



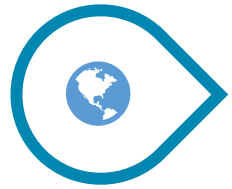
# GEOLOCALIZACIÓN CON ANDROID

DESARROLLO DE APLICACIONES MÓVILES AVANZADAS

**LUIS HUMBERTO RIVAS RODRÍGUEZ**

INGENIERO EN SISTEMAS INFORMÁTICOS  
Y MÁSTER EN DIRECCIÓN ESTRATÉGICA DE EMPRESAS.

## Agenda



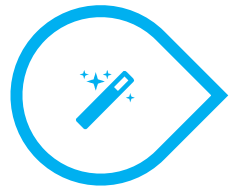
# 1 Generalidades

Conceptos

## Location

Funcionalidades

2



# 3 Desafío de Clases

Ejemplo Práctico

## Cierre

Valoraciones Finales

4



# RESULTADOS DE APRENDIZAJE

Competencias a desarrollar al finalizar la sesión.



## **INTERPRETA**

Los componentes para usar la geolocalización dentro de una interfaz gráfica en Android.



## **COMPRENDE**

La relación que tienen las librerías y su importancia en el desarrollo de aplicaciones basadas en Android.



## **CONSTRUYE**

Aplicaciones Android haciendo uso de los componentes avanzados como son las librerías en Android.

# CONTEXTO DE USO

Requerimientos principales



- Con las API de ubicación de los Servicios de Google Play, tu app puede solicitar la ubicación más reciente del dispositivo del usuario. En la mayoría de los casos, te interesará la ubicación actual del usuario, que suele equivaler a la ubicación más reciente del dispositivo.
- LocationServices es uno de los elementos de la API que ofrece Google Services para la gestión de ubicaciones.

# LocationManager

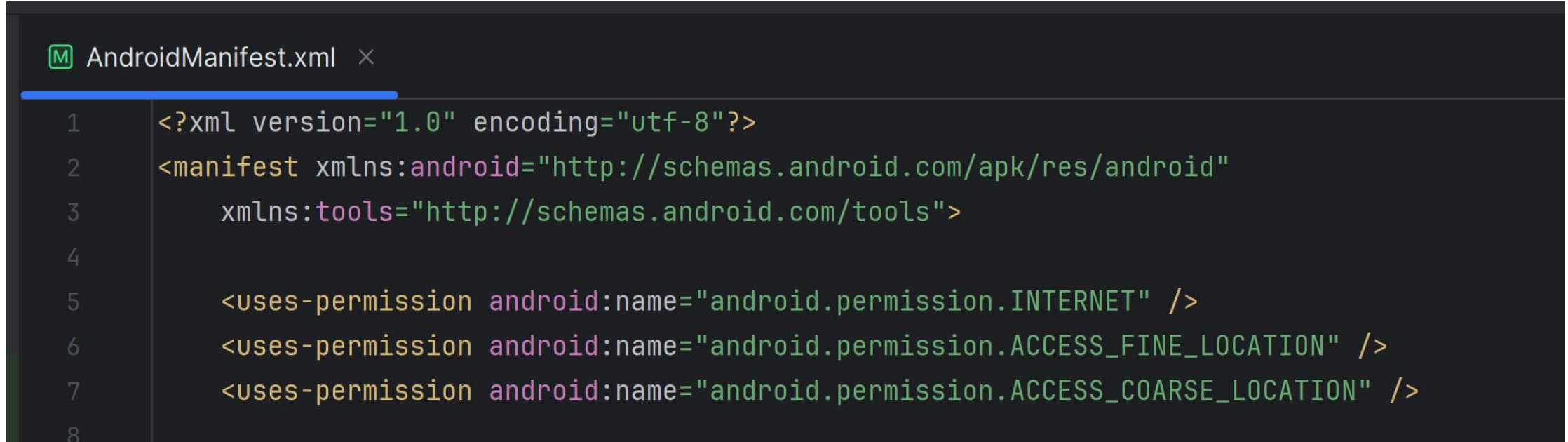
CLASE



- Esta clase proporciona acceso a los servicios de ubicación del sistema.
- Estos servicios permiten que las aplicaciones obtengan actualizaciones periódicas de la ubicación geográfica del dispositivo, o que sean notificados cuando el dispositivo se acerque a una determinada ubicación geográfica.

# PERMISOS

## CONTEXTUALIZACIÓN

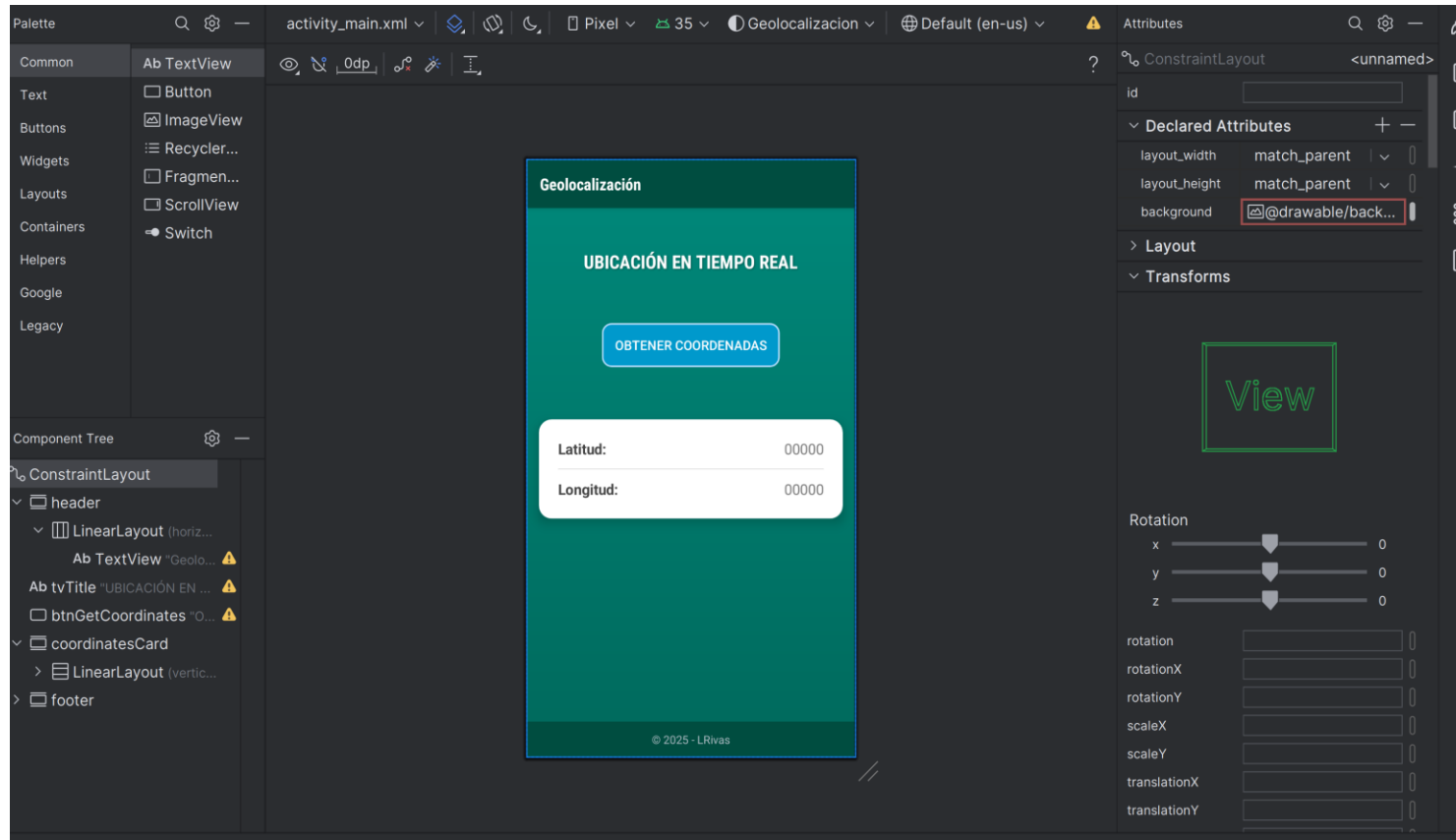


```
M AndroidManifest.xml ×
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools">
4
5   <uses-permission android:name="android.permission.INTERNET" />
6   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
7   <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
8
```

- Se debe asignar los permisos relacionados a la geolocalización, es importante recalcar que si se usa ACCESS\_FINE\_LOCATION la documentación recomienda hacer uso de ACCESS\_COARSE\_LOCATION.

# DISEÑO DE LA INTERFAZ

## USO DEL CONTROL



- La idea principal de la aplicación es que a través de un botón obtener la ubicación actual del dispositivo.



# FUNCIONES CLAVES

## REQUESTLOCATIONUPDATES

```
gestorUbicacion.requestLocationUpdates(  
    LocationManager.GPS_PROVIDER,  
    minTimeMs: 1000,  
    minDistanceM: 0.5f,  
    localizacion  
)  
  
if (gestorUbicacion.isProviderEnabled(LocationManager.NETWORK_PROVIDER)) {  
    gestorUbicacion.requestLocationUpdates(  
        LocationManager.NETWORK_PROVIDER,  
        minTimeMs: 1000,  
        minDistanceM: 0.5f,  
        localizacion  
    )  
}
```

- El método **requestLocationUpdates** nos permitirá obtener la ubicación de nuestro dispositivo, utilizando diferentes providers y en intervalos de tiempo.
- Se debe usar un **LocationListener** para gestionar los métodos en los que cambió de lugar o se movió el dispositivo móvil.



# MÉTODOS CLAVES

## onLocationChanged

```
class Localizacion(  
    private val textoLatitud: TextView,  
    private val textoLongitud: TextView,  
    private val mainActivity: MainActivity  
) : LocationListener {  
  
    private var lastLocation: Location? = null  
    private val geocoder: Geocoder by lazy { Geocoder(mainActivity, Locale.getDefault()) }  
  
    fun setLastLocation(location: Location) {  
        lastLocation = location  
        updateAddress(location)  
    }  
  
    override fun onLocationChanged(loc: Location) {  
        textoLatitud.text = String.format("%.6f", loc.latitude)  
        textoLongitud.text = String.format("%.6f", loc.longitude)  
        updateAddress(loc)  
        lastLocation = loc  
    }  
}
```

- Para tener acceso en tiempo real o en intervalos de tiempo, se debe hacer uso de un **onLocationChanged** para capturar a través de un **LocationListener** las coordenadas del dispositivo móvil.

# MÉTODOS CLAVES

onProviderDisabled, onProviderEnable, onStatusChanged

```
override fun onProviderDisabled(provider: String) {  
    Toast.makeText(mainActivity, text: "GPS Desactivado", Toast.LENGTH_SHORT).show()  
}  
  
override fun onProviderEnabled(provider: String) {  
    Toast.makeText(mainActivity, text: "GPS Activado", Toast.LENGTH_SHORT).show()  
}  
  
override fun onStatusChanged(provider: String, status: Int, extras: Bundle) {  
    when (status) {  
        LocationProvider.AVAILABLE -> {}  
        LocationProvider.OUT_OF_SERVICE -> {}  
        LocationProvider.TEMPORARILY_UNAVAILABLE -> {}  
    }  
}
```

- También podemos manejar los momentos en que el usuario habilita o deshabilita los servicios de localización del dispositivo.
- El método onStatusChanged lo utiliza el LocationListener para determinar el estado del servicio.



- Tomando como referencia los contenidos visto en la sesión, elabore una aplicación que permita capturar las coordenadas del dispositivo móvil y se muestren por pantalla.



# VALORACIONES FINALES

Comentarios sobre el tema.

1 **LocationListener**  
Es la clase encargada de gestionar el estatus del sistema de geolocalización en una aplicación Android.

2 **ACCESS\_FINE\_LOCATION y ACCESS\_COARSE\_LOCATION**  
Son los permisos indispensables para el manejo de geolocalización en aplicaciones Android.

3 **Creativo**  
Saber distribuir y seleccionar los diferentes métodos y propiedades de una clase, permitirá al programador trabajar de forma rápida y eficiente



# ¿PREGUNTAS?

DESARROLLO DE APLICACIONES MÓVILES AVANZADAS

**LUIS HUMBERTO RIVAS RODRÍGUEZ**

INGENIERO EN SISTEMAS INFORMÁTICOS  
Y MÁSTER EN DIRECCIÓN ESTRATÉGICA DE EMPRESAS.