



Universidad
Tecmilenio®





Desarrollo de aplicaciones en plataforma Android

Mapeo





La API de Google Maps se integró con los Google Play Services aproximadamente a finales del año 2012. Trayendo consigo importantes mejoras, como la utilización de mapas vectoriales y mejoras en el sistema de caché, produciendo aplicaciones con mejor y mayor rendimiento, mayor velocidad y menor consumo de datos.

Como parte de esos cambios se logró cambiar la forma de integrar los mapas de los MapActivity y MapView a un nuevo tipo de fragment llamado MapFragment, obviamente con las ventajas que conllevan estos componentes.



Las nuevas versiones de API integran el manejo de menor tráfico con el servidor, el uso de fragments y los gráficos en 3D.

El API de Google Maps para Android se compone de un conjunto básico de clases que se combinan para proporcionar capacidades de mapeo de aplicaciones de Android.

Los elementos clave de un mapa, entre los mas importantes son:

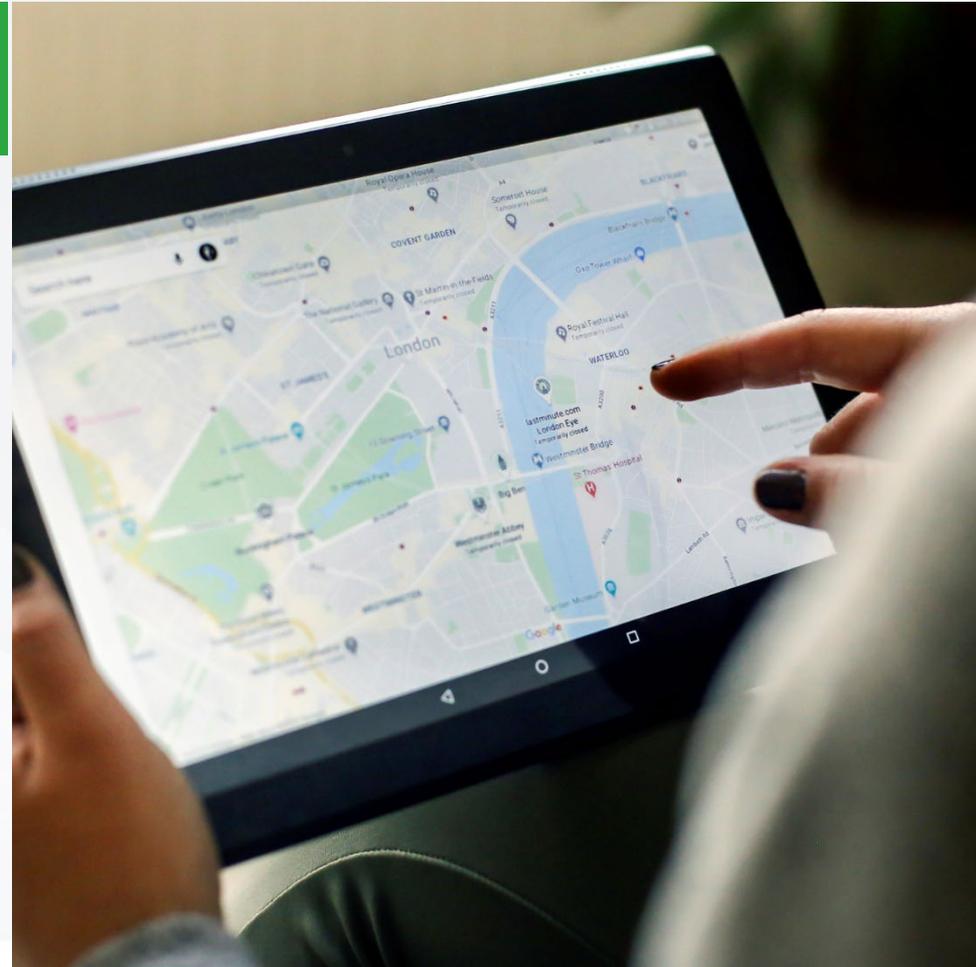
GoogleMap - La clase principal de la API de Google Maps para Android.

Es la responsable de la descarga y visualización de mosaicos de mapas y para visualizar y responder a asignar controles. El objeto GoogleMap no ha sido creada directamente por la aplicación sino que se crea cuando se crean instancias MapView o MapFragment.



MapView - Una subclase de la clase View, esta clase proporciona la lona vista en la que el mapa se dibuja por el objeto GoogleMap, permitiendo que un mapa para ser colocado en el diseño de la interfaz de usuario de una actividad.

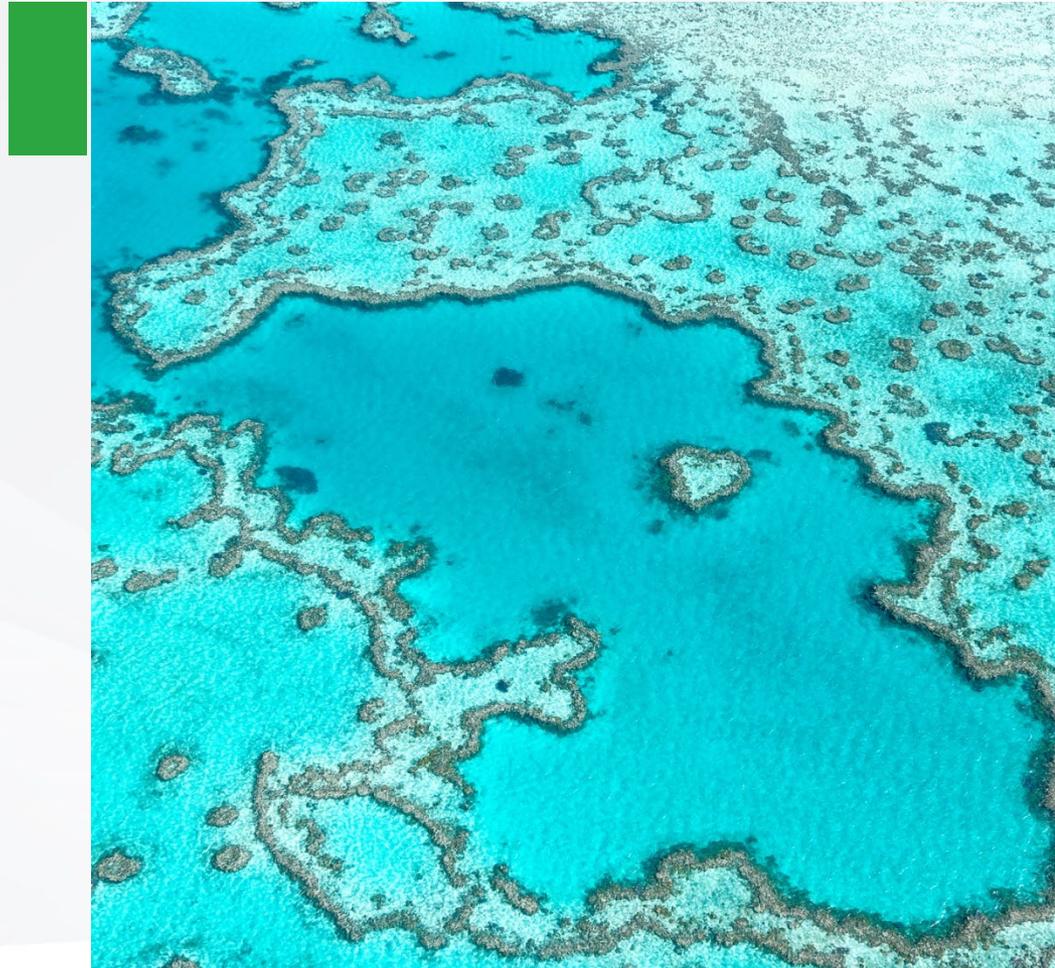
SupportFragmentManager - Una subclase de la clase Fragmento, esta clase permite que un mapa para ser colocado dentro de un fragmento en una disposición Android.



Marcador – Esta clase permite las ubicaciones que hay que marcar en un mapa.

- Los marcadores se añaden a un mapa mediante la obtención de una referencia al objeto GoogleMap asociado con un mapa y luego hacer una llamada al método de esa instancia de objeto addMarker ().

Formas - El dibujo de líneas y formas en un mapa se logra mediante el uso de las clases de polilínea, polígono y Circle.

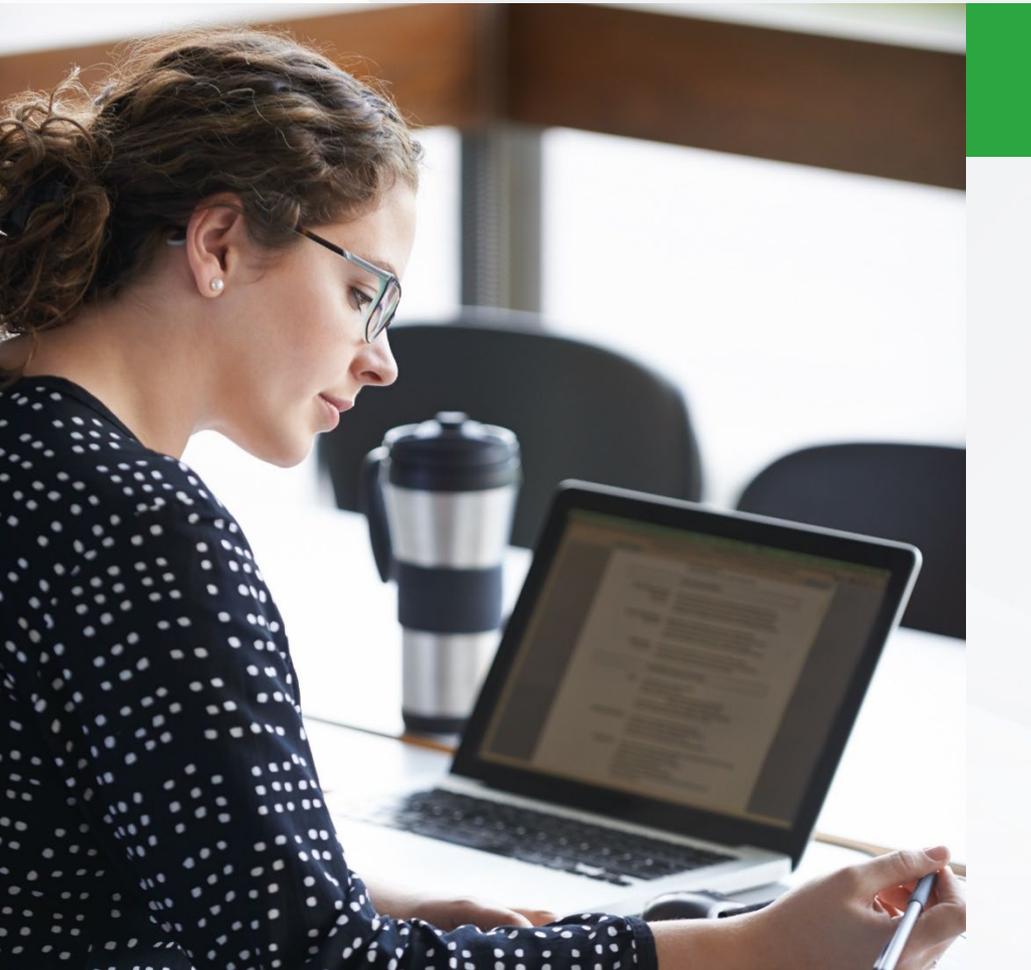


Recuerda la importancia de registrar la aplicación dentro de la consola de las API de Google para poder hacer uso de la API de Google Maps.

En caso de errores al ejecutar la aplicación del mapa:

- Asegúrate que tu aplicación tenga acceso a Internet.
- Si se ejecuta la aplicación en un emulador, verifica de que el emulador está ejecutando una versión de Android que incluye las API de Google.
- Verifica en la Consola de API de Google que la API de Google Maps para Android se ha habilitado en el panel Servicios.





- ¿Cuáles son los principales requisitos en Android para que una aplicación de Google Maps se pueda ejecutar correctamente?
- ¿Cuál es el proceso para obtener la API de Google?
- Al ejecutar la aplicación, ¿cuáles son los elementos que debes revisar para evitar errores?





Imagina que puedes diseñar aplicaciones por ejemplo para un usuario que requiere integrar mapas para su negocio, será muy agradable ofrecerle este servicio como parte de su aplicación. Ahora ya conoces los pasos necesarios para comenzar a utilizar los servicios de mapas de Google utilizando la API de Google Maps para Android.



Desarrollo de aplicaciones en plataforma Android

Sensor





Actualmente los dispositivos Android en su mayoría tienen sensores integrados que miden el movimiento, la orientación y diversas condiciones ambientales.

Estos sensores son capaces de proporcionar datos sin procesar con alta precisión y exactitud, y son útiles para supervisar el movimiento o posicionamiento tridimensional del dispositivo, o bien para supervisar los cambios en el entorno del dispositivo, como temperatura ambiental, la proximidad de los dedos del usuario y hasta la autenticación digital para acceder a una app.



Para tener acceso a los sensores es importante comprender su marco de trabajo. El marco de trabajo del sensor es parte del paquete [android.hardware](#) e incluye las siguientes clases e interfaces:

SensorManager

- Se usa esta clase para crear una instancia del servicio del sensor, ya que proporciona varios métodos para acceder a sensores y escucharlos, registrar y cancelar el registro de objetos de escucha de sensores de eventos y adquirir información de orientación.



Sensor

- Esta clase se usa para crear una instancia de un sensor específico (por ejemplo de presión, de proximidad, etc). Esta clase proporciona varios métodos que permiten determinar las capacidades de un sensor.

SensorEvent

- El sistema usa esta clase para crear un objeto de evento de sensor, que proporciona información sobre un evento del sensor.
- Un objeto de evento de sensor incluye información como: los datos sin procesar del sensor, el tipo de sensor que generó el evento, la exactitud de los datos y la marca de tiempo del evento.



SensorEventListener

- Se usa esta interfaz para crear dos métodos de devolución de llamada que reciben notificaciones (eventos del sensor) cuando cambian los valores del sensor o cuando cambia la exactitud del sensor.

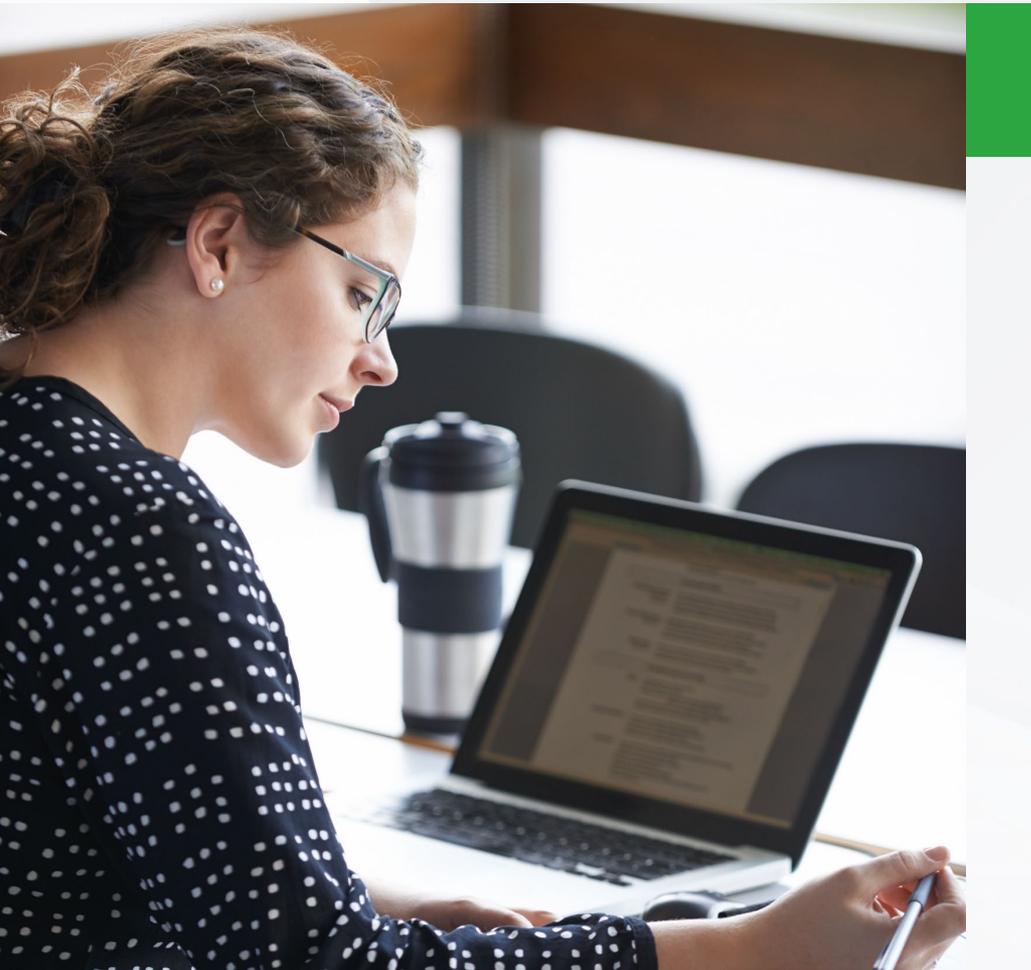
Recuerda que para identificar los sensores que están en un dispositivo, primero se obtiene una referencia al servicio del sensor. Creando una instancia de la clase **SensorManager** que llama al método **getSystemService()** y pasando el argumento **SENSOR_SERVICE**.

```
private SensorManager sensorManager;  
...  
sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
```



Al ejecutar una aplicación de sensor, recuerda la importancia de la API, ya que proporciona métodos que permiten determinar las capacidades de cada sensor, como el rango máximo, la resolución y los requisitos de potencia.





- ¿Cómo puede obtener datos a través de los sensores?
- ¿Cuál es la clase que permite escuchar al sensor?
- ¿Qué clase permite especificar qué tipo de sensor se está utilizando?





Actualmente se habla mucho del internet de las cosas, donde la conexión de los dispositivos móviles permiten maniobrar de manera más sencilla desde maquinaria en la industria y también en usos más personales.

Entre esas principales aplicaciones se encuentran la monitorización de las variables propias de un entorno o los condicionantes de rendimiento de una actividad, control de acceso a través de autenticación o bien el control del funcionamiento de una maquinaria.



Desarrollo de aplicaciones en plataforma Android

Publicación en Google Play





Ya has aprendido y practicado realizar diversas aplicaciones en el entorno Android, es hora de poner en marcha ese conocimiento y aventurarte a realizar una aplicación que decidas compartir con más usuarios.

Con el fin de que tu aplicación esté disponible para el público en general, es necesario crear un paquete para el lanzamiento que se pueda instalar y ejecutar en sus dispositivos Android.

Dicho paquete contiene los mismos componentes que el archivo APK de depuración (código fuente compilado, recursos, archivo de manifiesto y otros) y se compila usando las mismas herramientas de compilación. Sin embargo, a diferencia del archivo APK de depuración, el archivo APK listo para el lanzamiento se firma con tu propio certificado.



Para preparar tu app para el lanzamiento, por lo general se realizan varias tareas principales.

Cada tarea principal puede incluir una o más tareas más pequeñas según la forma en que lances tu aplicación.

Preparando tu app para el lanzamiento.

Configurar tu app para el lanzamiento

- Como mínimo, hay que eliminar las llamadas de Log y el atributo android:debuggable de tu archivo de manifiesto, si lo contiene.
- Revisa los requerimientos. Es posible que debas configurar varios ajustes para cumplir con los requisitos de Google Play o adaptar cualquier método que uses para lanzar tu aplicación.



Compilar y firmar una versión de actualización de tu app.

- Puedes usar los archivos de compilación de Gradle con el tipo de compilación release a fin de compilar y firmar una versión de actualización de la app. Consulta la sección Cómo compilar y ejecutar desde Android Studio.



Probar la versión de lanzamiento de tu app

- Antes de distribuir tu app, debes probar por completo la versión de actualización en al menos un teléfono celular y una tablet de destino.

Actualizar los recursos de la app para el lanzamiento

- Asegúrate de que todos los recursos de la app, como los archivos y gráficos multimedia, estén actualizados y sean visibles correctamente.

Preparar servidores y servicios remotos de los cuales dependa tu app

- Si tu app hace uso de servidores o servicios externos, se requiere que estés al pendiente de que sean seguros y estén listos para la producción.



Si vas a lanzar tu app a través de Google Play, considera:

Preparar materiales de promoción.

- Para aprovechar al máximo las capacidades de mercado y publicidad de Google Play, crea capturas de pantalla, videos, gráficos y texto promocional.

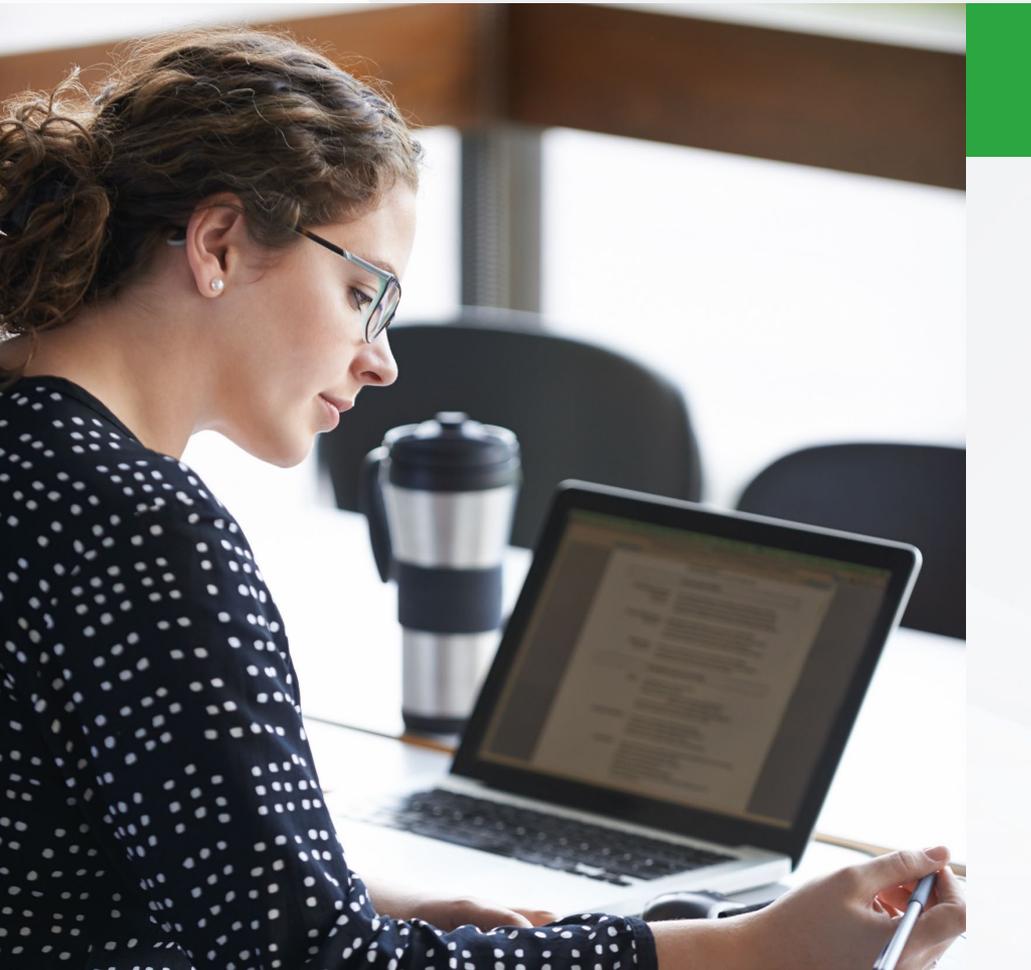
Configurar opciones y subir recursos

- Google Play te permite orientar tu app hacia un grupo de usuarios y dispositivos de todo el mundo. Cuando configuras varios ajustes de Google Play, puedes elegir los países a los que desees llegar, los idiomas que quieras usar y el precio que pretendes imponer en cada región. También puedes configurar detalles de la ficha, como el tipo de aplicación, la categoría y la clasificación del contenido.

Publicar la versión de lanzamiento de tu app.

- Una vez que los ajustes de la publicación están configurados correctamente y que la aplicación subida está lista para lanzarse al público, solamente hay que ir a la opción [Publicar de Play Console](#), y listo.





- Como programador ¿cuáles son los requisitos mínimos indispensables que debes revisar para publicar tus aplicaciones?
- ¿De qué manera puedes promocionar tu aplicación para que pueda llegar a más usuarios?





¿Estás listo para lanzar tu aplicación móvil?

Recuerda que simplemente debes seguir los pasos mencionados para asegurarte de que todo está cubierto para su lanzamiento.

¡Me da mucha emoción ver tu aplicación publicada!

