Manual Instalación AppMovil

Contenido

[**1.** **RESUMEN** 3](#_Toc202860769)

[**2.** **REQUERIMIENTOS DE SOFTWARE** 4](#_Toc202860770)

[**3.** **CREAR UN API KEY PARA EL PROYECTO** 4](#_Toc202860771)

[**4.** **INSTALAR FLUTTER Y DEPENDIENCIAS** 4](#_Toc202860772)

[**5.** **CONFIGURACIÓN** 4](#_Toc202860773)

# **RESUMEN**

El presente manual contempla los pasos necesarios, para instalar y configurar las dependencias que le permitirán generar los APK para la aplicación móvil, la que es un componente de la plataforma Urban Sensor, esta se orienta a la captura de incidencias en terreno, para este componente, se requiere utilizar Dart y Flutter, el presente manual, no contempla requerimientos de hardware.

# **REQUERIMIENTOS DE SOFTWARE**

En este apartado se describen los requisitos de software que se debe instalar y configurar antes de comenzar el proceso de despliegue, se recomienda usa las versiones exactas solicitadas en la tabla1 “requerimientos de software” que a continuación presentamos:

**Tabla 1 “Requerimientos de software”**

|  |  |  |
| --- | --- | --- |
| Tipo | Nombre | versión |
| Framework | Flutter | 3.29.3 |
| Herramientas | SDK | 3.7.2 |

A continuación, se explicará el proceso para configurar y desplegar la aplicación, este proceso se realiza en un equipo local, desde el que posteriormente generará los diferentes APK

# **CREAR UN API KEY PARA EL PROYECTO**

En este apartado, deben indicar como se crea un API KEY para Google Maps, indiquen desde como crear al usuario, como se accede a la consola, luego como crear la API, la API, debe considerar

Maps SDK for Android

Maps SDK for iOS

Places API

Geocoding API

Recuerden generarlo como un flujo paso a paso y colocar imágenes.

# **INSTALAR FLUTTER Y DEPENDIENCIAS**

En este apartado deben explicar paso a paso como se instala Flutter, recuerden en todo momento usar las versiones de Flutter y SDK que les indico en el apartado 2, al final de este apartado indiquen que deben descomprimir el archivo que colocarán en el drive, agreguen un manual para instalar Visual Studio Code y Android.

# **CONFIGURACIÓN**

En este punto si gustan pueden cambiar las imágenes,

Ahora realizaremos las configuraciones necesarias para que la aplicación funcione y quede vinculada a su servidor

En este punto deben recordar que ngnix se debe agregar la línea para el tamaño de los archivos de subida (sin no lo han hecho) y lo mismo para el Settings.py

# **5.1 Asignar la IP o Dominio**

Para comenzar diríjase al archivo “lib/config.dart” debe reemplazar el valor de la constante “baseUrl”, por la ip o dominio que tiene configurado en el servidor, donde reside la aplicación web, en este archivo además debería reemplazar el valor de la constante “defaultComuna” el nombre de la comuna para la que está desplegando la aplicación

Texto

El contenido generado por IA puede ser incorrecto.

Ahora diríjase a la siguiente ruta “android/app/src/main/res/xml” acá encontrará el archivo “network\_security\_config.xml”, en este archivo reemplace la IP que aparece, por ip que tiene configurado en el servidor, donde reside la aplicación web

Texto

El contenido generado por IA puede ser incorrecto.

# **5.2 Agregar API Key**

El primer paso es ir a al archivo “AndroidManifest.xml”, ubicado en “android/app/src/main/”, aproximadamente en la línea 42, reemplace el valor de “android:value” por la API\_KEY que generó en el punto 3.

Texto

El contenido generado por IA puede ser incorrecto.

Lo siguiente será ir al archivo “database\_helper.dart”, ubicado en “lib/helpers/”, aproximadamente en la línea 842, reemplace el valor de “googleApiKey” por la API\_KEY que generó en el punto 3.

Texto

El contenido generado por IA puede ser incorrecto.

Por último, se debe ir a modificar el archivo “boton\_modal\_mapa.dart”, ubicado en “lib/ widgets/”, aproximadamente en la línea 177, reemplace el valor de “googleApiKey” por la API\_KEY que generó en el punto 3.

Texto

El contenido generado por IA puede ser incorrecto.

Si tiene algún problema con las funcionalidades de los mapas, debe verificar que la API\_KEY allá sido colocada correctamente, de persistir los errores, verifique la configuración de su API\_KEY en la consola de Google.

# **5.3 Agregar Imagen**

Para comenzar diríjase al directorio “assets”, en este directorio, deberá cargar la imagen que desea usar en el “login” de la aplicación, por defecto, aparece “logoConst.png” elimine esta imagen y coloque la que usted desee, solo debe considerar que la extensión debe ser “png”

Texto

El contenido generado por IA puede ser incorrecto.

Una vez realizado este cambio, diríjase al archivo “pubspec.yaml” y reemplace el nombre de la imagen por defecto, por la que usted coloco en el directorio “assets”

Texto

El contenido generado por IA puede ser incorrecto.

Para finalizar debe asignar esta imagen, en el archivo “logo\_imagen.dart”, ubicado en “lib/ screens/ screen\_login/”, aproximadamente en la línea 17, reemplace el nombre de la imagen por defecto, por la imagen que usted agregó

Texto

El contenido generado por IA puede ser incorrecto.

Cabe señalar que si al renderizar la imagen, esta no se encuentra correctamente centrada, puede modificar los valores “top, left, width y height” para que cuadre en base a la imagen subida.

Con todo esto configurado, debería poder generar el APK, no obstante, le recomendamos que ejecute la aplicación en un dispositivo virtual, para corroborar que todo funcione.

Para generar el archivo APK, abra el proyecto, en Visual Studio Code, espere a que carguen las dependencias, luego presione “ctrl+ñ” para abrir la consola, luego ejecuta el comando “flutter build apk –release”

Texto

El contenido generado por IA puede ser incorrecto.

Al terminar el proceso, específicamente en la ruta “build\app\outputs\flutter-apk” se encontrará el archivo “app-release.apk”, el cual ya estaría listo para instalarlo en dispositivos Android.