



Ambiente de desarrollo y sus elementos de configuración

Elementos de un proyecto Android (Parte I)

Implementar una interfaz de usuario utilizando buenas prácticas en el manejo de estilos para brindar un aspecto visual e interacciones acordes a lo requerido

- **Unidad 1:** Ambiente de desarrollo y sus elementos de configuración.
- **Unidad 2:** Elementos de la interfaz, navegación e interacción.
- **Unidad 3:** Fundamentos de GIT y GitHub.



Te encuentras aquí



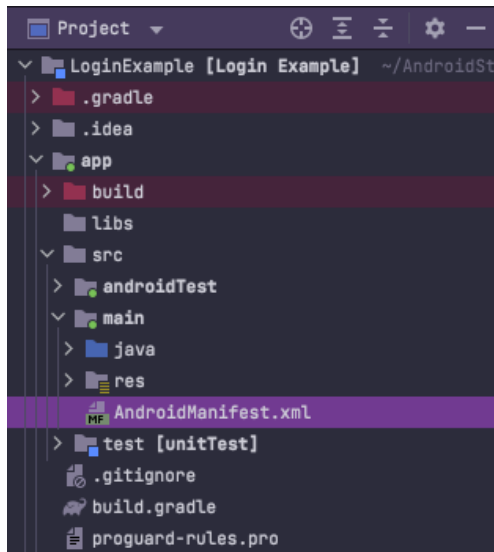
¿Qué aprenderás en esta sesión?

Identifica los elementos de configuración principales de un proyecto Android

`/* Manifest */`

AndroidManifest

¿Qué es?



Describe información esencial de la app para el SO y Google Play:

- Nombra el paquete de la app que la identifica como única.
- Describe los componentes de la app.
- Declara los permisos que necesita la app.
- Por convención debe llamarse **AndroidManifest.xml**.

XML

¿Qué es?

- Es un **lenguaje de etiquetas** desarrollado por el World Wide Web Consortium ([W3C](#)). No pertenece a ninguna compañía.
- Se utiliza para almacenar datos en forma legible para humanos.
- Se enfoca en el sentido semántico de los datos, dejando de lado la presentación.
- Las etiquetas representan una jerarquía de conceptos anidados.

{desafío}
latam_

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>
    <text>Don't forget
me!</text>
    <alarm>Sunday 23,
06:00</alarm>
  </body>
</note>
```

Ejemplo de AndroidManifest.xml

- **android:icon** configura el ícono que identifica la app (se muestra en el escritorio del dispositivo).
- **android:label** indica el nombre de fantasía de la app (aparece con ese nombre en el teléfono).

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cl.dal.loginexample">

    <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/AppTheme">
        <activity android:name=".ui.login.LoginActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Ejemplo de AndroidManifest.xml

- `android:theme` indica el tema (colores) que se está utilizando
- `<activity android:name...>` indica que la actividad LoginActivity es parte de esta app, y que además corresponde al punto de entrada de la app.
 - De necesitar algún permiso se debe declarar dentro de `<application>`

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cl.dal.loginexample">

    <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/AppTheme">
        <activity android:name=".ui.login.LoginActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```


/* Assets */

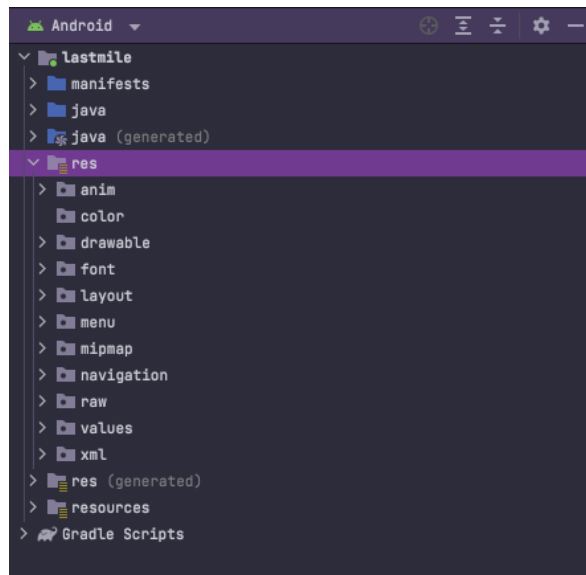
Assets

¿Qué son?

Los recursos son archivos adicionales con contenido estático que se almacenan en el directorio res/. En tiempo de ejecución, es Android el que hace uso de los recursos apropiados basado en la configuración actual del dispositivo.

Algunos de los tipos de recursos soportados son:

- Animación
- Lista de estados de colores
- Drawable (bitmaps o XML)
- Layout
- Menú
- String
- Style
- Font



Estructura XML de un asset

<resources>

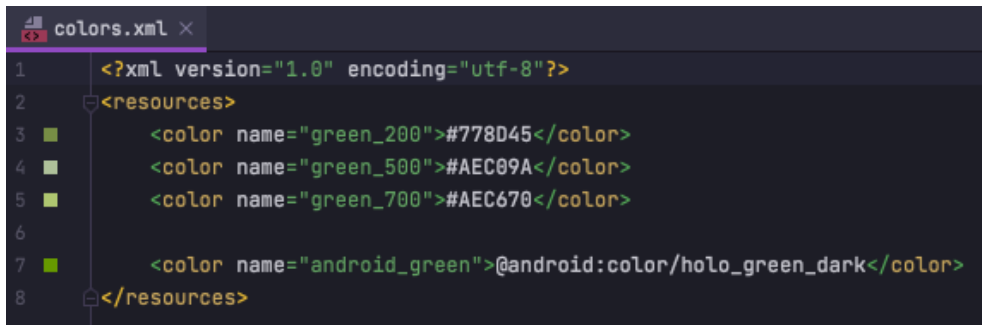
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  ....
</resources>
```

<resources> es el elemento raíz de los archivos XML de los assets (directorio /res).

Se incluye como cabecera la versión de XML que está utilizando y el encoding correspondiente.

Colors

res/values/colors.xml



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <color name="green_200">#778D45</color>
4     <color name="green_500">#AEC09A</color>
5     <color name="green_700">#AEC670</color>
6
7     <color name="android_green">@android:color/holo_green_dark</color>
8 </resources>
```

Cada color definido puede ser reutilizado en cualquier componente de la app.

Hay colores como el **colorPrimary**, que es el principal de la aplicación, que son utilizados por varios componentes nativos como la barra de aplicaciones para su color de fondo.

Todos los colores de la app deben estar definidos en este archivo.

Se definen en formato hexadecimal o referenciando otro recurso.



Dimens

res/values/dimens.xml

Permite reutilizar y estandarizar las dimensiones utilizadas para definir márgenes entre los elementos de la pantalla, padding para espaciar el contenido de un elemento o el tamaño para presentar una imagen.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <dimen name="margin_normal">16dp</dimen>
  <dimen name="margin_small">8dp</dimen>
  <dimen name="padding_default">7dp</dimen>
</resources>
```

Internacionalización

res/values/strings.xml

Las apps tienen el potencial de ser globales y al estar disponibles en todo el mundo, deben adaptarse a las configuraciones del sistema operativo.

Entre esas configuraciones, una de las más comunes es el idioma.

- Cuando la aplicación es ejecutada, Android intenta usar la aplicación usando el idioma configurado.
- En cada archivo se define una clave para el texto y su valor asociado al idioma correspondiente.

Internacionalización

res/values/strings.xml

Por defecto, existe el archivo Strings.xml que contiene los textos en inglés, este archivo se utiliza cuando no se encuentran disponibles las traducciones para un idioma.

Se puede crear un nuevo archivo de traducciones haciendo clic sobre el archivo **Strings.xml** (botón derecho) → **Open Translations Editor**

```
<resources>
  <string name="app_name">EmailExample</string>
  <string name="to">to</string>
  <string name="subject">subject</string>
  <string name="message">message</string>
  <string name="send">send</string>
  <string name="button_3">Third Button</string>
  <string name="button_2">Second Button</string>
  <string name="button_1">First Button</string>
</resources>
```

```
<resources>
  <string name="app_name">EmailExample</string>
  <string name="to">para</string>
  <string name="subject">asunto</string>
  <string name="message">mensaje</string>
  <string name="send">enviar</string>
  <string name="button_3">Tercer botón</string>
  <string name="button_2">Segundo botón</string>
  <string name="button_1">Tercer botón</string>
</resources>
```

Styles

res/values/styles.xml

Un recurso de estilo define el formato y el aspecto de una interfaz de usuario.

Se puede aplicar un estilo a cualquier componente individual (por ejemplo, un botón desde dentro del archivo de diseño), a una actividad completa o una aplicación completa (desde dentro del archivo AndroidManifest.xml).

```
<resources>
  <style name="LoginTextInputStyle" parent="Widget.Design.TextInputLayout">
    <item name="colorAccent">@color/green_light</item>
    <item name="colorControlActivated">@color/green_light</item>
  </style>
</resources>
```


Themes

res/values/themes.xml

Un tema define un conjunto de recursos con nombre a los que estilos o componentes pueden hacer referencia. Los temas asignan nombres con significado (semánticos) a los recursos, por ejemplo, `colorPrimary` para el color principal a usar en toda la app.

Los temas trabajan en conjunto con los estilos

Por ejemplo, un estilo define que un botón tiene el color principal referenciando a `colorPrimary`, y luego el tema define realmente cual es el color a utilizar. Cuando el dispositivo pasa al modo nocturno, la app cambia aplicando los colores definidos en el tema, sin tener que modificar el estilo.

Themes

res/values/themes.xml

En este tema se definen las combinaciones de colores a utilizar en la app para el modo Dark. Se pueden agregar todos los temas que se desee.

```
<resources xmlns:tools="http://schemas.android.com/tools">
  <!-- Base application theme. -->
  <style name="Theme.LoginExample" 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>
</resources>
```

`/* drawables */`

Recurso de elemento de diseño

Drawable

Un recurso de elemento de diseño (drawable) es un concepto general para un gráfico que se puede dibujar en la pantalla.



En la [documentación oficial](#) aparece un listado completo de los distintos tipos de elementos de diseño.

Recurso de elemento de diseño

Drawable

Hay distintos tipos de elementos de diseño (drawable):

- **Mapas de bit** (.png, .jpg, .gif), por ejemplo, para incluir un logo en la app.
- **Lista de capas:** Gestiona una matriz de otros elementos de diseño que se dibujan en el orden de la matriz (el índice más grande se dibuja encima), por ejemplo, para dibujar un cuadrado con un cuadrado dentro tendríamos 2 capas.
- **Lista de estados:** Archivo XML que hace referencia a distintos drawables para diferentes estados, por ejemplo, los colores de un botón cuando se presiona y se suelta.
- **Elementos de diseño de transición:** Archivo XML que define un elemento de diseño capaz de fundirse entre dos recursos de elementos de diseño, por ejemplo, al cambiar de una pantalla a otra.

`/* raw */`

Raw

¿Qué son?

Son archivos arbitrarios para guardar SIN procesar.

Permite, por ejemplo, guardar una configuración estática (archivo de configuración de extensión .json) y archivos multimedia como vídeo o audio.



Caso práctico

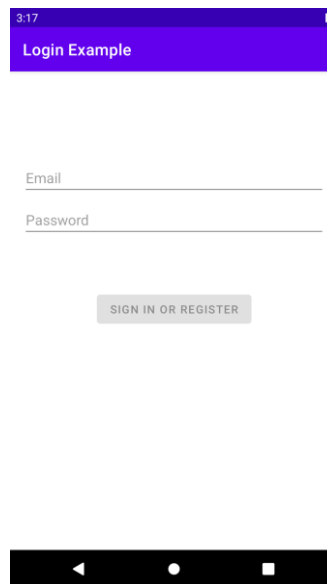


Encuentra y reproduce las diferencias

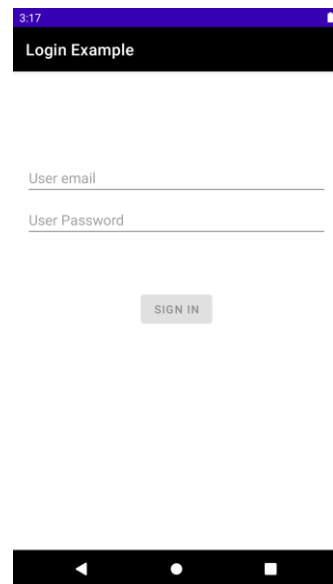
Ya contamos con nuestra primera pantalla de login y debemos aplicar algunos cambios de colores y textos.

Revisa ambas imágenes, encuentra y reproduce las diferencias.

Actual



Nueva



Caso práctico

Encuentra y reproduce las diferencias

- ¿Cuántas diferencias hay entre la pantalla actual y la nueva?
- ¿Cuáles son las diferencias?
- ¿Qué archivos de recursos hay que modificar para lograr la nueva interfaz?

¿Cuáles elementos que se
referencian en el
AndroidManifest.xml se
encuentran en el directorio
res?



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cl.dal.loginexample">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".ui.login.LoginActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```



Próxima sesión...

Identifica los elementos de configuración principales de un proyecto Android

{desafío}
latam_

*Academia de
talentos digitales*

