



MANUAL TÉCNICO DEL SISTEMA

Elaborado por:
Martín Bojorque
Iván Ulloa

Tabla de contenido

RESUMEN	1
OBJETIVO	1
INTRODUCCIÓN	2
1. ASPECTOS TEÓRICOS DEL DESARROLLO	3
1.1. CONCEPTOS GENERALES	3
1.1.1. Apps de organización personal	3
1.1.2. Analítica de datos	3
1.1.3. Inteligencia de negocios	3
1.2. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO	3
1.2.1. Node.js.....	3
1.2.2. Librerías adicionales utilizadas.....	3
1.2.3. Ionic Framework.....	4
1.2.4. Firebase.....	4
2. DIAGRAMAS DE MODELADO	5
2.1. ESQUEMA GENERAL DE LA APLICACIÓN	5
Figura 1. <i>Esquema de la aplicación</i>	5
2.2. FUNCIONALIDADES Y ROLES	6
Figura 2. <i>Esquema de funcionalidades</i>	6
2.3. DICCIONARIO DE DATOS	6
Tabla 1. <i>Diccionario de datos</i>	6
2.4. DIAGRAMA DE COMPONENTES	8
Figura 3. <i>Diagrama de componentes</i>	8
2.5. EESQUEMA DE FUNCIONALIDAD DE CLOUD FUNCTIONS	8
Figura 4. <i>Despliegue de código para estimación de presupuesto</i>	9
3. ASPECTOS TÉCNICOS DEL DESARROLLO	9
3.1. MODIFICACIÓN LOCAL	9
3.1.1. Node.js.....	9
3.1.2. Angular.....	10
3.1.3. Visual Studio Code.....	11
3.1.4. Paquetes Adicionales y plugins	11

3.1.5. Android Studio.....	12
3.1.5.1. SDK.....	13
3.1.5.2. Device Manager	13
3.1.6. Ionic Framework.....	13
3.1.7. Capacitor.....	14
4. REQUERIMIENTOS DEL SISTEMA	14
5. BIBLIOGRAFÍA	14

RESUMEN

Se detallan aspectos técnicos del desarrollo de la aplicación Gestionar Gastos esclareciendo aspectos como la estructura del sistemas, comportamiento, funcionalidades, plataformas y herramientas usadas. El manual contiene información de teórica, modelado, aspectos técnicos específicos y requerimientos del sistema. Aspectos técnicos y teóricos abordados: Node.js y algunas librerías usadas, Ionic Framework, Google Firebase y sus funciones.

OBJETIVO

Dar a conocer los conocimientos generales empleados en el desarrollo de la aplicación Gestionar Gastos en conjunto con descripciones de la estructura y funcionalidad de la aplicación dando un entendimiento global del sistema al lector.

INTRODUCCIÓN

El objetivo de este documento es brindar una guía técnica de la Aplicación Gestionar Gastos a futuros desarrolladores que busquen administrar, editar, configurar o dar mantenimiento al sistema apropiadamente. A continuación, se detallan las secciones a tratar:

- **ASPECTOS TEÓRICOS:** Dar a conocer definiciones, conceptos y conocimiento básico teórico acerca de los componentes que conforman el desarrollo del aplicativo. Estos conocimientos incluyen plataformas, editores, librerías y herramientas utilizadas en la construcción del sistema.
- **DIAGRAMAS DE MODELADO:** Algunos diagramas que describen el comportamiento y funcionamiento de la aplicación.
- **ASPECTOS TÉCNICOS DEL DESARROLLO:** Guías que busca instruir al lector sobre los componentes del sistema. Los temas incluyen almacenamiento de datos, configuración de plataformas de desarrollo, estructura de la aplicación y recomendaciones de uso.
- **REQUERIMIENTOS DEL SISTEMA:** Detalla los requerimientos de software y hardware de las herramientas de desarrollo y el aplicativo.

1. ASPECTOS TEÓRICOS DEL DESARROLLO

La aplicación Gestionar Gastos busca dar al usuario una herramienta para llevar registro de sus gastos personales y de otros miembros de la familia. Esto combinado con una gestión de presupuestos, historiales de gastos, visualizaciones gráficas de los gastos y una predicción global de los gastos.

1.1. CONCEPTOS GENERALES

1.1.1. Apps de organización personal

Es una categoría de aplicaciones móviles que permiten mantener un registro de actividades de toda índole de esta manera gestionando actividades: diarias, semanales y anuales; con el objetivo de mantener una vida más organizada, controlar ciertos aspectos de manera rigurosa y/o generar buenos hábitos de comportamiento.

1.1.2. Analítica de datos

Busca analizar datos para obtener nuevo conocimiento que afecte la influencia que tiene sobre estos a aquel que los estudia. En el contexto de informática, análisis se refiere a la aplicación de sistemas informáticos al análisis de grandes volúmenes de datos para toma de decisiones (Runkler, 2020).

1.1.3. Inteligencia de negocios

La inteligencia empresarial es un término que se creó a mediados de la década de 1990 para describir la acción de tomar datos en su forma original y convertirlos en algo utilizable, lo cual puede ser la base de decisiones empresariales. Es un término que acoge a otras varias disciplinas de datos como: minería de datos, análisis estadístico, pronósticos y apoyo a la toma de decisiones (Pareek, 2006).

1.2. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO

1.2.1. Node.js

Concebido como un entorno de código abierto y multiplataforma controlado por eventos en tiempo de ejecución del lado del servidor utilizando JavaScript, Node.js se enfoca en el desarrollo de aplicaciones escalables, permitiendo establecer y gestionar varias conexiones simultáneas de manera asíncrona, la comunicación es simplificada al no usar subprocesos y no depender de multi-threading para soportar la ejecución de la lógica de negocios.

1.2.2. Librerías adicionales utilizadas

Todas las librerías fueron obtenidas usando el gestor de paquetes de Node.js o también conocido como NPM.

- **regresions-js**: Es un módulo de JavaScript que contiene una colección de métodos lineales basados en mínimos cuadrados para análisis simple de datos. Contiene código para regresiones lineales, exponenciales, logarítmicas y polinomiales.
- **Chart.js**: Librería gratuita y de código abierto de JavaScript enfocada en la construcción de gráficos utilizando la HTML como base. Soporta varios tipos de gráficos como: Barras, área polar, burbuja, líneas, dispersión, etc.

- **firebase:** Librería que provee herramientas para incorporar todos los servicios e infraestructura de Google Firebase a tu proyecto.

1.2.3. Ionic Framework

De la documentación oficial proveída por Ionic se tiene que: *“Ionic es un kit de herramientas de código abierto para el desarrollo de apps móviles y de escritorio de alta calidad y desempeño utilizando tecnologías web HTML, CSS y JavaScript con integraciones de Frameworks como Angular, React y Vue”*. Caracterizada por permitir el desarrollo de aplicaciones como una página web estándar, siendo capaz de correr en varias plataformas incluidas Android y iOS, esto en conjunto con la disponibilidad de varios plugins para el uso de servicios extra y hardware específico dentro del dispositivo crean un entorno amigable de desarrollo que resulta en aplicaciones estéticamente agradables con alto nivel de personalización (Chaudhary, 2018). Requiere conocimientos previos de Angular (también permite el desarrollo en Vue o React).

1.2.4. Firebase

Firebase es una plataforma propietaria de Google que ofrece herramientas para el desarrollo de aplicaciones web y móviles, su oferta incluye infraestructura de backend completamente administrada, supervisión del rendimiento de aplicaciones, generación de estadísticas de los usuarios y uso de recursos. Las características que son ofrecidas pueden ser activadas o desactivadas de acuerdo con las necesidades de los desarrolladores y los precios varían dependiendo del uso de recursos y almacenamiento.

- **Cloud Functions para Firebase:** De acuerdo con la documentación oficial: “Es un framework sin servidores que te permite ejecutar de forma automática el código de backend en respuesta a las solicitudes HTTPS. Tu código JavaScript o TypeScript se almacena en la nube de Google y se ejecuta en un entorno administrado”
- **Cloud Firestore:** De acuerdo con la documentación oficial: “Usa nuestra base de datos NoSQL flexible, escalable y en la nube a fin de almacenar y sincronizar datos para el desarrollo tanto del lado del cliente como del servidor”. Existe también la posibilidad de la implementación de una base de datos en tiempo real para mantener los datos sincronizados entre apps junto con soporte sin conexión.
- **Firebase Authentication:** De acuerdo con la documentación oficial: “Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook y Twitter, y mucho más”

2. DIAGRAMAS DE MODELADO

2.1. ESQUEMA GENERAL DE LA APLICACIÓN

Visualización general de la estructura del sistema en relación con las funciones ofrecidas por Firebase.

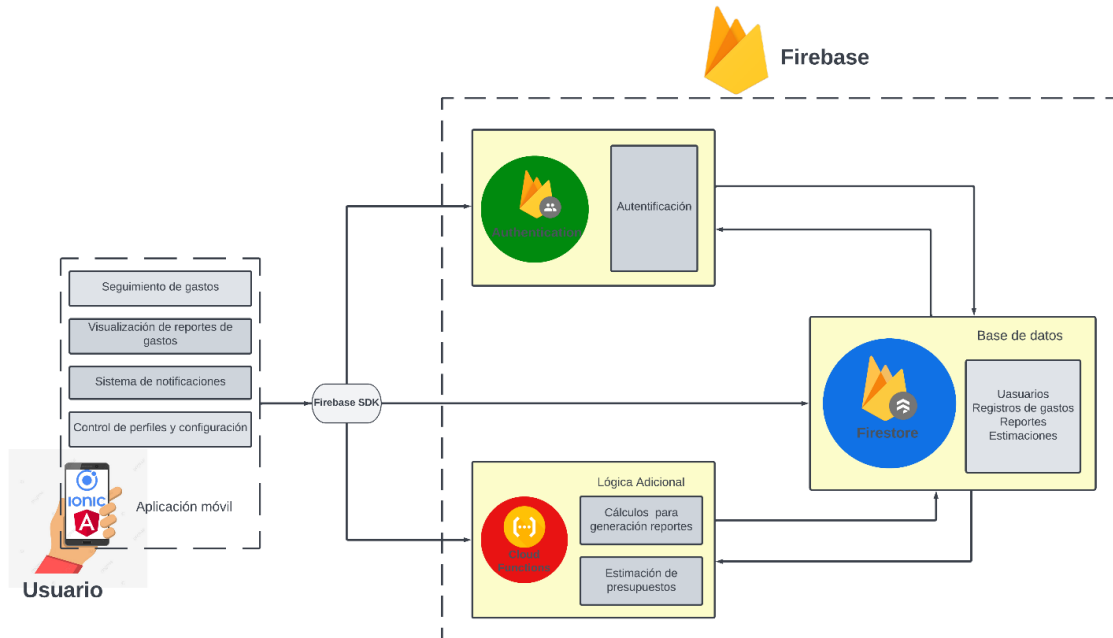


Figura 1. Esquema de la aplicación

2.2. FUNCIONALIDADES Y ROLES

Esquema de las funcionalidades ofrecidas dependiendo de acuerdo con los roles asignados. Roles existentes: Miembros de la familia y jefe del hogar.



Figura 2. Esquema de funcionalidades

2.3. DICCIONARIO DE DATOS

Disposición de los datos guardados en la base de datos no relacional de Firestore. La información incluye las entidades (y los nombres de las colecciones en Firestore), los nombres de los atributos, el tipo de datos y una descripción.

Tabla 1. Diccionario de datos

Colección	Atributos	Tipo de dato	Descripción
Usuario (users)	uid	Cadena de texto	Hash como ID del documento
	email	Cadena de texto	Correo electrónico de la cuenta
	password	Cadena de texto	Contraseña de la cuenta
	role	Char	"U" para miembro de la familia "A" para jefe de hogar
	displayName	Cadena de texto	Nombre de propietario de la cuenta
	description	Cadena de texto	Mensaje de bienvenida de la cuenta
	photoURL	Cadena de texto	URL de referencia a Firebase Storage
	createdAt	Fecha	Fecha en la que fue creada la cuenta
	lastLogin	Fecha	Fecha de último ingreso a la cuenta
	provider	Cadena de texto	Tipo de inicio de sesión "gestion-gastos" para ingreso con correo y contraseña

			"google" inicio de sesión con Google
	id_familia	Numérico	Familia asignada a la cuenta
	active	Booleano	Cuenta activa o no
Familia (families)	id	Cadena de texto	Hash como ID del documento
	nombre	Cadena de texto	Nombre de la familia
	presupuesto_global	Numérico	Presupuesto global de la familia
	primer_dia_mes	Numérico	Día de inicio de mes
Presupuesto (presupuestos)	id	Cadena de texto	Hash como ID del documento
	cantidad	Numérico	Monto del presupuesto
	id_categoria	Cadena de texto	Categoría de gastos
	id_familia	Cadena de texto	Familia asignada al presupuesto
	fecha	Fecha	Fecha de modificación del presupuesto
Gasto (gastos)	id	Cadena de texto	Hash como ID del documento
	monto	Numérico	Monto del gasto
	descripcion	Cadena de texto	Razón del gasto
	fecha	Fecha	Fecha de realización del gasto
	id_categoria	Cadena de texto	Categoría de gastos
	id_usuario	Cadena de texto	ID de usuario del gasto
	id_familia	Cadena de texto	Familia asignada al gasto
Categoría (categories)	id	Cadena de texto	Hash como ID del documento
	nombre	Cadena de texto	Nombre de la categoría
	descripción	Cadena de texto	Descripción de la categoría

2.4. DIAGRAMA DE COMPONENTES

Esquema de los componentes (módulos) de la aplicación junto con sus relaciones de dependencia con las características del back-end (Firebase).

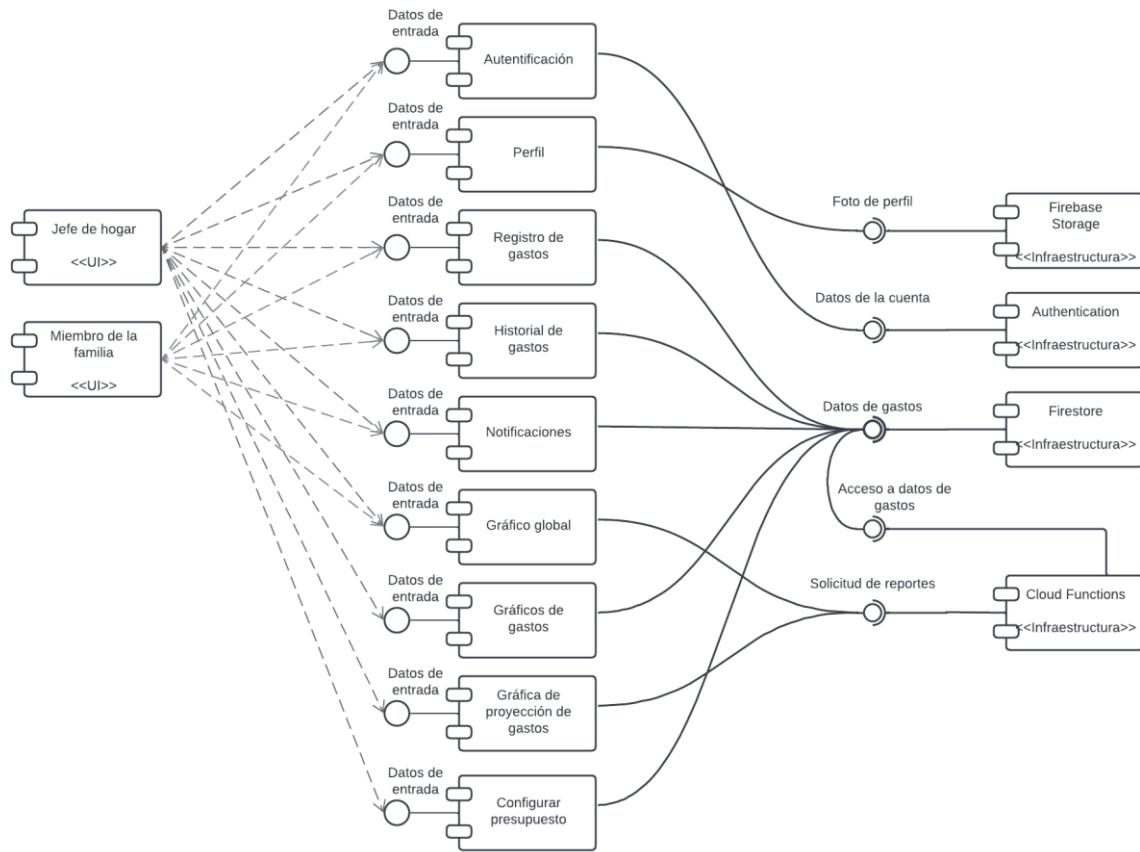


Figura 3. Diagrama de componentes

2.5. EESQUEMA DE FUNCIONALIDAD DE CLOUD FUNCTIONS

Describe brevemente el uso de Cloud Functions dentro de la aplicación de manera visual en donde el código desarrollado en el entorno de Node.js es desplegado en los servidores de Google y estos servicios son solicitados por la aplicación.

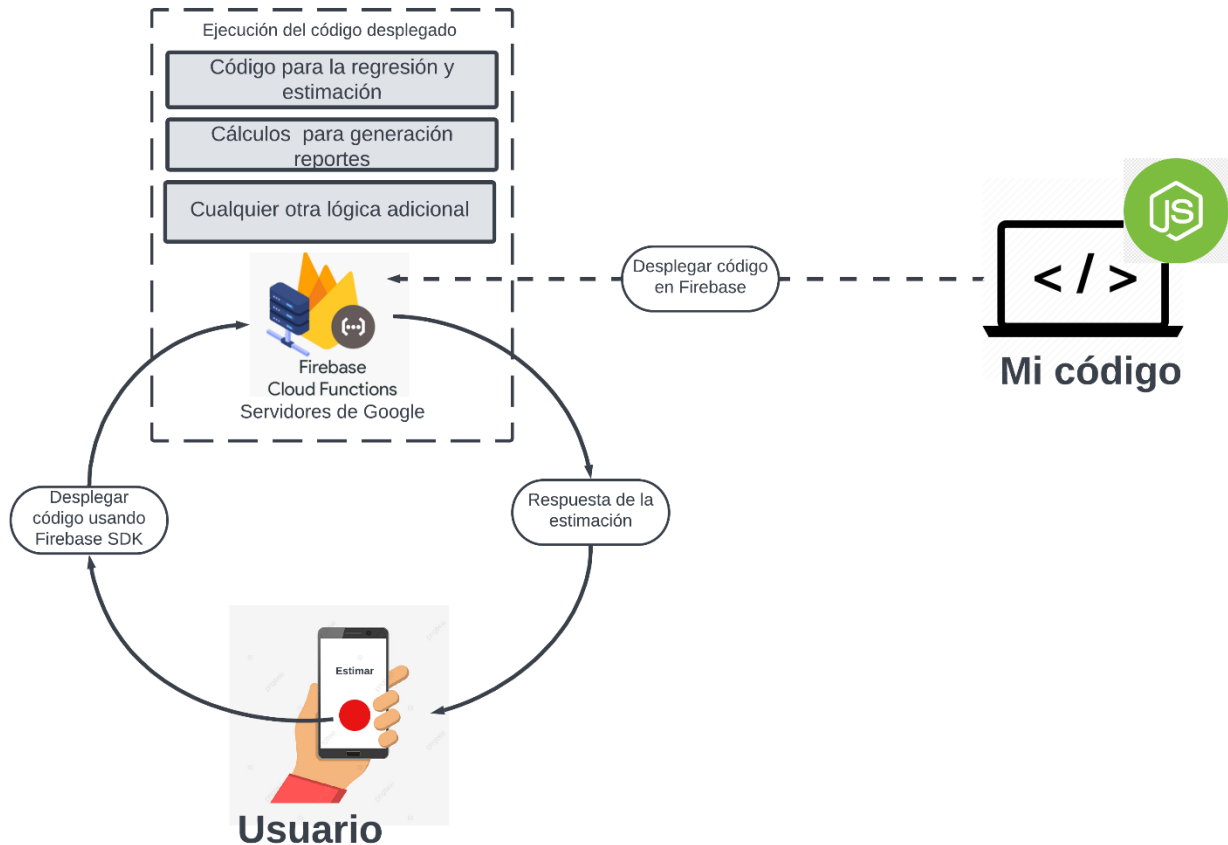


Figura 4. Despliegue de código para estimación de presupuesto

3. ASPECTOS TÉCNICOS DEL DESARROLLO

En esta siguiente sección se realizará una descripción detallada sobre los aspectos técnicos del aplicativo, estos están relacionados con la instalación de las herramientas necesarias para realizar modificaciones requeridas.

3.1. MODIFICACIÓN LOCAL

3.1.1. Node.js

Se requiere de la instalación de Nodejs 16.13.1, se recomienda la instalación de esta versión en específico. (<https://nodejs.org/download/release/v16.13.1/>)

Index of /download/release/v16.13.1/

../	30-Nov-2021 19:33	-
docs/	30-Nov-2021 19:42	-
win-x64/	30-Nov-2021 19:41	-
win-x86/	01-Dec-2021 16:13	3153
SHASUMS256.txt	01-Dec-2021 16:13	3690
SHASUMS256.txt.asc	01-Dec-2021 16:13	310
SHASUMS256.txt.sig	01-Dec-2021 11:51	43966837
node-v16.13.1-aix-ppc64.tar.gz	01-Dec-2021 11:33	28506601
node-v16.13.1-darwin-arm64.tar.gz	01-Dec-2021 11:33	18427632
node-v16.13.1-darwin-arm64.tar.xz	01-Dec-2021 12:14	29938975
node-v16.13.1-darwin-x64.tar.gz	01-Dec-2021 12:15	20056120
node-v16.13.1-darwin-x64.tar.xz	01-Dec-2021 11:27	562423
node-v16.13.1-headers.tar.gz	01-Dec-2021 11:27	381248
node-v16.13.1-headers.tar.xz	01-Dec-2021 11:15	32385016
node-v16.13.1-linux-arm64.tar.gz	01-Dec-2021 11:16	20985584
node-v16.13.1-linux-arm64.tar.xz	01-Dec-2021 12:31	29679419
node-v16.13.1-linux-armv7l.tar.gz	01-Dec-2021 12:32	17985684
node-v16.13.1-linux-armv7l.tar.xz	01-Dec-2021 11:16	34444979
node-v16.13.1-linux-ppc64le.tar.gz	01-Dec-2021 11:18	22100280
node-v16.13.1-linux-ppc64le.tar.xz	01-Dec-2021 11:15	32655582
node-v16.13.1-linux-s390x.tar.gz	01-Dec-2021 11:16	20575440
node-v16.13.1-linux-s390x.tar.xz	01-Dec-2021 11:29	32395116
node-v16.13.1-linux-x64.tar.gz	01-Dec-2021 11:30	21606428
node-v16.13.1-linux-x64.tar.xz	01-Dec-2021 11:40	16721509
node-v16.13.1-win-x64.7z	01-Dec-2021 11:40	25689871
node-v16.13.1-win-x64.zip	01-Dec-2021 11:38	15557278
node-v16.13.1-win-x86.7z	01-Dec-2021 11:38	24039967
node-v16.13.1-win-x86.zip	01-Dec-2021 11:40	28020736
node-v16.13.1-x64.msi	01-Dec-2021 11:38	26267648
node-v16.13.1-x86.msi	01-Dec-2021 11:48	55634645
node-v16.13.1.pkg	01-Dec-2021 11:19	64385859
node-v16.13.1.tar.gz	01-Dec-2021 11:24	33907544
node-v16.13.1.tar.xz		

Mediante esta página web se obtiene el archivo de instalación correspondiente para cada sistema operativo que puede hacer uso de nodejs y la arquitectura necesaria (X86 o X64).

3.1.2. Angular

Adicionalmente para la modificación es necesario la instalación de Angular 13.1.2.

```

Angular CLI

ngular CLI: 13.1.2
ode: 16.13.1
ackage Manager: npm 8.1.2
S: win32 x64

ngular:
..

ackage          Version
-----
angular-devkit/architect    0.1301.2 (cli-only)
angular-devkit/core        13.1.2 (cli-only)
angular-devkit/schematics  13.1.2 (cli-only)
schematics/angular        13.1.2 (cli-only)

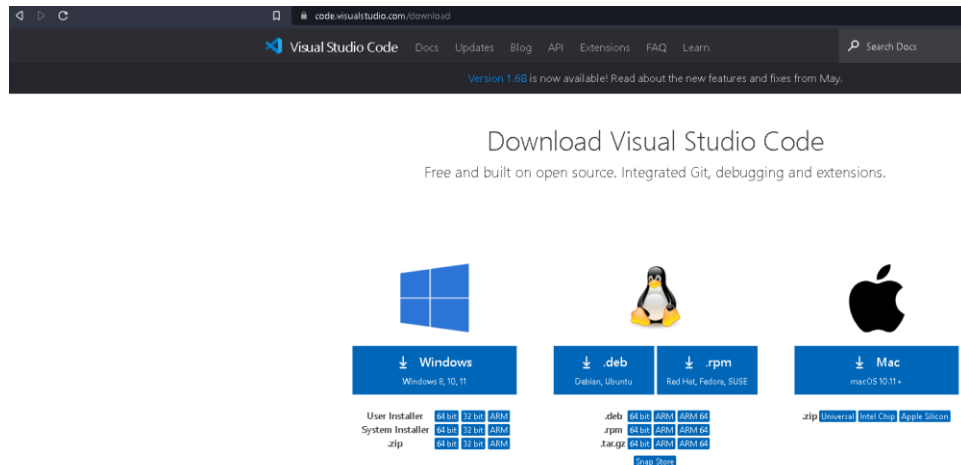
```

Esto se realiza desde la consola de comando, mediante el comando `npm install @angular/cli` y especificando la versión requerida.

3.1.3. Visual Studio Code

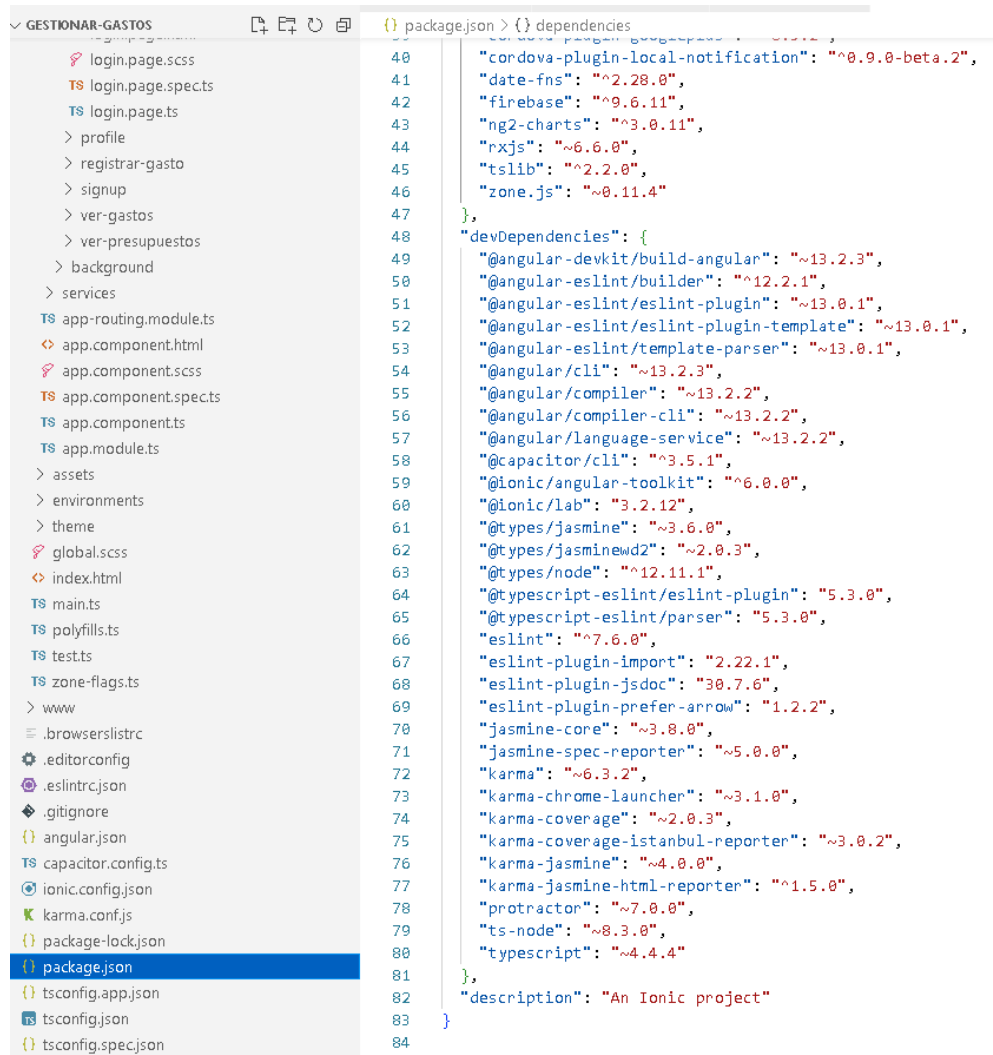
El editor de texto utilizado para el desarrollo del proyecto es Visual Studio Code. Para ello se pueden encontrar las diferentes versiones de este en la apgina web. <https://code.visualstudio.com/download>

Esta herramienta se puede utilizar en varios sistemas operativos.



3.1.4. Paquetes Adicionales y plugins

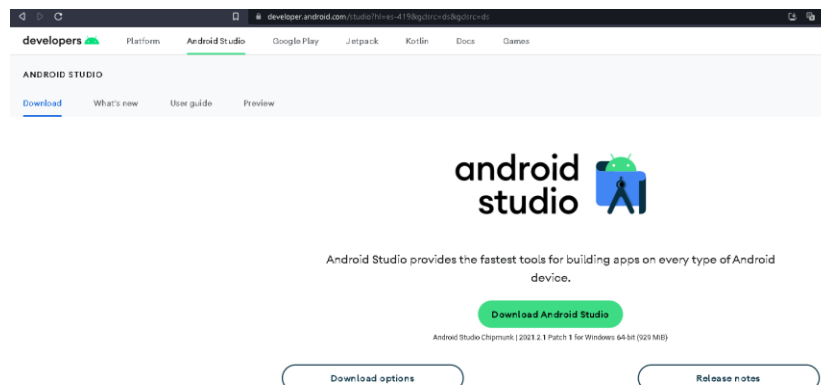
Para este paso es necesario haber descargado el proyecto, desde Github. Dentro de este se encuentra el archivo `package.json`, este archivo contiene todos los paquetes y plugin que se utilizaron para el proyecto. Ubicados en la carpeta "gestionar-gastos" desde la consola se utiliza el código `npm install`, este se encargará de instalar los paquetes descritos en este.



3.1.5. Android Studio

Para la depuración y ejecución del proyecto en la plataforma Android es necesario la instalación de Android Studio y varias funcionalidades dentro de este.

Para comenzar la herramienta está disponible en <https://developer.android.com/studio?hl=es-419&gclid=ds&gclid=ds>

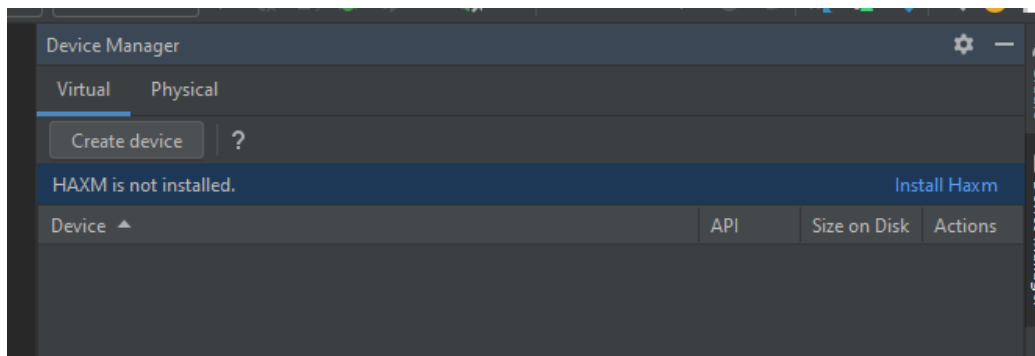


3.1.5.1. SDK

<input type="checkbox"/>	Android TiramisuPrivacySandbox Preview	TiramisuPrivacySandbox	3	Not installed
<input type="checkbox"/>	Android API 33	33	1	Not installed
<input type="checkbox"/>	Android API 32	32	1	Partially installed
<input type="checkbox"/>	Android 12.0 (S)	31	1	Partially installed
<input type="checkbox"/>	Android 11.0 (R)	30	3	Partially installed
<input type="checkbox"/>	Android 10.0 (Q)	29	5	Not installed
<input checked="" type="checkbox"/>	Android 9.0 (Pie)	28	6	Installed
<input type="checkbox"/>	Android 8.1 (Oreo)	27	3	Not installed
<input checked="" type="checkbox"/>	Android 8.0 (Oreo)	26	2	Installed
<input type="checkbox"/>	Android 7.1.1 (Nougat)	25	3	Not installed
<input type="checkbox"/>	Android 7.0 (Nougat)	24	2	Not installed
<input type="checkbox"/>	Android 6.0 (Marshmallow)	23	3	Not installed
<input type="checkbox"/>	Android 5.1 (Lollipop)	22	2	Not installed
<input type="checkbox"/>	Android 5.0 (Lollipop)	21	2	Not installed
<input type="checkbox"/>	Android 4.4W (KitKat Wear)	20	2	Not installed
<input type="checkbox"/>	Android 4.4 (KitKat)	19	4	Not installed
<input type="checkbox"/>	Android 4.3 (Jelly Bean)	18	3	Not installed
<input type="checkbox"/>	Android 4.2 (Jelly Bean)	17	3	Not installed

Para el proyecto se utilizaron los SDK, correspondientes a la imagen. Estos pueden ser modificados a discreción y consideración de la persona. Esto dependiendo de la versión de Android que se desee usar.

3.1.5.2. Device manager



El proyecto puede ser desplegado en dispositivos virtuales o físicos. En caso de tener un dispositivo virtual se deberá crear dicho dispositivo según las reglas del software.

En caso de tener un dispositivo físico sobre el cual desplegar el proyecto se deberá tener en cuenta la herramienta ADB y activar las opciones de desarrollador dentro del dispositivo.

3.1.6. Ionic Framework

Este complemento se instala en el punto 3.1.4. Este framework es el utilizado para la realización del proyecto para diferentes plataformas. En caso de requerir arrancar el proyecto para navegadores se deberá utilizar el comando: `ionic serve`; en caso de que se busque desplegar en la plataforma Android se utilizará el comando: `ionic capacitor build android`.

3.1.7. Capacitor

Este proyecto utiliza capacitor para la integración de las plataformas móviles. Sobre esta herramienta se despliegan los diferentes plugins que permitirán las interacciones con los dispositivos móviles. Esto está integrado en el proyecto y se instalar mediante el punto 3.1.4

4. REQUERIMIENTOS DEL SISTEMA

Requisitos del sistema para Android Studio

A continuación, se describen los requisitos del sistema para Android Studio en Windows.

- Microsoft® Windows® 8/10/11 de 64 bits
- Arquitectura de CPU x86_64; procesador Intel Core de segunda generación o posterior, o CPU AMD compatible con un [hipervisor de Windows](#)
- 8 GB de RAM o más
- 8 GB de espacio disponible en el disco como mínimo (IDE + SDK de Android + Android Emulator)
- Resolución de pantalla mínima de 1280 × 800

Dado que el software con más requisitos es Android Studio es el que tiene requisitos más definidos se tomará como base para las recomendaciones sobre el sistema sobre el cual se utilizará. La imagen anterior desvela los requisitos de sistema pedidos por este programa para su funcionamiento. Estos y más requerimientos o consejos se encuentran en: <https://developer.android.com/codelabs/basic-android-kotlin-compose-install-android-studio?hl=es-419#1>

5. BIBLIOGRAFÍA

Tully, C., & Alfaraz, C. (2017). Youth and mobility: The lifestyle of the new generation as an indicator of a multi-local everyday life. *Applied Mobilities*, 2(2), 182-198.

Runkler, T. A. (2020). *Data analytics*. Springer Fachmedien Wiesbaden.

Pareek, D. (2006). Business Intelligence for telecommunications. Auerbach Publications.

Tilkov, S., & Vinoski, S. (2010). Node. js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80-83.

Chaudhary, P. (2018). Ionic Framework. *International Research Journal of Engineering and Technology*, 5(05), 3181-3185.