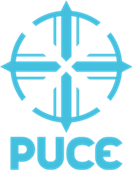
**PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR**

**SEDE ESMERALDAS**



**DESARROLLO BASADO EN PLATAFORMA**

**TEMA:**

APP WEB CON APACHE Y CORDOVA

**AUTOR:**

Andrés Carvajal Arias

**TUTOR:**

ING.JAIME SAYAGO

**ESMERALDAS – ECUADOR**

**2025 – 2026**

**Índice**

[Introducción 3](#_Toc201351022)

[Desarrollo: 4](#_Toc201351023)

[Apache Cordova 4](#_Toc201351024)

[Imagen 1. Instalación de extensión en Visual Studio Code. 4](#_Toc201351025)

[Android Studio 4](#_Toc201351026)

[Imagen 2. Instalación de Android Studio desde el sitio oficial. 4](#_Toc201351027)

[Imagen 3. Emulación de dispositivos. 5](#_Toc201351028)

[Imagen 4. Edición de variables para Android home. 5](#_Toc201351029)

[Imagen 5. Edición del Path. 5](#_Toc201351030)

[Imagen 6. Descarga de Gradle. 6](#_Toc201351031)

[Imagen 7. Variable de Gradle\_Home. 6](#_Toc201351032)

[Instalación de JQUERY 6](#_Toc201351033)

[Imagen 8. Instalación de jquery 6](#_Toc201351034)

[Uso de MockAPI 7](#_Toc201351035)

[Imagen 9. Uso de MoskApi. 7](#_Toc201351036)

[Imagen 10. Configuración del recurso “usuarios” 7](#_Toc201351037)

[Imagen 11. Configuración del recurso “restaurantes” 8](#_Toc201351038)

[Imagen 12. Implementación de la API. 8](#_Toc201351039)

[Imagen 13. Aplicación en el emulador. 9](#_Toc201351040)

[Dificultades: 10](#_Toc201351041)

[Conclusión 11](#_Toc201351042)

## Introducción

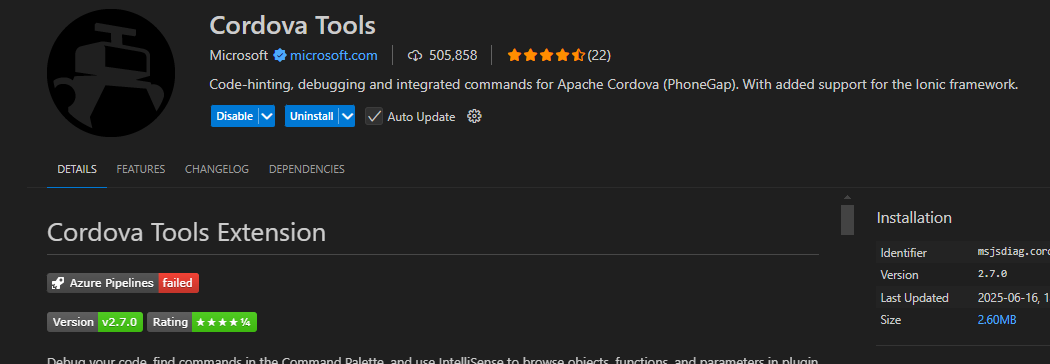
En este informe se documentan los pasos seguidos para el desarrollo de una aplicación móvil utilizando Apache Cordova, haciendo uso de una base de datos orientada a la gestión de restaurantes y usuarios.

Para ello, se llevó a cabo la instalación de tecnologías clave para el proyecto, el consumo de una API de prueba, y la utilización de MockAPI para una implementación sencilla del CRUD.

## Desarrollo:

**Instalación de tecnologías:**

### Apache Cordova



# Imagen 1. Instalación de extensión en Visual Studio Code.

Instalamos la extensión de “Cordova Tools” en Visual Studio Code para poder emplear correctamente Apache Cordova, tal como se observa en la imagen 1.

### Android Studio

Instalamos la Android Studio par a la emulación del “teléfono” ya que emplea la IDE para las pruebas.

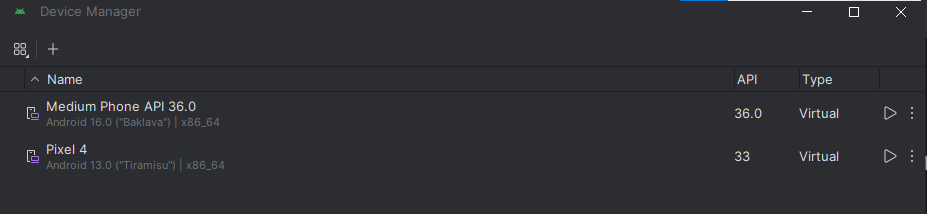
Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

# Imagen 2. Instalación de Android Studio desde el sitio oficial.

Tras la instalación se configuró el Device Manager para la creación de dispositivos

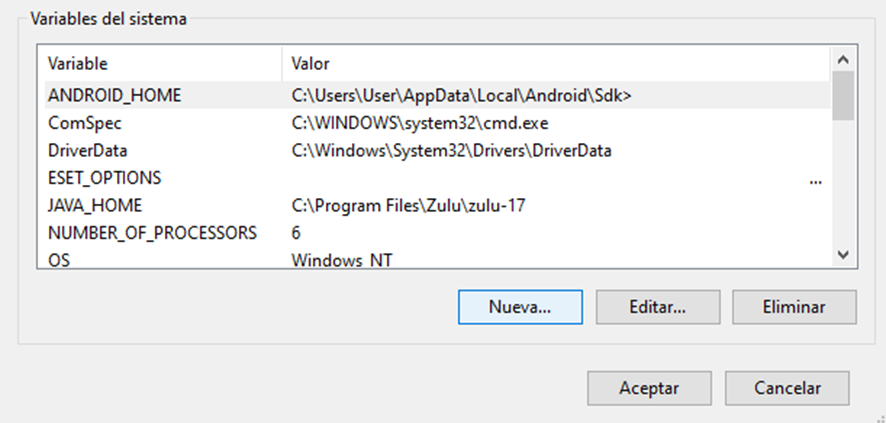
Virtuales.



# Imagen 3. Emulación de dispositivos.

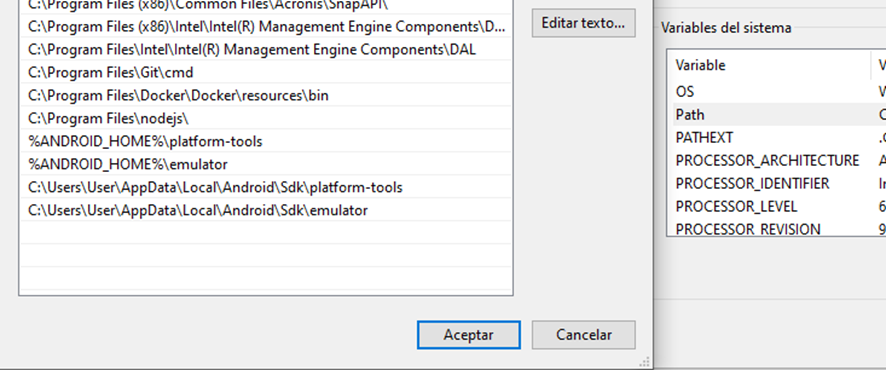
En la imagen 3 se pueden observar los dispositivos que hemos creado y que podremos emular para las pruebas.

Para garantizar el correcto funcionamiento de la IDE debemos configurar las variables de entorno.



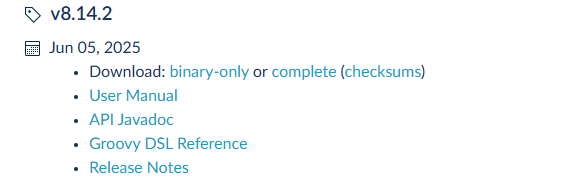
# Imagen 4. Edición de variables para Android home.

Configuramos las variables de entorno de nuestra maquina con las que se observan en la Imagen 4, además de editar ciertas rutas en el Path para el correcto funcionamiento del proyecto.



# Imagen 5. Edición del Path.

Se edita el Path para que todo funcione bien en la máquina, incluimos la carpeta “platform-tools” en el path del sistema, permitiendo que Cordova pueda acceder a los componentes necesarios del entorno Android para las pruebas.Gradle



# Imagen 6. Descarga de Gradle.

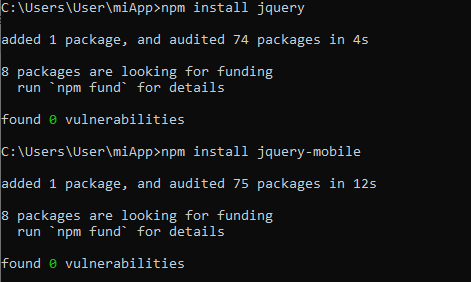
Es una herramienta la cual permite al proyecto la compilación por Android Studio, para construir la aplicación, en mi caso no se incluyó al momento de instalar Android Studio, por lo tanto, se descargó desde el sitio oficial.



# Imagen 7. Variable de Gradle\_Home.

Y se configuro en las variables de entorno el Gradle Home.

### Instalación de JQUERY

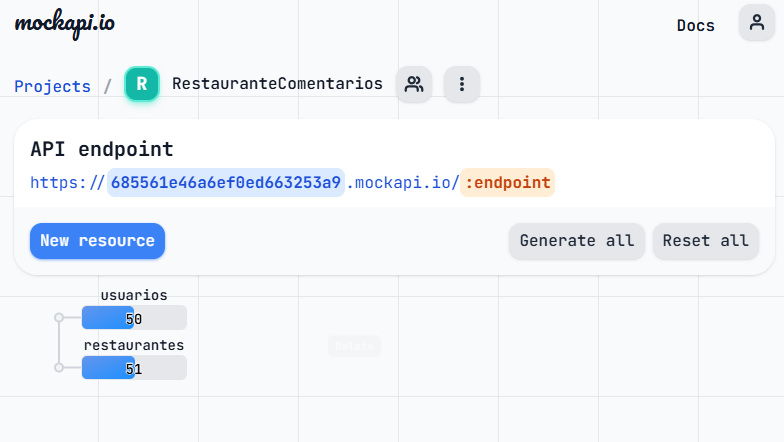


# Imagen 8. Instalación de jquery

Agregando “npm install jquery” al proyecto JQuery e instalando paquetes y “npm install jquery-mobile” ambos paquetes, quedando disponibles para el uso dentro del proyecto.

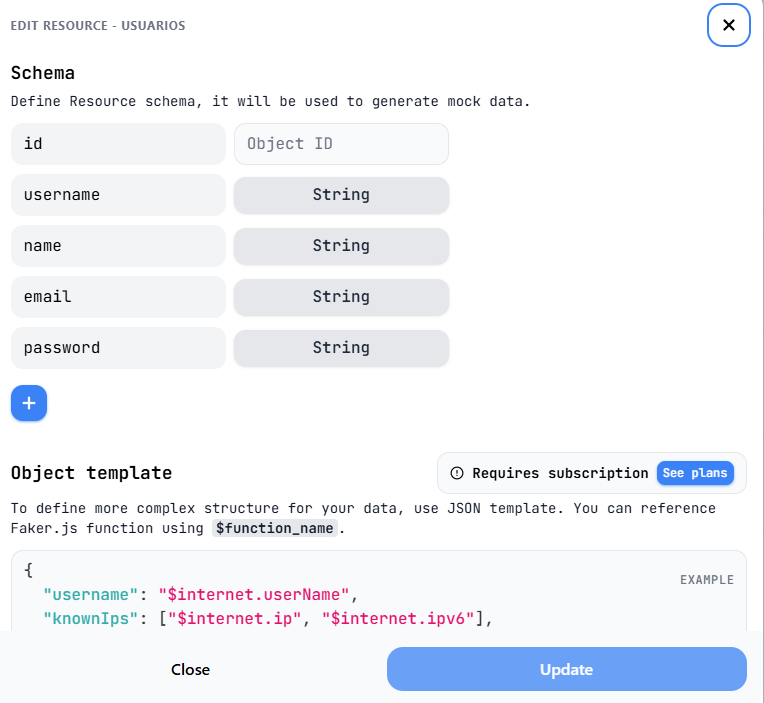
### Uso de MockAPI

En el backend se usó una API de MoskApi



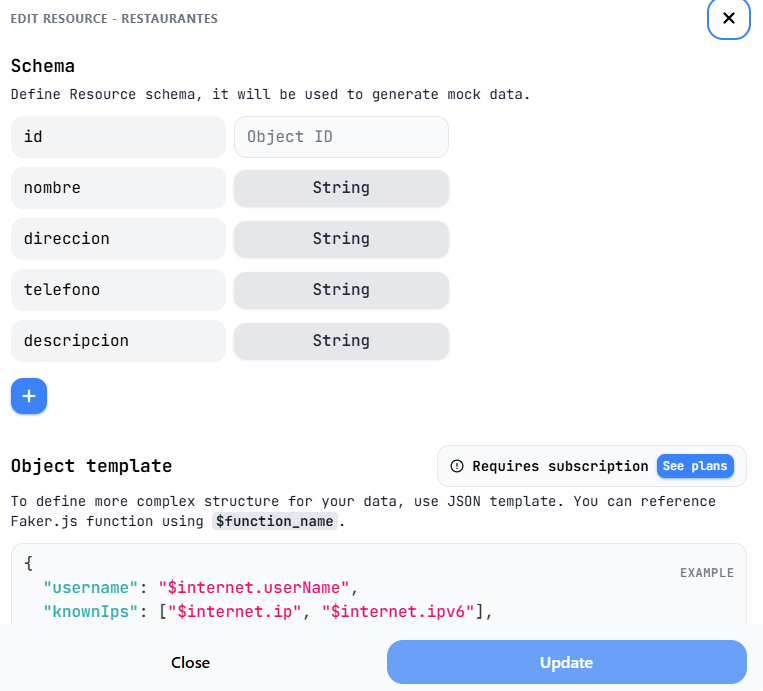
# Imagen 9. Uso de MoskApi.

Como se observa tenemos dos recursos, en este caso “usuarios” y “restaurantes” hasta el momento el plan gratuito solo deja tener 2 recursos, por ello no se implementó otro.



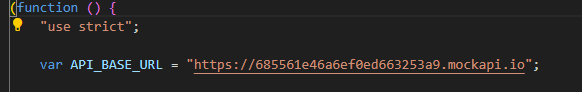
# Imagen 10. Configuración del recurso “usuarios”

Como se observa en la imagen 10, se creó para generar id, username, name, email, password y todos con valor String, a excepción de la id.



# Imagen 11. Configuración del recurso “restaurantes”

Como se observa en la imagen 11, se creó para generar id, nombre, dirección, teléfono, descripción y todos con valor String, a excepción de la id.



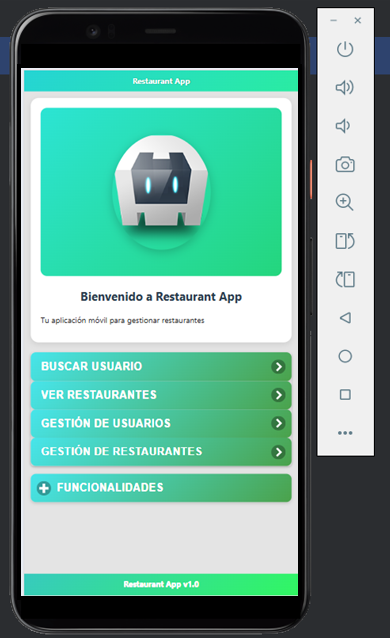
# Imagen 12. Implementación de la API.

Añadimos la API que nos de mockAPI, y la implementamos en nuestro código.

Después utilizamos los comandos:

* Cordova Clean Android
* Cordova Build Android
* Cordova Run Android

El comando cordova clean android se encarga de limpiar los archivos temporales y las carpetas de compilación generadas previamente, lo cual es útil para evitar conflictos o errores acumulados durante el desarrollo. A continuación, el comando cordova build android permite compilar la aplicación, generando el archivo ejecutable (.apk) necesario para su instalación. Finalmente, el comando cordova run android no solo compila la aplicación, sino que además la instala y ejecuta directamente en un dispositivo físico o emulador conectado, facilitando así las pruebas en tiempo real.



# Imagen 13. Aplicación en el emulador.

Tras usar los comandos anteriormente mencionados podremos correr el emulador y veremos nuestro proyecto.

## Dificultades:

En mi caso presente varias dificultades, entre ellas las descargas de las tecnologías puesto que en mi casa hay varias personas conectadas a una sola red y demoro mucho la instalación, la limitación de mi maquina ya que no ejecuta rápidamente los programas y varias veces se me “colgó” el emulador, y lo más destacable fueron los errores de variables y su configuración dentro de path, ya que tenía que editar y configurar bien las rutas, caso contrario no se ejecutaría el programa.

## Conclusión

Tras la culminación de la práctica relacionada con la aplicación web desarrollada durante el proyecto, se logró implementar exitosamente su versión móvil utilizando Apache Cordova. Esta aplicación móvil demostró un consumo eficiente de la API REST proporcionada por MockAPI, funcionando correctamente con las operaciones CRUD implementadas. A lo largo de este proceso, se adquirió una comprensión práctica de los fundamentos del consumo de APIs y la gestión de datos en tiempo real.

La experiencia adquirida en este proyecto sirvió como base para el desarrollo de futuras aplicaciones, que podrán beneficiarse de herramientas más avanzadas y diseños más depurados. Aunque se trató de una práctica académica, el enfoque aplicado permitió fortalecer habilidades clave en el manejo de diversas arquitecturas y tecnologías.