



Ambiente de desarrollo y sus elementos de configuración

Usando el Ambiente de Desarrollo (Parte II)

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?

Realiza empaquetamiento del proyecto Android en un archivo apk tipo depurable para su posterior distribución

¿Cuáles son los
elementos
indispensables para
desarrollar una
aplicación Android?



**/* Flujo de trabajo desde Android Studio
al dispositivo de trabajo */**

Dispositivos virtuales y físicos

Recordemos la sesión anterior

- ¿Qué tipo de dispositivos podemos utilizar para probar una app con Android Studio?
- ¿Cuántos dispositivos virtuales se pueden ejecutar simultáneamente?

`/* Los logs en Android Studio */`

Logs

¿Qué son?

Los logs son evidencias en formato de texto del funcionamiento de una aplicación y no son visibles para el usuario.

Cada registro que se crea tiene:

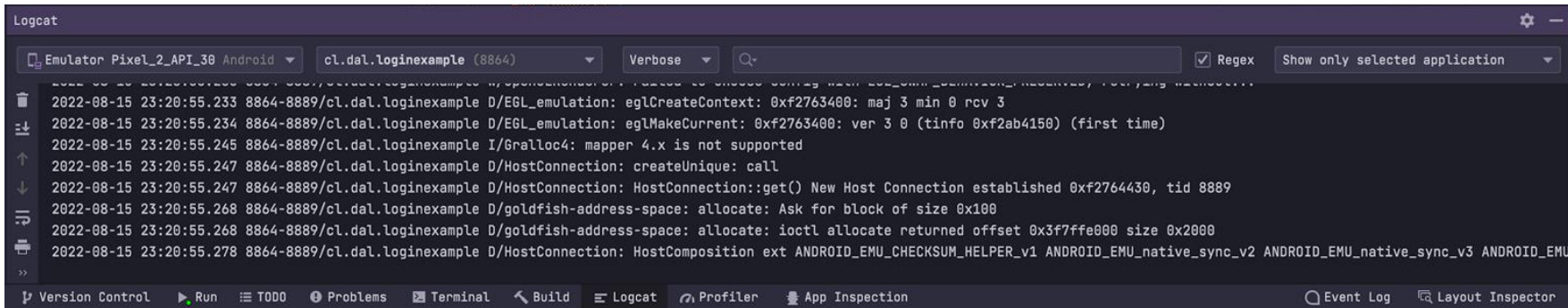
- Una marca temporal (fecha, hora, minuto, segundos, algunas veces milisegundos).
- Una categorización que generalmente describe la importancia del evento.
- Un mensaje que indica información interna del funcionamiento de la aplicación, por ejemplo, la información del usuario actual.

Logcat

Para crear un log se utiliza la clase Log, el nivel, una etiqueta (generalmente el nombre de la clase) y el mensaje a desplegar.

Log.d("NombreClase", "mensaje con información interna de la app")

Para visualizar los logs tenemos Logcat como parte de Android Studio:



Logcat

Un ejemplo de log sería:

```
2022-08-15 18:13:45.353 18485-18485/cl.dal.loginexample D/LoginActivity: onCreate: init...  
2022-08-15 18:13:45.353 18485-18485/cl.dal.loginexample D/LoginActivity: onCreate: init...
```

TimeStamp

Identificador de la app

Nivel
de
log

Etiqueta

Mensaje

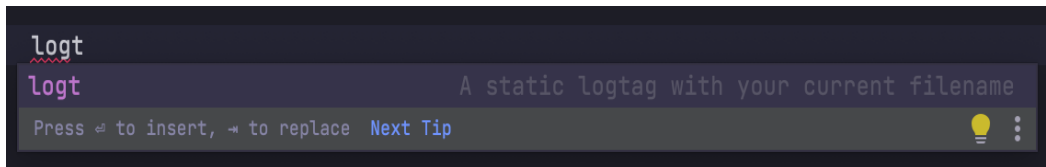
Niveles de log

Cómo se crea	Nivel de log	Descripción
Log.d(...)	Depuración	Es el nivel más utilizado durante el desarrollo. Generalmente se agregan muchos logs.
Log.i(...)	Informativo	Para información relevante que se quiere mostrar y que no corresponde al desarrollo.
Log.w(...)	Advertencia	Para avisar que hay algo que no está bien y que se debe poner atención.
Log.e(...)	Error	Solo los logs indicados como error son visibles.

Tip

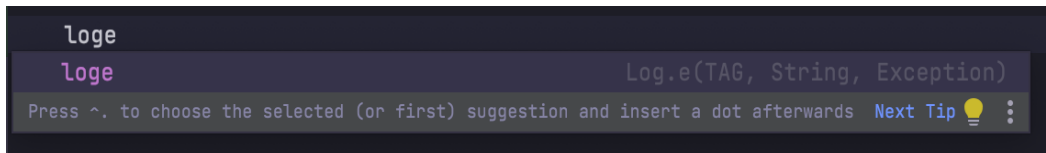
Live Templates

Android Studio incluye Live Templates que permiten escribir código de forma más rápida. Para crear el tag se escribe **logt** y el autocompletador permite crear un tag (etiqueta) a utilizar en los logs.



```
logt
logt          A static logtag with your current filename
Press ⌘ to insert, ⇧ to replace  Next Tip
```

Usando el mismo proceso, para crear un log se escribe **log<nivel de log>**, por ejemplo, **logd** o **loge**.



```
loge
loge          Log.e(TAG, String, Exception)
Press ^ to choose the selected (or first) suggestion and insert a dot afterwards  Next Tip
```

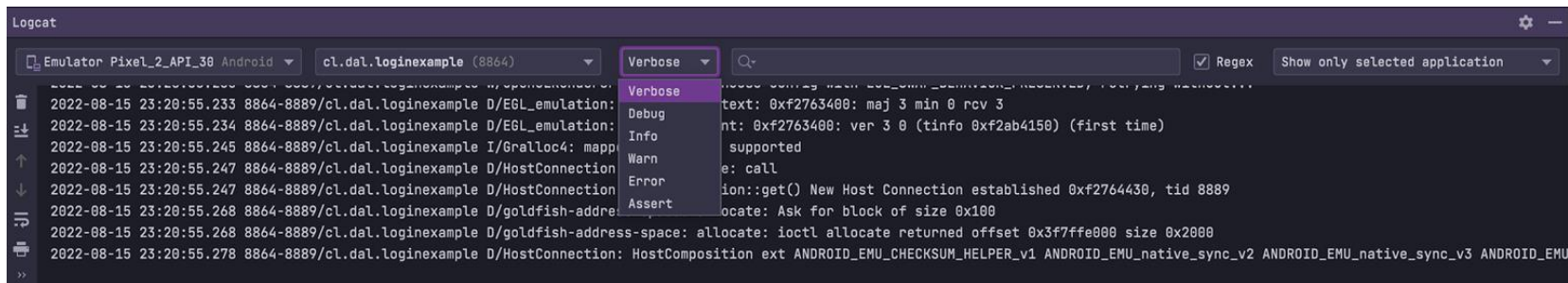
Actividad Práctica

Manos a la obra

1. Crear un proyecto desde el template de Login Activity.
2. Agregar log.
 - a. Crear un tag en la clase LoginActivity
 - b. En el método onCreate agregar un log por cada nivel de log (como mensaje utilizemos el nombre del método):
 - i. depuración
 - ii. informativo
 - iii. advertencia
3. Ejecutar el proyecto en un dispositivo.
4. Verificar en Logcat que se vean los distintos logs.
5. Filtrar por nivel de log ¿qué pasa con los logs que agregamos?

Solución

Logcat permite filtrar los distintos niveles de log



- ¿Qué ocurre al seleccionar el nivel de error?
- ¿Qué ocurre al seleccionar el nivel de depuración?

/* Depuración en Android Studio */

Formas de depurar

Usando el depurador

- Permite ver el estado de la app en tiempo de ejecución.
- La ejecución es más lenta.
- Permite interactuar con la app, por ejemplo, cambiando el valor de una variable durante la ejecución.

Usando Logs

- Permite ver información interna de la app.
- La información que se muestra depende del log escrito.

Formas de depurar la app

Depurador incorporado en Android Studio

Punto de interrupción (Breakpoint):

Es una marca en una línea de código, que en tiempo de ejecución se detendrá en este punto y permite evaluar el estado del programa, justo antes de que esa línea se ejecute.

En Android Studio, al hacer clic entre el número de línea y el borde izquierdo del editor, se marca un círculo rojo y se destaca la línea, indicando que se ha agregado un punto de interrupción.

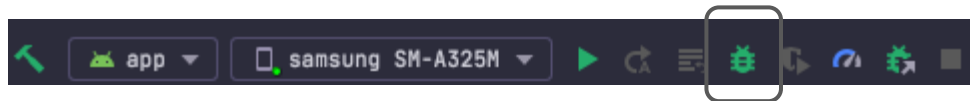
```
20 class LoginActivity : AppCompatActivity() {  
21  
22     private lateinit var loginViewModel: LoginViewModel  
23     private lateinit var binding: ActivityLoginBinding  
24  
25     override fun onCreate(savedInstanceState: Bundle?) {  
26         super.onCreate(savedInstanceState)  
27  
28         binding = ActivityLoginBinding.inflate(layoutInflater)  
29         setContentView(binding.root)
```

```
20 class LoginActivity : AppCompatActivity() {  
21  
22     private lateinit var loginViewModel: LoginViewModel  
23     private lateinit var binding: ActivityLoginBinding  
24  
25     override fun onCreate(savedInstanceState: Bundle?) {  
26         super.onCreate(savedInstanceState)  
27  
28         binding = ActivityLoginBinding.inflate(layoutInflater)  
29         setContentView(binding.root)
```

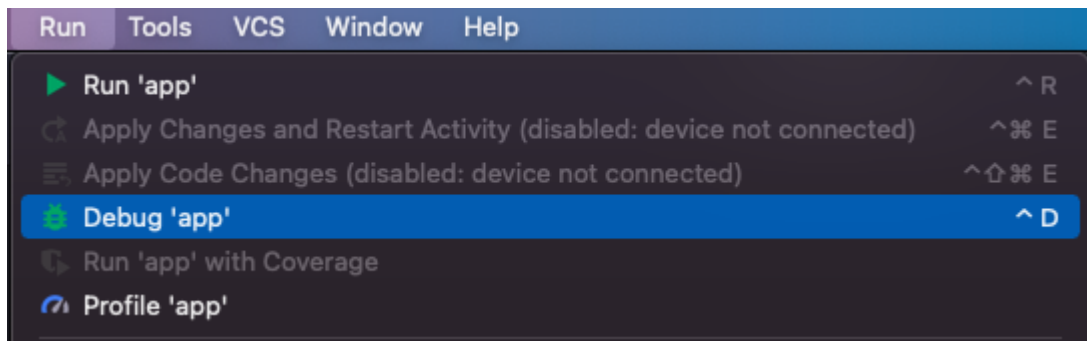
Formas de depurar la app

Depurador incorporado en Android Studio

Para poder depurar la app, se ejecuta en modo Debug, y se puede ejecutar desde la barra principal de la app:



O desde el menú Run → Debug 'app' (donde app es el nombre del módulo)



Formas de depurar la app

Logcat

Se puede ir agregando logs para indicar el estado de una variable o indicar con una marca la ejecución de un trozo de código específico (por ejemplo, al agregar un log al inicio de un método para indicar que se está ejecutando).

Es una **forma rápida de ver información del sistema**, pero no permite interactuar con el estado de la app (no se pueden modificar los valores en tiempo de ejecución).

Es importante usar los niveles de log para categorizarlos y permitir aplicar filtros para destacar la información relevante.

Actividad



Formas de depurar la app

- Agregar un log en la clase LoginActivity en el método onCreate(), y que la variable loading imprima su valor.
- Agregar un punto de interrupción en el log recién agregado.
- Ejecutar la app en modo debug.
- Revisar el estado de la variable loading y comparar con el log.

Tip: Android Studio permite buscar por nombre de clase usando un atajo de teclado: control + o (🏠)



/* apk de depuración */

APK de depuración

¿Qué es?

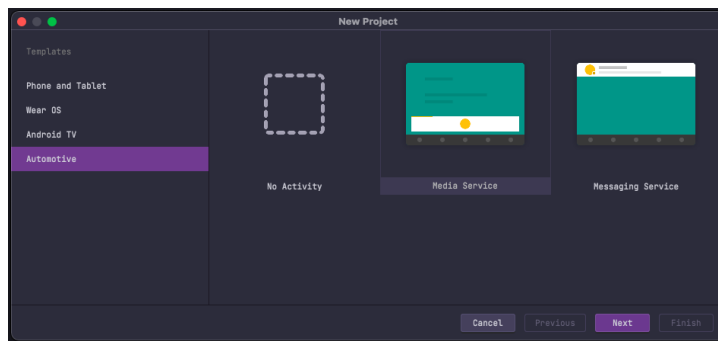
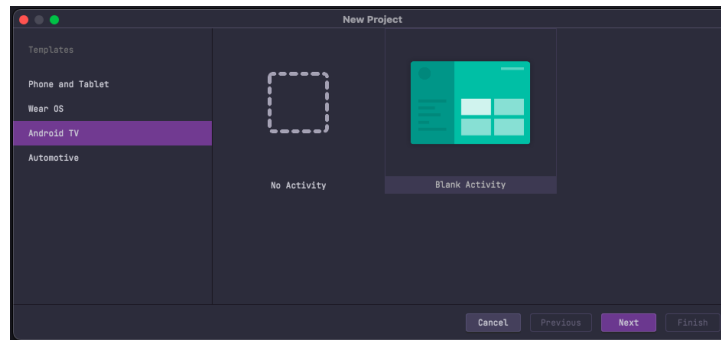
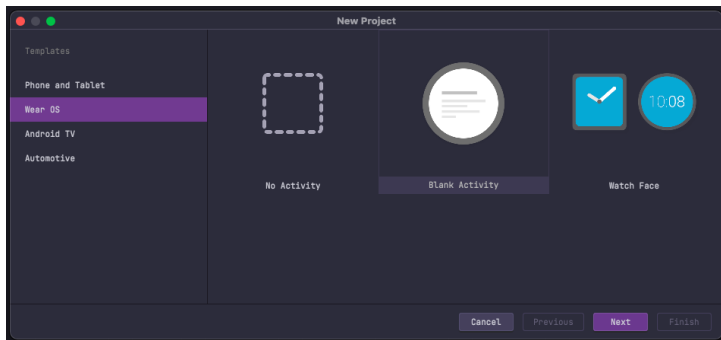
El paquete de Android (APK) es el formato instalable y ejecutable de Android para apps.

- Actualmente, Google ha empujado **AAB** (Android App Bundle) que es un formato de publicación para Android compatible con Google Play.
- En el caso de depuración, Android Studio genera el paquete y lo instala en el dispositivo.
- El paquete generado queda dentro del directorio `app/build/outputs/`

**/* Templates de proyectos
incluidos en Android Studio */**

Plantillas para relojes, TV y Automóvil

Ejemplos



Plantillas para relojes, TV y Automóvil

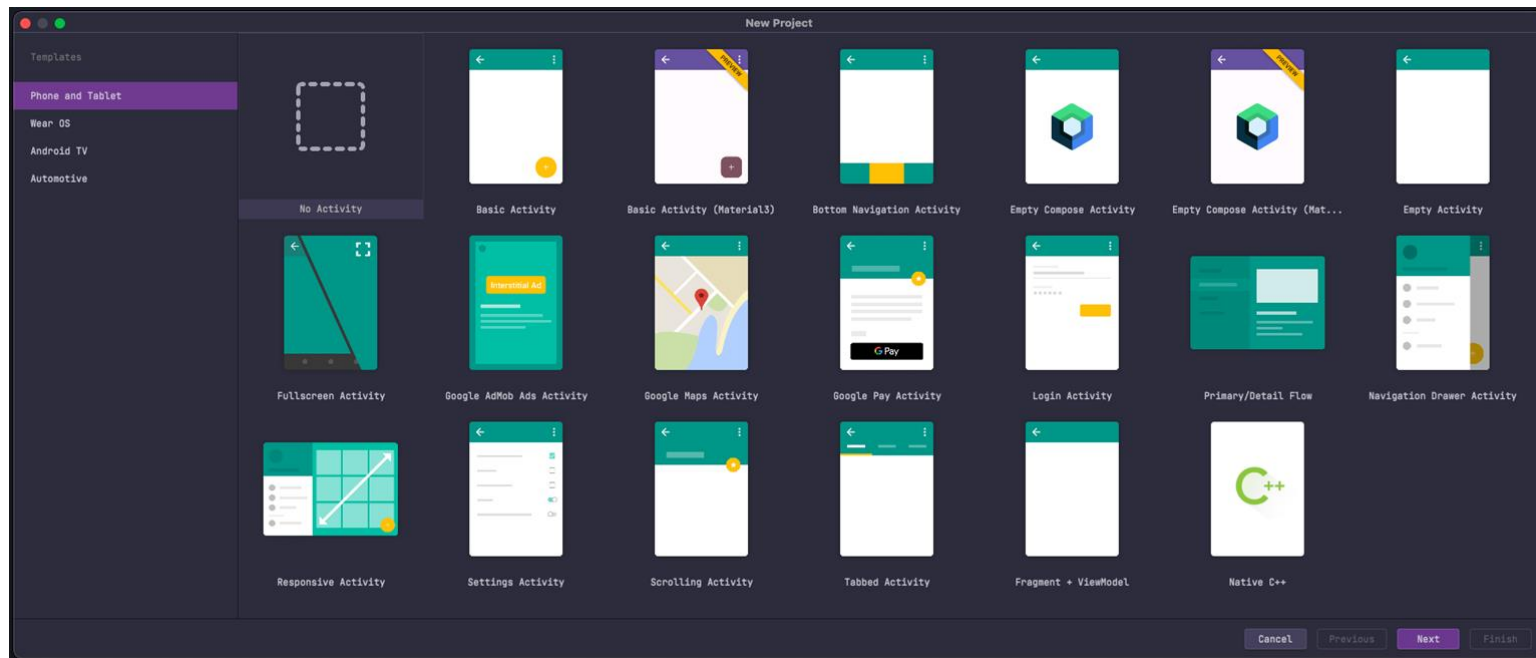
Ejemplos

Podemos encontrar Android en:

- Televisores inteligentes
- SmartWatch
- Automóviles, lo utilizan marcas como Audi, Kia y Lamborghini.

Si tuvieras que hacer una aplicación para mostrar una notificación cuando se alcance el objetivo de número de pasos diarios, ¿A qué tipo de dispositivo estaría enfocado? ¿Qué plantilla emplearías?

Plantillas para teléfonos y tabletas



Plantillas para teléfonos y tabletas

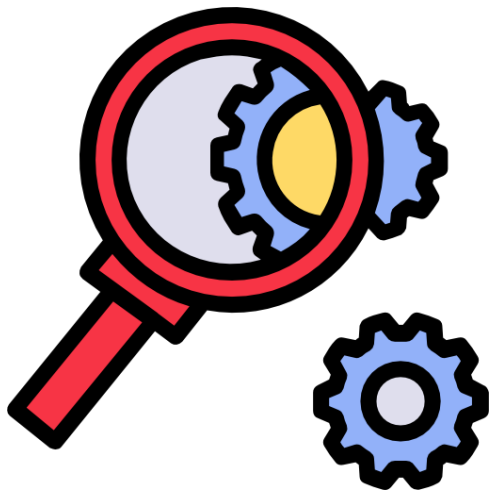
Algunas plantillas incluyen parte de la arquitectura recomendada por Google para desarrollar aplicaciones, o sea, definen e incluyen componentes comunes para solucionar un problema dado.

Por ejemplo, la plantilla “Google Maps Activity” incluye la pantalla con el mapa y los archivos de configuración donde se colocan las claves necesarias para que funcione el servicio.

Ejercicio guiado



Plantillas para teléfonos y tabletas



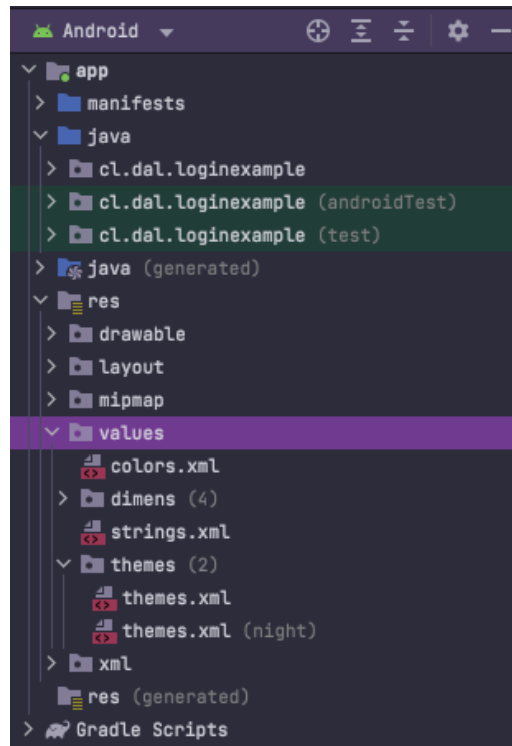
Descubramos juntos como luce una pantalla de configuración estándar para aplicaciones:

1. Usar la plantilla “Settings Activity”. Este proyecto incluye una pantalla de configuración estándar utilizada por la mayoría de las aplicaciones.
1. Ejecutar la app para validar el proyecto.

/* Vistas de proyectos */

Vista Android

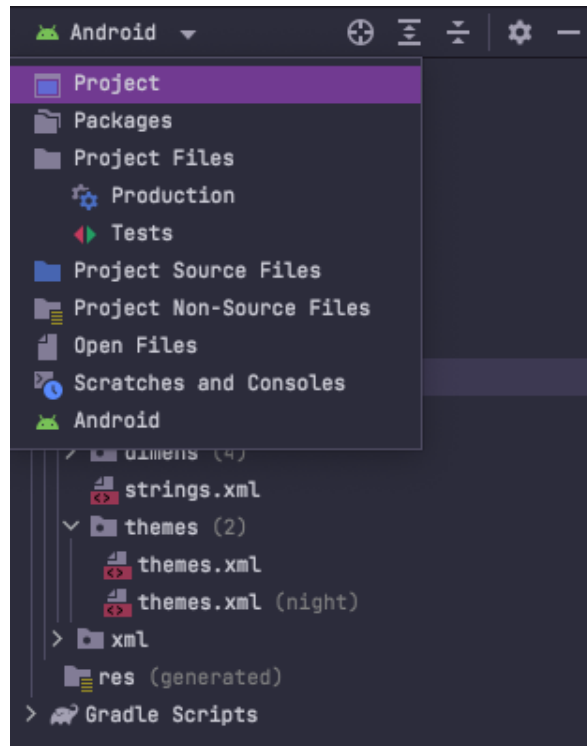
- Está pensando en el desarrollo de apps Android.
- Esconde información irrelevante (archivos de configuración, directorios ocultos).



Elegir un tipo de vista

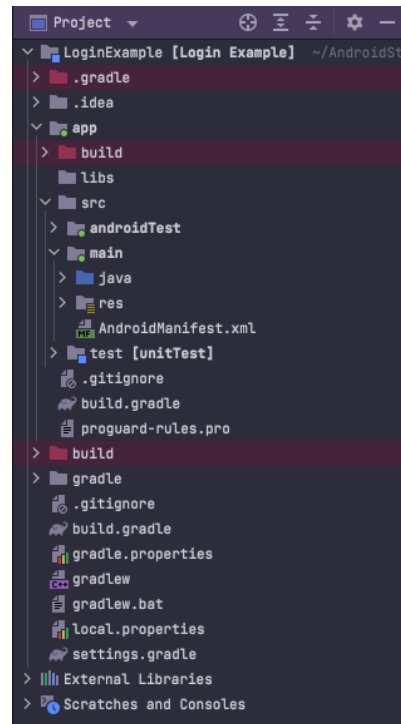
Cada vista sirve para explorar la estructura del proyecto, pero cada vista lo muestra de forma distinta.

Las más utilizadas son Project y Android.



Vista de proyecto

- Muestra la estructura de directorios del proyecto.
- Permite ver archivos y directorios ocultos.



¿En qué casos utilizamos logs
para depurar nuestra app?

¿En qué casos usamos el
depurador incluido en Android
Studio?





Próxima sesión...

Identifica los elementos de configuración principales de un proyecto Android

{desafío}
latam_

*Academia de
talentos digitales*

