

Antes de la programación en Android

Conceptos básicos



Bases

| ¿Qué hay dentro de Android?

La Máquina Virtual de Android: ART

¿Qué es una máquina virtual?

¿En qué me influye?



Java y Kotlin

¿Java?

¡¡Pero yo he estudiado Kotlin!!

Let's work together!



Reglas de Android

Trabajamos dentro de Android —→ Obligación de respetar esas reglas.

- Sistema de permisos.
- Uso de Activities y Fragments con ciclos de vida.
- Limitaciones de la API.





Android Manifest

¿Qué es?

Configuración del proyecto a nivel **externo**

Es un fichero de configuración xml cuya misión es “explicar” a Android, Google Play y otras aplicaciones cómo interactuar con nuestra App.

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="DragonBall"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.DragonBall"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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



¿Para qué sirve?

Indica a Android como deberá interactuar con tu aplicación.

Ejemplos:

- A qué nombre debe responder
- Que activities componen la app
- Qué estímulos puede recibir mi app
- Qué pantalla debe abrir en primer lugar
- Qué permisos necesita usar mi app
- Etc.

Dependiendo de la vista seleccionada lo encontraremos en un lugar u otro.



Gradle

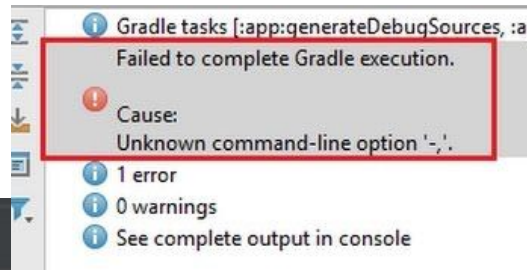
¿Qué es?

Configurador del proyecto a nivel interno

- Es un sistema de compilación de código - Librerías, plugins, etc. -. Utiliza código en Groovy o Kotlin scripting

Es decir, te permite incorporar de manera “sencilla” librerías a tus proyectos de forma super sencilla

- Un dolor de cabeza menos malo que cualquier otra solución
- Lento



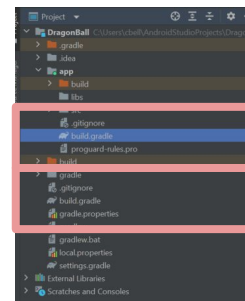
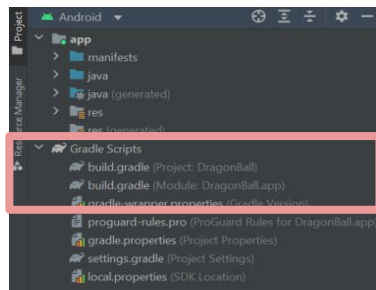
¿Para qué sirve?

Importante: ¡No existe un solo gradle!

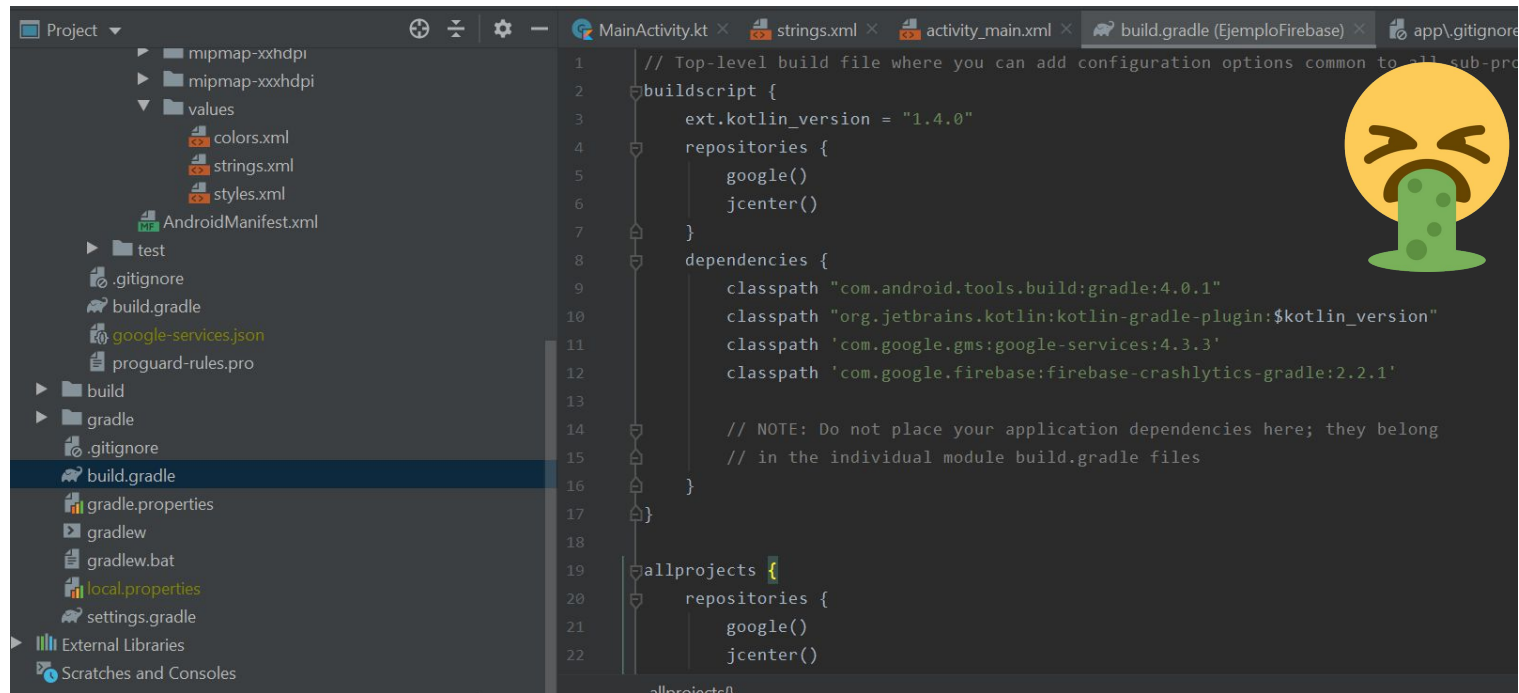
Hay un gradle por módulo + un gradle general.

Cada gradle configura su módulo de forma independiente.

Dependiendo de la vista seleccionada lo encontraremos en un lugar u otro.



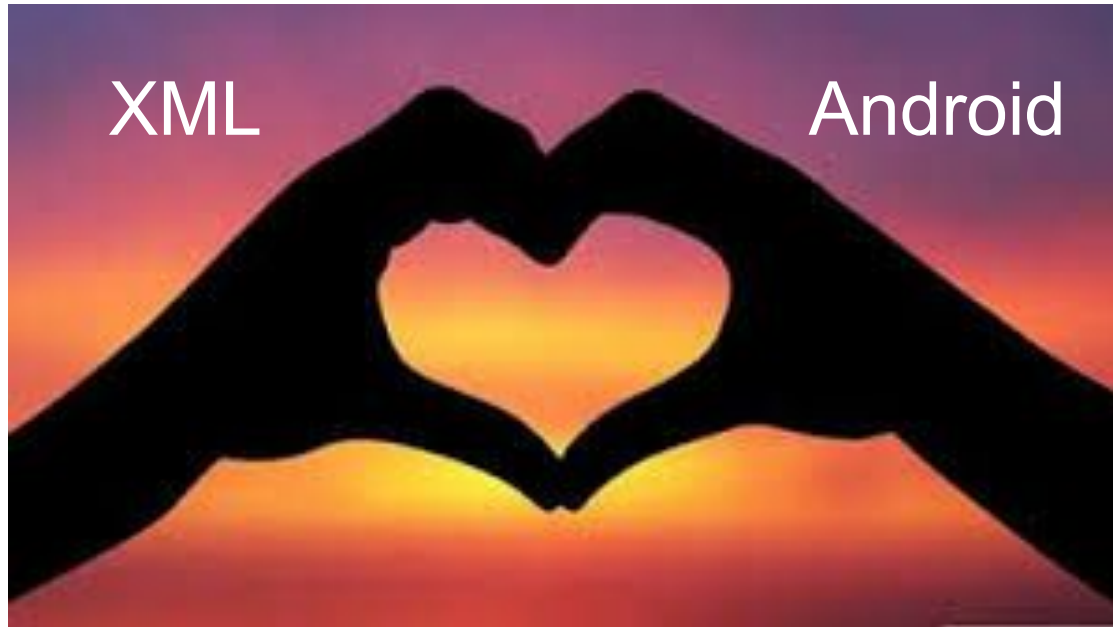
¿Cómo se ve?



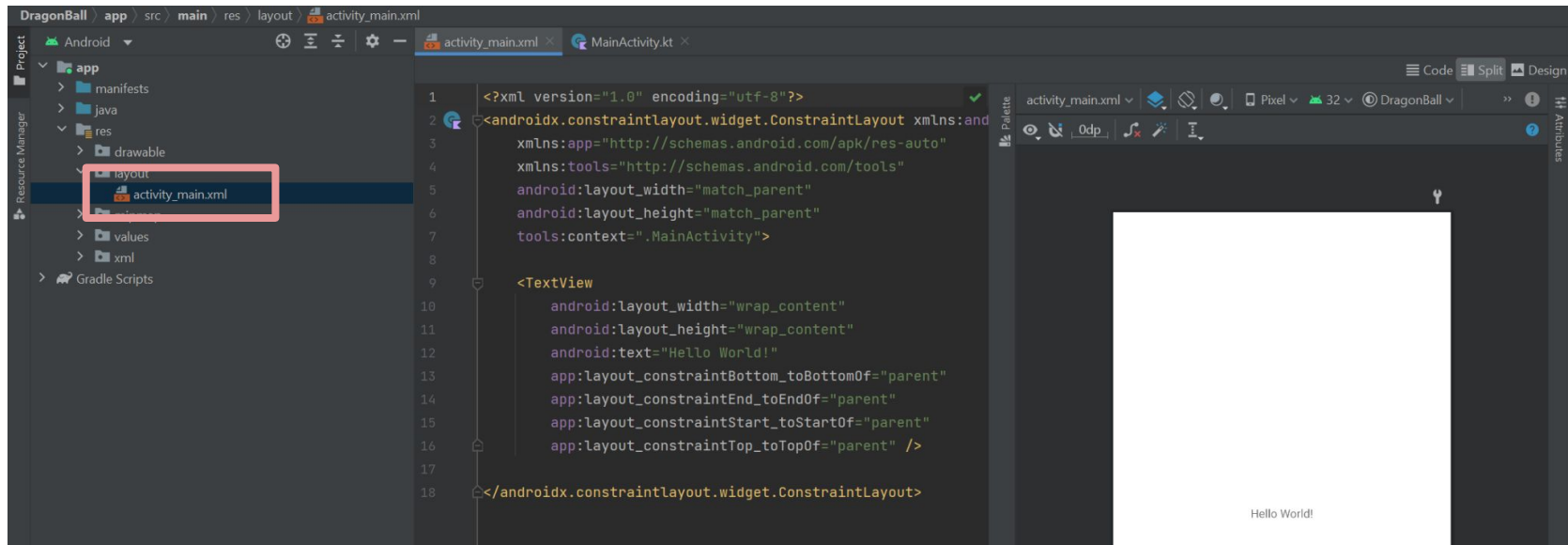


Interfaz Gráfica

¿Cómo se crean las Interfaces Gráficas?



¿Dónde creamos las Interfaces Gráficas?





Strings

¿Cómo se gestionan los String en Android?

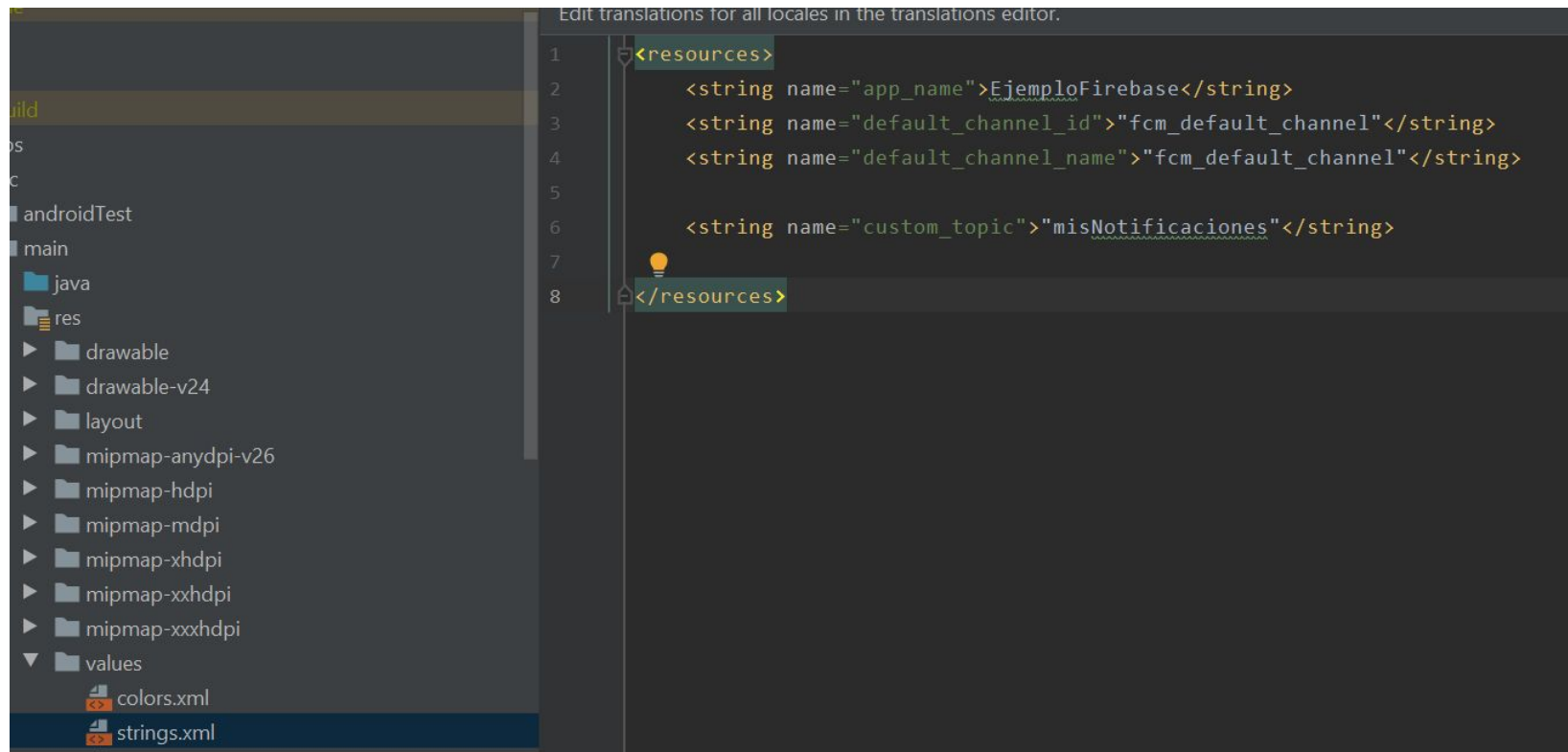
Opción mala: Los strings se pueden generar por código como hemos visto hasta ahora en otros lenguajes.

Opción buena: Ficheros string.xml.

¿Por qué?

Facilita la traducción de apps al centralizar los String en un solo fichero **por idioma**.

Los Strings en Android: Archivo XML



¿Cómo funcionan?

Si nuestra aplicación dispone de un fichero `string.xml` específico para el idioma con el que se ha configurado el móvil Android donde se ejecuta nuestra app, entonces Android automáticamente lo usará.

Con esto obtenemos traducciones fáciles sin necesidad de código extra.

Desde la aplicación se usará el siguiente código para acceder al string.

```
getString(R.string.custom_topic)
```



Ejemplo

<https://github.com/KeepCodingMobile16/Fundamentos-Android/commit/e784a3278c16b4b3033e32407f4ada14c314fba>



Colors, Dimensions, Styles...



¿Cómo se gestionan?

La filosofía es similar a los Strings.

El objetivo es crear archivos xml que definan ciertos aspectos configurables de la app.

Por ejemplo:

- Colores
- Dimensiones
- Estilos

¿Colores?

Definir los colores en un archivo xml nos permite usarlos desde otros archivos xml (ejemplo, interfaces gráficas)

Mediante técnicas más avanzadas (flavours) nos permitirá modificar toda la paleta de colores de un solo golpe.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <color name="purple_200">#FFBB86FC</color>
4      <color name="purple_500">#FF6200EE</color>
5      <color name="purple_700">#FF3700B3</color>
6      <color name="teal_200">#FF03DAC5</color>
7      <color name="teal_700">#FF018786</color>
8      <color name="black">#FF000000</color>
9      <color name="white">#FFFFFFFF</color>
10 </resources>
```

¿Dimensiones?

Definir las dimensiones en un archivo xml nos permite usarlos desde otros archivos xml (ejemplo, interfaces gráficas)

Mediante técnicas más avanzadas (flavours) nos permitirá modificar los tamaños de toda la app de un solo golpe.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <dimen name="default_margin">16dp</dimen>
</resources>
```


¿Estilos?

Un estilo consiste en una gama de propiedades aplicables a los elementos de la interfaz gráfica por defecto.

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

¿Jerarquía de Estilos?

Los estilos tienen diversos niveles de aplicabilidad.

Son aplicables a nivel de:

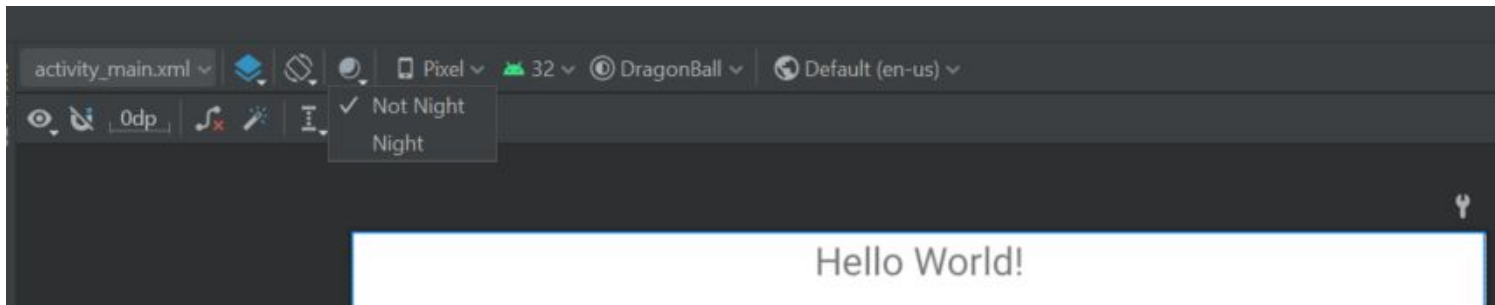
- App, siendo aplicados en el “Android manifest”.
- Activity, siendo aplicados en el “Android manifest”.
- Vista, siendo aplicados en la creación de la interfaz gráfica.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.keepcoding.dragonball">

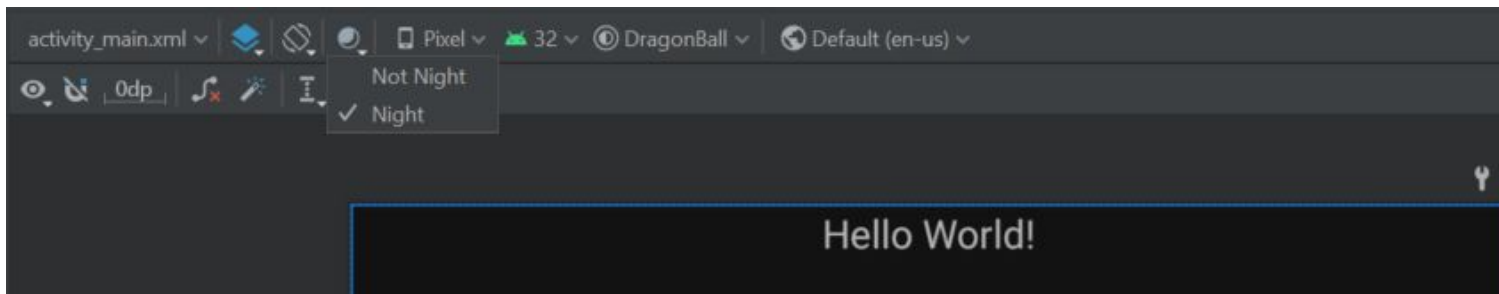
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.DragonBall"
        tools:targetApi="31">
```

Tema oscuro vs Tema claro

Claro:



Oscuro:



Ejercicio Destacado



Crea una nueva aplicación con el paquete:

com.keepcoding.dragonball

Escoge una paleta de colores:

<https://color.adobe.com/es/create/color-wheel>



Ejemplo



<https://github.com/KeepCodingMobile16/Fundamentos-Android/commit/b66a66e432e4f481f3bdb87e9f3ec59aebf83d05>



Logs

¿Cómo imprimir un Log?

```
Log.v(TAG, "Aquí logueo cosas poco importantes o rutinarias")
```

```
Log.d(TAG, "Aquí logueo cosas referentes al debug de mi aplicación")
```

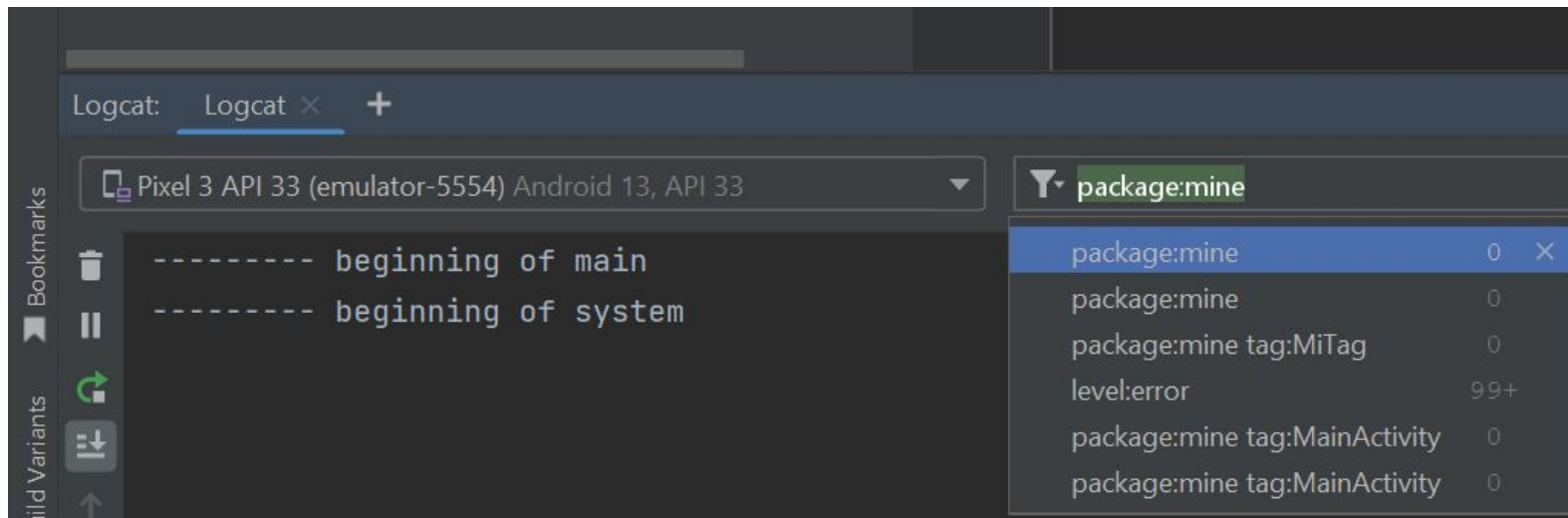
```
Log.i(TAG, "Aquí logueo información importante")
```

```
Log.w(TAG, "Aquí logueo cosas que no deberían haber pasado o que han fallado pero hemos sobreponernos al fallo. Debe usarse poco")
```

```
Log.e(TAG, "Aquí logueo errores")
```

```
Log.a(TAG, "Aquí logueo el resultados de los asserts cuando estoy haciendo unit testing")
```

¿Cómo visualizar un Log?





Control de Versiones: Git

Control de Versiones

Android Studio implementa un cliente de gestión de versiones bastante completo.

- Acepta Git, Subversion, Mercurial...
- Además tiene soporte para GitHub.

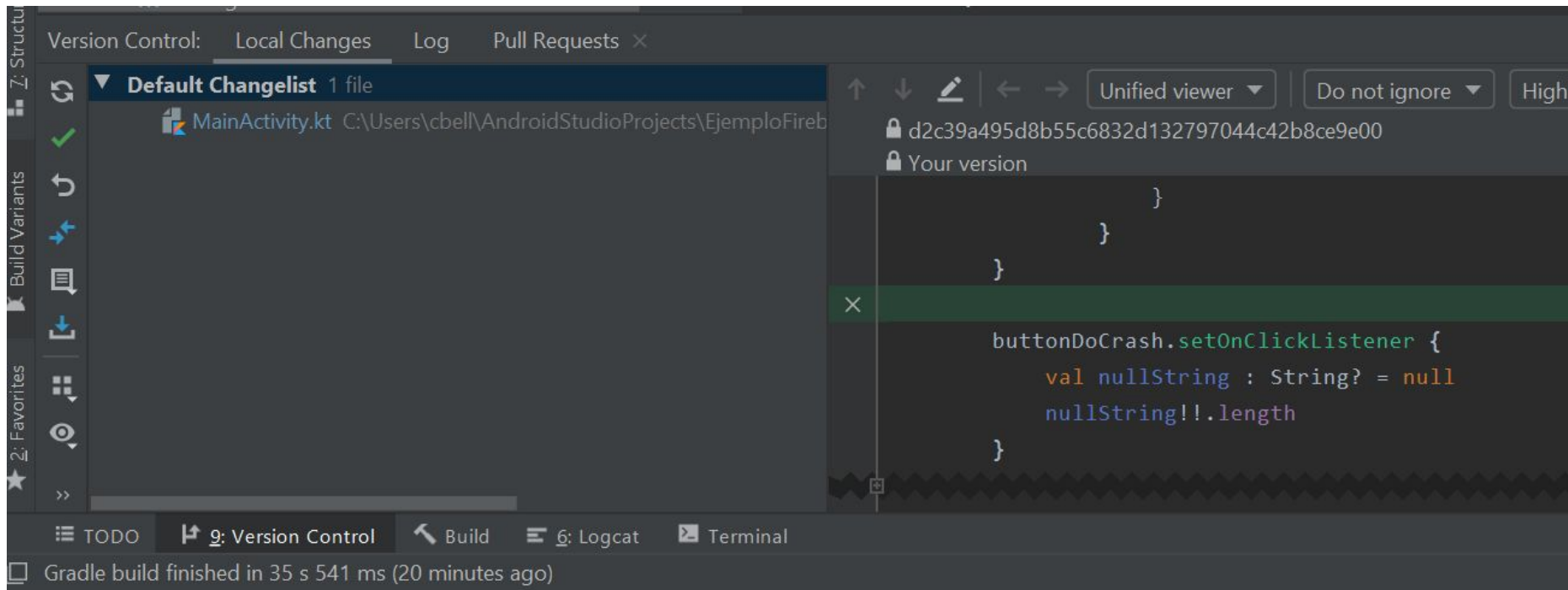


¡Atención! Vamos a utilizar GitHub constantemente.

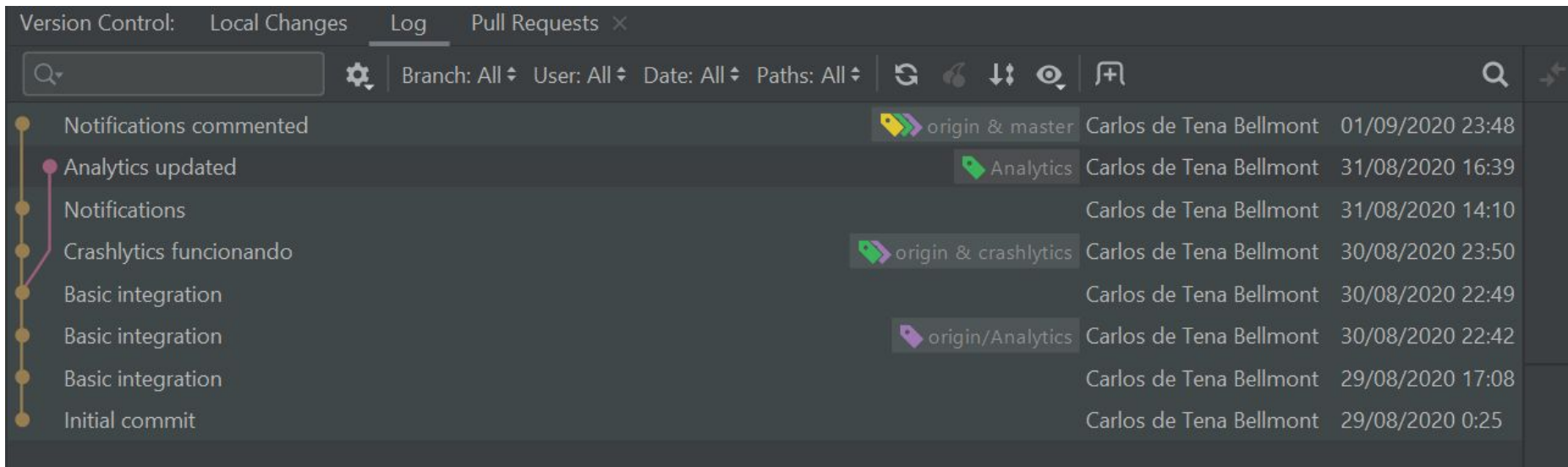


Si no sabes utilizarlo, es indispensable que aprendas.

Git en Android Studio: Local Changes



Git en Android Studio: Logs



The screenshot shows the 'Log' tab in the Version Control panel of Android Studio. The interface includes a search bar, filter buttons for Branch, User, Date, and Paths, and a toolbar with icons for refresh, merge, compare, and other Git operations. The log history is displayed as a list of commits, each with a commit message, a branch/tag indicator, the author's name, and the commit time. The commit messages include 'Notifications commented', 'Analytics updated', 'Notifications', 'Crashlytics funcionando', 'Basic integration', and 'Initial commit'. The branches shown are 'origin & master', 'Analytics', 'origin & crashlytics', and 'origin/Analytics'.

Commit Message	Branch/Tag	Author	Date
Notifications commented	origin & master	Carlos de Tena Bellmont	01/09/2020 23:48
Analytics updated	Analytics	Carlos de Tena Bellmont	31/08/2020 16:39
Notifications		Carlos de Tena Bellmont	31/08/2020 14:10
Crashlytics funcionando	origin & crashlytics	Carlos de Tena Bellmont	30/08/2020 23:50
Basic integration		Carlos de Tena Bellmont	30/08/2020 22:49
Basic integration	origin/Analytics	Carlos de Tena Bellmont	30/08/2020 22:42
Basic integration		Carlos de Tena Bellmont	29/08/2020 17:08
Initial commit		Carlos de Tena Bellmont	29/08/2020 0:25



Ejemplo



<https://github.com/KeepCodingMobile16/Fundamentos-Android/commit/cbff870a4f18c52e969d71debb843dfa42fdf342>

Ejercicio Destacado

Inicializa tu proyecto en Git

Haz tu primer commit

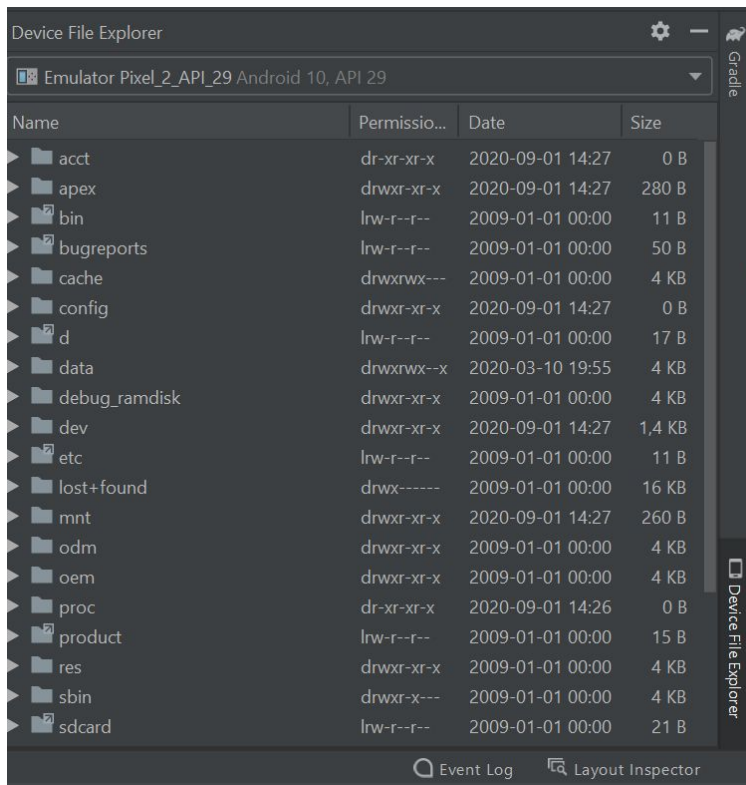
Envía tu proyecto a tu GitHub





Analizando tu dispositivo: Explorador de archivos

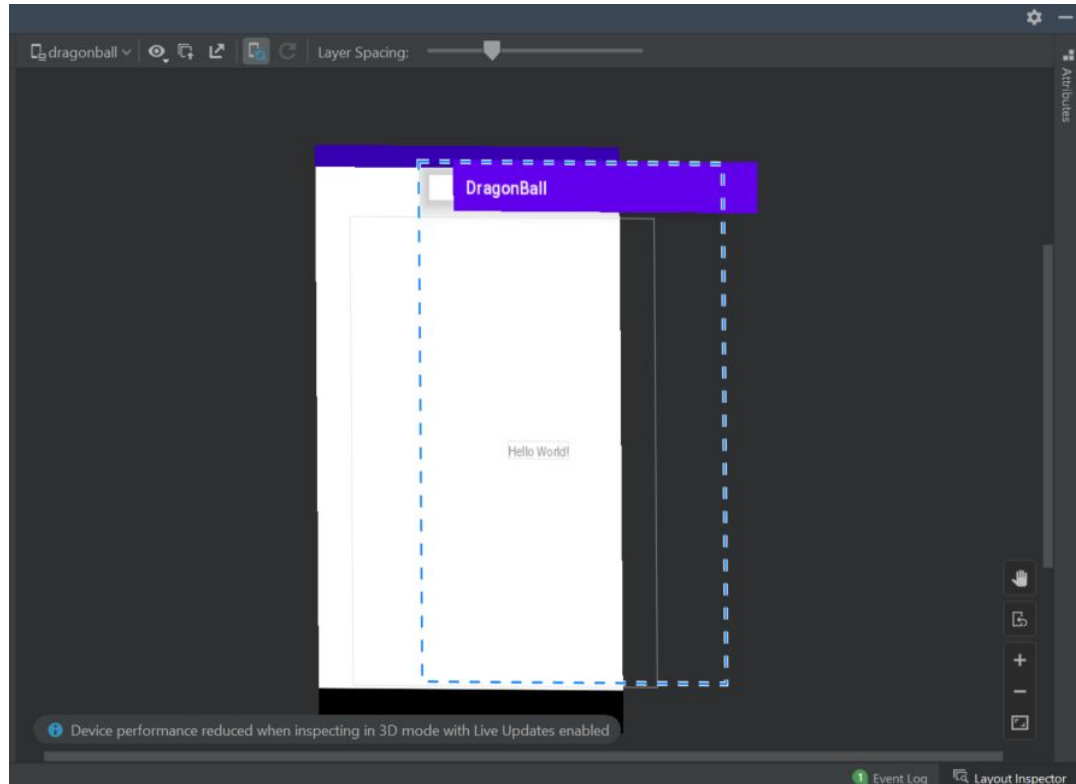
File Explorer en Android Studio





Analizando tu dispositivo: Layout Inspector

Layout Inspector





¡Ejercicio para Casa!

Investiga cómo funciona el sistema de archivos en Android.

¿Dónde se instalan las App?

¿Dónde se guardan los documentos?

¿A qué carpetas tiene acceso el usuario?

¿Qué sistema de permisos existe?



Debugger



¿Qué es?

Herramienta que nos permite observar en tiempo real el estado de mi aplicación en un momento dado.

Es decir, nos permite ejecutar nuestra app y forzar su parada en un momento dado (breakpoint).



Conceptos básicos



Iniciar el Debugger

Agregar Breakpoints

Observar la secuencia de ejecución



Observando el valor de las variables



¡Y cambiando los valores en vivo!



Futuro... ¡cercano!

JetPack Compose



Jetpack
Compose

¡¡Lo veis en SuperPoderes!!



Afianza los conceptos



Afianza los conceptos

<https://forms.office.com/r/jU1ZZDcicz>



Fin