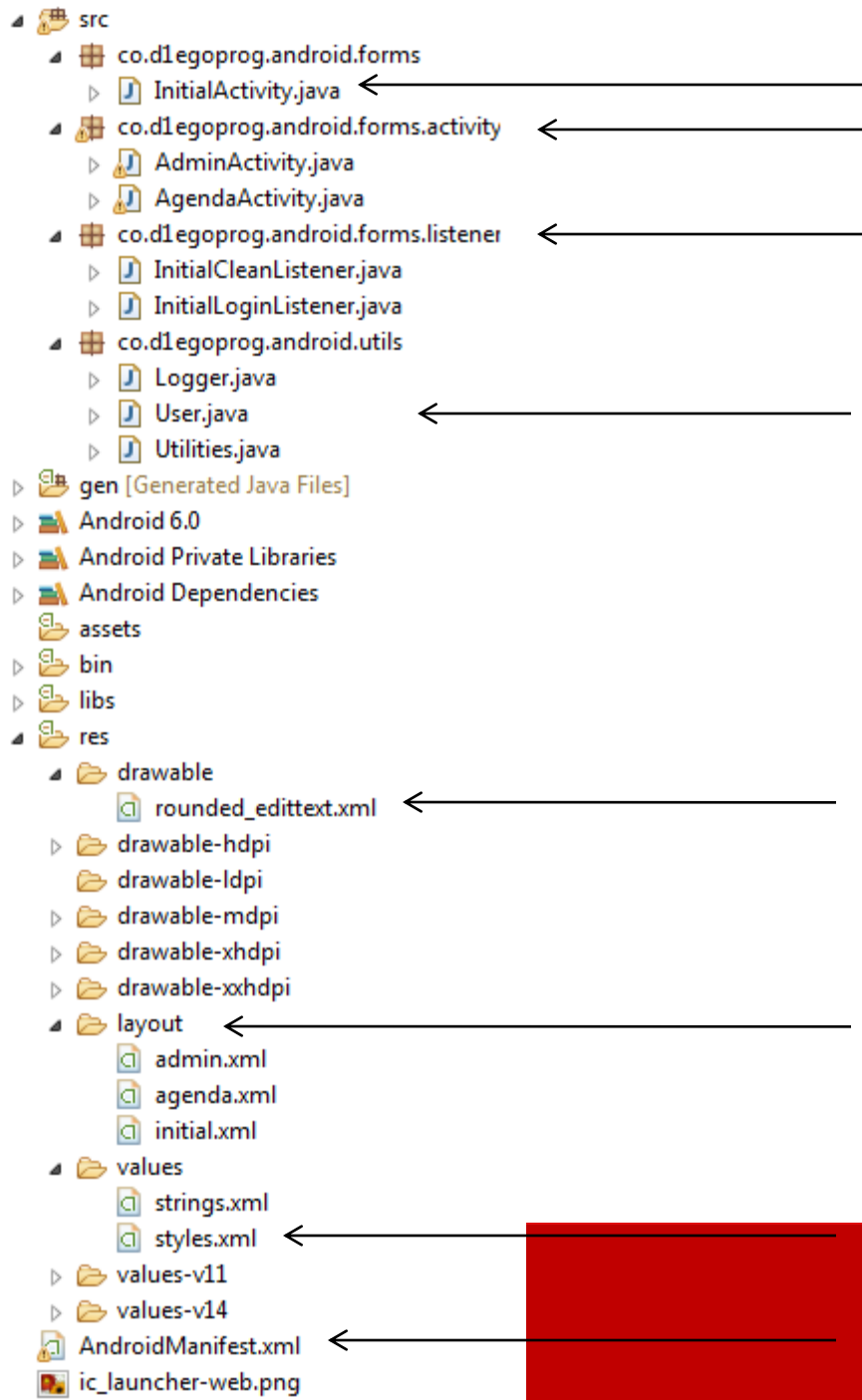
- Estudiar y ejecutar exitosamente el proyecto
- Observar los cambios realizados
- Implementar la Agenda en Proyecto

# Recursos

- Descargar de GitHub el proyecto de Formularios

<https://github.com/d1egoprog/classes/tree/master/msd/2.%20Android%20-%20FormExample>

## ExampleForm



Clase que carga la Actividad Inicial  
Actividad Paquete General de Actividades

Actividad Paquete General de Listeners

Clase describe los Usuarios

Estilo de Texto Redondeado

Listado General de “Pantallas”

Archivo de Estilos

Manifiesto de Aplicación

# AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="co.diegoprogram.android.forms"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk
8         android:minSdkVersion="8"
9         android:targetSdkVersion="21" />
10
11     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
12
13     <application
14         android:allowBackup="true"
15         android:icon="@drawable/ic_launcher"
16         android:label="@string/app_name"
17         android:theme="@style/AppTheme" >
18         <activity
19             android:name=".InitialActivity"
20             android:label="@string/app_name" >
21             <intent-filter>
22                 <action android:name="android.intent.action.MAIN" />
23
24                 <category android:name="android.intent.category.LAUNCHER" />
25             </intent-filter>
26         </activity>
27         <activity android:name=".activity.AdminActivity" />
28         <activity android:name=".activity.AgendaActivity" />
29
30     </application>
31
32 </manifest>
33
```

Se registra que se accederá al sistema de archivos

Se registra las actividades adicionales

# InitialActivity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
    setContentView(R.layout.initial);

    initAppDataStorage();
    initExternalFiles();
    initUsers();

    setButtonClean((Button)this.findViewById(R.id.buttonClean));
    getButtonClean().setOnClickListener(new InitialCleanListener());

    setButtonLogin((Button)this.findViewById(R.id.buttonLogin));
    getButtonLogin().setOnClickListener(new InitialLoginListener(users));
}
```

Se inicializan los recursos  
del sistema de archivos

Obligamos al Android a  
que nos muestre  
horizontalmente la pantalla



# users.properties

(crearlo por medio del Explorador de archivos)

- Compilar y ejecutar la aplicación (1 vez), cuando se ejecuta falla, pero crea la estructura de directorios en la carpeta Download del dispositivo (revisar manualmente)
- Crear el archivo en la carpeta forms/system/user.properties

users.number=3      ← Cantidad de N Usuarios

user1.login=user

user1.password=user

user1.role=user

user2.login=admin

user2.password=admin

user2.role=admin

user3.login=user2

user3.password=user2

user3.role=user

Información de usuario N

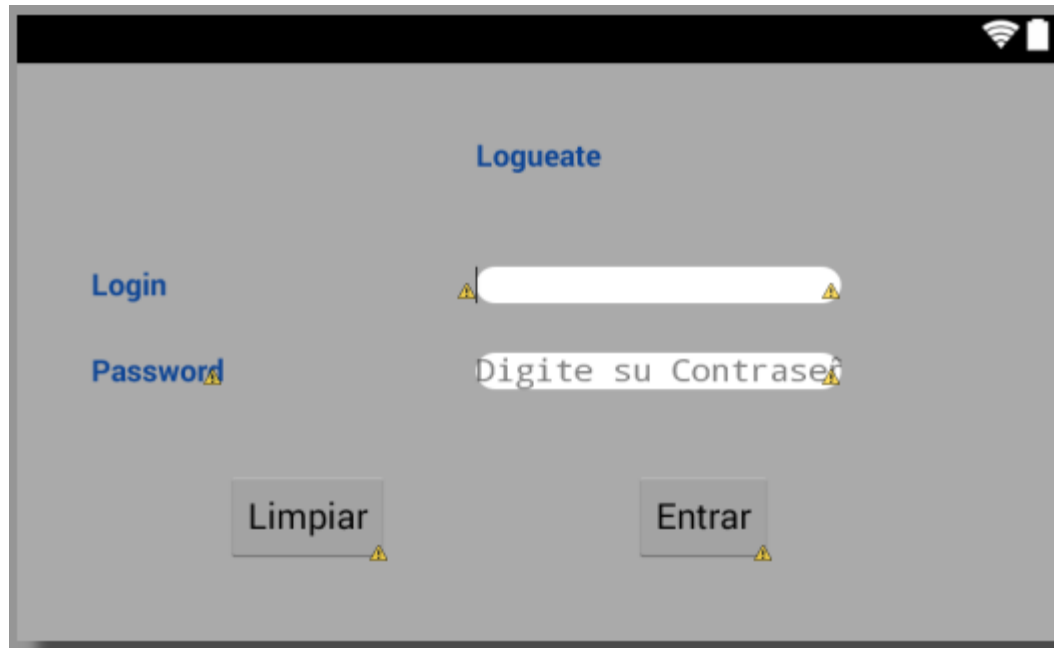
# InitialLoginListener.java

```
public void onClick(View v) {  
    setEditLogin((EditText) v.findViewById(R.id.editLogin));  
    setEditPassword((EditText) v.findViewById(R.id.editPassword));  
  
    if ((User)users.get(getEditLogin().getText().toString()) == null){  
        getEditLogin().setText("");  
        getEditPassword().setText("");  
        Utilities.messageBox(v.getContext(), "El Usuario no existe");  
        return;  
    }  
  
    User user = (User)users.get(getEditLogin().getText().toString());  
  
    if (user.getPassword().equals(getEditPassword().getText().toString())){  
        if (user.getRole().equals("admin")){  
            Intent intentStart = new Intent(v.getContext(), AdminActivity.class);  
            v.getContext().startActivity(intentStart);  
        }else{  
            Intent intentStart = new Intent(v.getContext(), Object.class);  
            v.getContext().startActivity(intentStart);  
        }  
    }else{  
        getEditLogin().setText("");  
        getEditPassword().setText("");  
        Utilities.messageBox(v.getContext(), "Error de Contraseña, Vuelva a Intentarlo");  
    }  
}
```

Preguntamos al grupo de usuarios si existe el login

Si existe el Login, Valida la contraseña y el Rol del Usuario

# Initial.xml



A screenshot of a mobile application login screen. The interface has a grey background. At the top, there is a black status bar with a white Wi-Fi icon and a battery icon. The title "Logueate" is centered at the top in blue text. Below the title, there are two labels: "Login" and "Password", both in blue text. To the right of "Login" is a white text input field with a yellow warning icon on the left and a yellow warning icon on the right. To the right of "Password" is a white text input field with the placeholder text "Digite su Contraseña" and a yellow warning icon on the right. At the bottom, there are two buttons: "Limpiar" on the left and "Entrar" on the right, both in black text on a grey background. Each button has a yellow warning icon at the bottom right corner.

# Initial.xml

```
<TextView
    android:id="@+id/textView2"
    style="@style/labelStyle" ←
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/textView1"
    android:layout_marginLeft="38dp"
    android:layout_marginTop="47dp"

    android:text="@string/initialLogin" />
```

Creamos estilos para todos los TextView values/styles.xml

```
<EditText
    android:id="@+id/editLogin"
    style="@style/editStyle" ←
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView1"
    android:layout_alignTop="@+id/textView2" ←
    android:hint="@string/initialHintLogin" ←
    android:background="@drawable/rounded_edittext" ←
    android:ems="10" >

    <requestFocus /> ←
</EditText>
```

Creamos estilos para todos los EditText

Regalamos una pista para el Usuario

Añadimos Mascara visual res/drawable/rounded\_e...

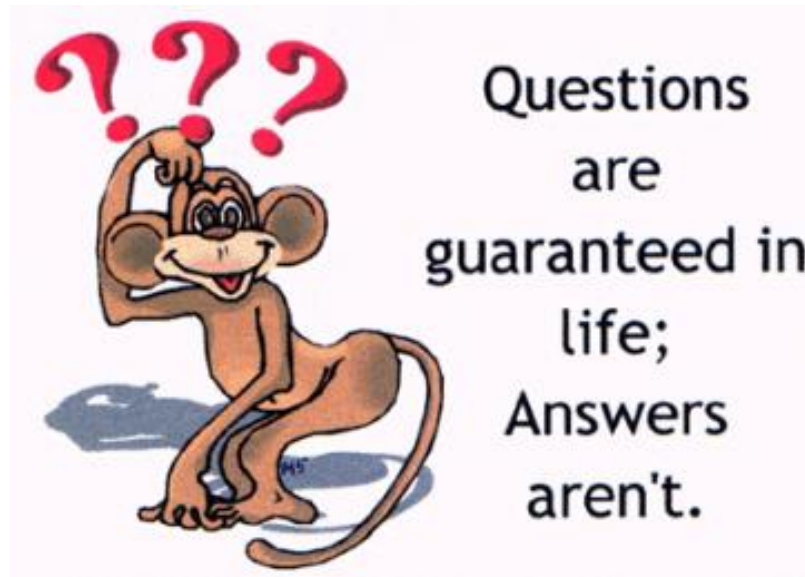
Pedimos el foco cuando inicia el formulario

# Asignación

- ✓ Crear un nuevo usuario y validarlo.
- ✓ Implementar el Create de la Agenda desarrollada web en formulario Android con lo expuesto en el ejemplo.
- ✓ [adicional] Persistir los usuarios en un CSV en la carpeta forms/data (revisar clase Utilities.java).

# Continuación?

- ✓ Acceso a la Cámara.
- ✓ Acceso a la Base de Datos via HTTP



# ¿Preguntas?

Diego Alberto Rincón Yáñez MSc.  
Twitter: @d1egoprog.