

Moralar

# Manual de instalación para App Profesionales TTS

## Tabla de contenido


<b>Tabla de contenido</b>	<b>1</b>
<b>1. Detalles del proyecto</b>	<b>2</b>
1.1 Módulos que utiliza la app	2
1.2 Dependencias necesarias para su correcto funcionamiento	4
1.2.1 Chequear la raíz del proyecto:	4
1.2.2 Chequear que sí tengamos las librerías externas del proyecto:	4
<b>2. Requisitos para su instalación en local</b>	<b>5</b>
2.1 Requisitos antes de la instalación	5
2.2 Conocimientos previos	5
<b>3. Cómo ejecutar el programa en local</b>	<b>6</b>
3.1 Traer las librerías por la terminal	6
3.2 Instalar el emulador desde Android studio	6
3.3 Configurar el endpoint de la api	7
3.4 Abrir el emulador que instalamos	7
<b>4. Compilación del proyecto</b>	<b>9</b>
4.1 Compilar a APK	9

# 1. Detalles del proyecto

**App de profesionales TTS Moralar:** Es una aplicación móvil hecha en [Flutter](#) donde los asesores de tipo rol **Profesional TTS** podrán revisar el detalle de una familia, con toda la información importante para reagendar, cancelar agendamientos, extraer reportes de matchs, revisar los indicadores en el dashboard y mucho más.

## 1.1 Módulos que utiliza la app

- a. **Dashboard:** Nos sirve para revisar los indicadores como:
  - Total de familias inscritas
  - Porcentajes de etapas de agendamiento
  - Cantidad de Inmuebles disponibles para la compra
  - Cantidad de inmuebles vendidos
  
- b. **Familias:** Es el módulo principal de la aplicación donde podremos extraer un reporte de excel de las familias inscritas, también podremos ver el detalle de la familia donde encontraremos toda la información recopilada de la familia tales como:
  - Ver agendamientos programados de la familia
  - Revisar si ha respondido el cuestionario.
  - Revisar si ha realizado las encuestas
  - Revisar si está inscrito a algunos cursos
  
- c. **Agendamientos:** Es un módulo donde podremos:
  - Filtrar los agendamientos por estado, tipo de agendamiento etc.
  - Ver el detalle de un agendamiento con su color de estado correspondiente y su información detallada tanto del morador como los detalles del agendamiento.
  - Podremos reagendar y cancelar el agendamiento



d. **Matches:** En este módulo podremos:

- Filtrar por número de catastro, nombre del morador titular, cpf, código de inmueble etc.
- Extraer un reporte de matchs a excel
- Ver la información de cada match

e. **Cuestionarios:** Aquí podremos ver la lista de todos los cuestionarios y también responder el cuestionario que no haya sido llenado.

f. **Editar perfil:** Aquí se podrá cambiar la información básica del perfil profesional.

g. **Cambiar contraseña:** En este módulo podrán cambiar la contraseña de inicio de sesión en la app de profesionales tts.

## 1.2 Dependencias necesarias para su correcto funcionamiento

En esta sección se hablará de lo que se debe tener preparado antes de hacer una instalación en la máquina local.

### 1.2.1 Chequear la raíz del proyecto:

a. Opcion A:

- Clonar el proyecto el cuál ya le han pasado el repositorio, acceder a la rama “Development” y luego ingresar a la raíz del proyecto donde la carpeta se llama ***moralar-appts-flutter***.

Opción B:

- Acceder a la carpeta de insumos e ingresar a la carpeta ***moralar-appts-flutter***, dentro de esa carpeta será la raíz del proyecto.

- b. Ya teniendo en cuenta cuál es la raíz del proyecto, necesitaríamos revisar que dentro de esta carpeta sí se encuentra el archivo README.md, de esa forma sabemos que sí es la raíz del proyecto de la cuál es fundamental tener presente a la hora de ejecutar el proyecto en local.

### 1.2.2 Chequear que sí tengamos las librerías externas del proyecto:

Es fundamental tener estas carpetas que mencionaremos más adelante ya que dependemos de esas librerías para que el proyecto pueda ejecutarse bien.

- Paso 1: Revisar que fuera de la raíz del proyecto sí tengamos estas carpetas:
  - **moralar-widgets**: Esta librería personalizada contiene los widgets (componentes) reutilizables del aplicativo.
  - **mega-flutter**: Esta librería personalizada contiene las funcionalidades de autenticación y seguridad del aplicativo.
- Paso final: Ingresar al archivo ***pubspec.yaml*** que se encuentra en la raíz del proyecto ***moralar-appts-flutter*** y verificar que en la sección de **dependencies** sí tenga acceso a esas rutas como lo mencionamos en el paso 1.

## 2. Requisitos para su instalación en local

En esta sección se hablará de cómo poder ejecutar nuestra aplicación de flutter desde una máquina local de windows, teniendo en cuenta que debe cumplir con los conocimientos previos y haber realizado con éxito el índice 1.2.

### 2.1 Requisitos antes de la instalación

Se debe tener instalado estas versiones o las superiores para poder ejecutar el proyecto en la máquina local.

- a. **Dart:** [3.6.1](#)
- b. **Flutter:** [3.27.3](#)
- c. **Android studio:** [Cualquier versión](#) que soporte el SDK (  $\geq 2.19.0$   $< 3.0.0$  )
- d. **Visual studio code insiders:** [Cualquier versión](#)
- e. Extensiones de Vs-code:
  - Dart
  - Flutter
  - Flutter & dart Utilities (Opcional)
  - Flutter helpers (Opcional)
  - Flutter Intl (Opcional)
  - Flutter Snippets (Opcional)
- f. Configurar las variables de entorno [\(guía\)](#)

### 2.2 Conocimientos previos

1. Saber los comandos de [Dart pub](#).
2. Saber los comandos de [Flutter pub](#).
3. Saber ejecutar un emulador en android studio.
4. Saber instalar un emulador y diferenciar si el emulador es api 33 o api 34 etc.
5. Tener conocimientos de cómo ver los errores del output en vs code por si se presenta un error al compilar [\(guía\)](#).
6. Saber debuggear un proyecto de flutter desde vs code, [guía completa](#)

### 3. Cómo ejecutar el programa en local

En esta sección se explicará a modo resumen cómo debuggear el proyecto de flutter en vs code. Si desea revisar la guía completa a más detalle por favor revise este [link](#).

#### 3.1 Traer las librerías por la terminal

- Paso 1: Ejecutar Vs code insiders e ingresar a la raíz del proyecto.
- Paso 2: Ir a la pestaña View -> Terminal.
- Paso final: Ejecutar los siguientes comandos para **moralar-widgets**, **mega-flutter** y **moralar-apptts-flutter**. También asegurarse que si se completan con éxito:
  - `dart pub get`
  - `flutter pub get`

#### 3.2 Instalar el emulador desde Android studio

- Paso 1: Abrir android studio.
- Paso 2: Crear un nuevo decide en el device manager.
- Paso 3: Escoger cualquier teléfono, el que utilizamos para el desarrollo de Moralar es **Pixel 7 Pro**.
- Paso 4: Aplicar el botón siguiente.
- Paso 5: Escoger el api level **#34** (UpSideDownCake).
- Paso 6: Aplicar el botón siguiente.
- Paso final: Verificar la configuración y darle al botón finalizar.

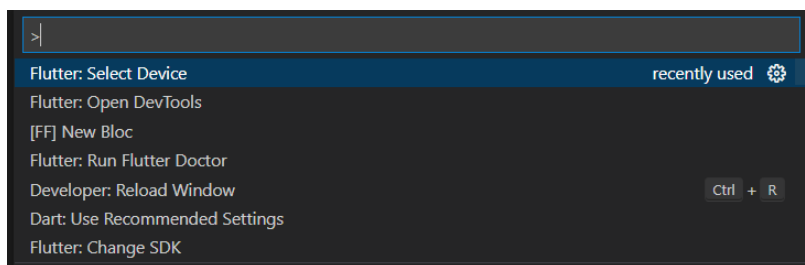
### 3.3 Configurar el endpoint de la api

En este apartado se configurará a que servidor y api le estaremos haciendo todo tipo de peticiones https desde nuestra aplicación de flutter.

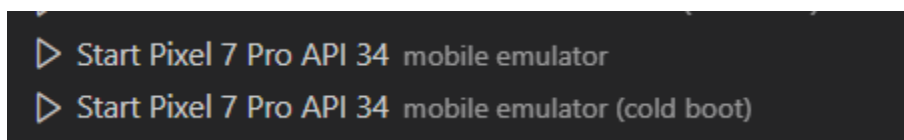
- Paso 1: Iremos a la carpeta de la librería personalizada de moralar que llamamos ***moralar-widgets/lib***.
- Paso 2: Ingresaemos al archivo de la carpeta lib llamado ***moralar\_widgets.dart***.
- Paso 3: Buscar la variable ***baseUrlForHTTP*** y cambiar el dominio de la url pero dejando intacto la ruta /api/v1.
- Paso 4: Buscar la variable ***baseUrlAssets*** y cambiar el dominio de la url pero dejando intacto la ruta /content/upload.
- Paso 5: Ejecutar los siguientes comandos para ***moralar-widgets***. También asegurarse que si se completen con éxito:
  - `dart pub get`
  - `flutter pub get`

### 3.4 Abrir el emulador que instalamos

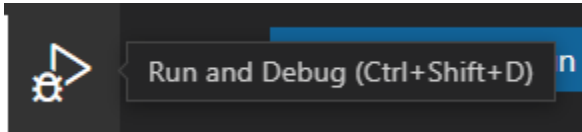
- Paso 1: Abrir vs code insiders desde la raíz del proyecto ***moralar-apptts-flutter***.
- Paso 2: ejecutar el atajo de teclado `Ctrl+shift+P` desde vs code insiders donde saldrá esta lista.



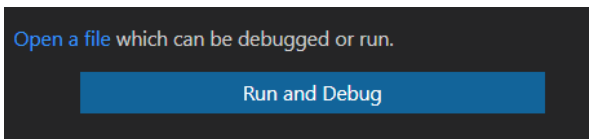
- Paso 3: Hacer clic en la opción: Flutter: Select Device.
- Paso 4: Escoger la opción del emulador que instalamos previamente en este caso el que utilizamos es: Start Pixel 7 Pro Api 34.



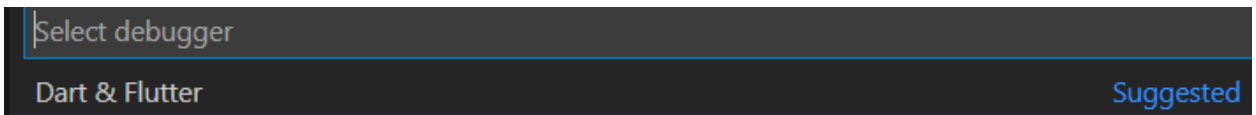
- Paso 5: Esperar a que se nos abra el emulador del dispositivo móvil y terminar la configuración básica del emulador (La configuración del dispositivo se hará solo una vez).
- Paso 6: ir a la sección de Run and debug de vs code insiders.



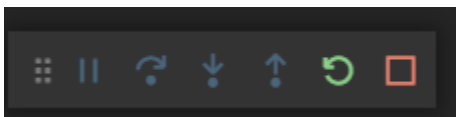
- Paso 7: Ejecutar el botón run and debug.



- Paso 8: Nos saldrá esta lista y escogeremos Dart & Flutter.



- Paso 9: Si todo está ok nos saldrá estos botones de debugeo.

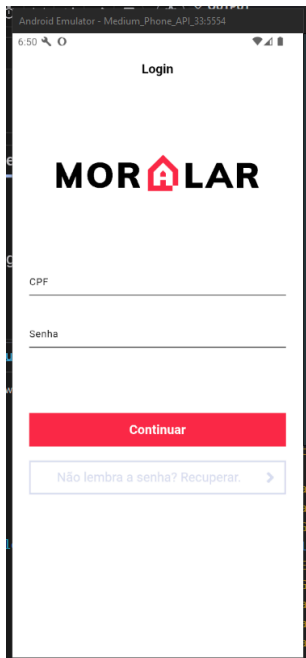


- Paso 10: Si no se ocultan los botones anteriores podremos revisar en la pestaña View -> Output cómo nos saldrá que el proyecto está cargando y luego de varios minutos se nos ejecutará en el emulador la aplicación que queríamos revisar.

Nota: tener en cuenta si salen letras rojas en el output y deberás revisarlas en busca de errores de instalación.



- Paso final: si todo sale correcto, nos saldrá el login de Moralar en nuestro emulador.



## 4. Compilación del proyecto

### 4.1 Compilar a APK

En esta sección hablaremos acerca de cómo compilar nuestro aplicativo de flutter a un Apk de android.

- Paso 1: Abrir la terminal desde la raíz de nuestro proyecto.
- Paso 2: ejecutar los comandos.
  - `flutter pub get`
  - `flutter build apk`
- Paso 3: Esperar a que se nos genere el apk e ir a la ruta que nos indica que el proceso de compilado finalizó exitosamente.

```
Running Gradle task 'assembleRelease'...  
✓ Built build\app\outputs\flutter-apk\app-release.apk (23.2MB)
```