



INSTITUTO TECNOLÓGICO SUPERIOR  
**SUDAMERICANO**  
QUITO - ECUADOR

ESCUELA DE  
DESARROLLO DE SOFTWARE

**PROYECTO DE TITULACIÓN**

DESARROLLO E IMPLEMENTACIÓN DE UNA RED SOCIAL PARA EL INSTITUTO  
SUPERIOR TECNOLÓGICO SUDAMERICANO QUITO

AUTOR: MOLINA ROJAS SAMIR ANTONIO

TUTOR: MSc. VILLASIS FABRIZIO

AGOSTO 2024  
QUITO – ECUADOR

## **AUTORÍA**

Yo, Samir Antonio Molina Rojas portador de la cédula de ciudadanía No. 1726369653, declaro bajo juramento que el trabajo aquí descrito, es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado e investigado en base a las referencias bibliográficas que se incluyen en este documento. Esta investigación no contiene plagio alguno y es resultado de un trabajo serio desarrollado en su totalidad por mi persona.

---

**Samir Antonio Molina Rojas**

## CERTIFICACIÓN

Una vez que se ha culminado la elaboración del proyecto de titulación cuyo tema es: “Desarrollo e implementación de una red social para el Instituto Superior Tecnológico Sudamericano Quito”, certifico que el mismo se encuentra habilitado para su defensa pública.

---

**MSc. Fabrizio Villasis Chiriboga**  
**Coordinador de la Escuela de**  
**Desarrollo de Software**  
**Instituto Superior Tecnológico Sudamericano Quito**

## CERTIFICACIÓN

Por medio del presente certifico que el señor Samir Antonio Molina Rojas ha realizado y concluido su trabajo de titulación, cuyo tema es: “Desarrollo e implementación de una red social para el Instituto Superior Tecnológico Sudamericano Quito”, para obtener el título de Tecnólogo en Desarrollo de Software, bajo mi tutoría.

---

**MSc. Fabrizio Vicente Villasis Chiriboga**  
**Director del Proyecto de Titulación**

## AGRADECIMIENTOS

Quiero expresar mi más profundo agradecimiento al Instituto Superior Tecnológico Sudamericano y a todas las personas que han sido parte de esta extraordinaria travesía de crecimiento personal, en especial aquellos que me han brindado su invaluable orientación y han hecho posible la culminación de esta tesis.

Mi profunda gratitud se dirige al MSc. Fabrizio Villasis, mi tutor, cuya orientación precisa, sabios consejos, paciencia inquebrantable y apoyo constante han sido fundamentales en cada paso de este desafiante camino. Su presencia ha sido una luz en los momentos más oscuros, brindándome dirección y motivación para superar cada obstáculo.

También deseo expresar mi profunda gratitud a mis padres, Wilson Homero Molina Tejada y Elizabeth del Rosario Rojas Andrade, quienes durante veinte años han sido mi más sólido apoyo. Con su inquebrantable amor, sabiduría y esfuerzo incansable, han forjado un camino para mí, garantizando que no me falte nada en la vida. Su capacidad para superar las adversidades y apreciar las alegrías de la vida me ha inspirado a seguir adelante y a aspirar a mayores logros.

Asimismo, agradezco de corazón a mis entrañables amigos, en especial a Sebastián Cevallos y Camila Pancho, quienes han sido mi mejor compañía y resguardo durante muchos años. Su constante apoyo, escucha comprensiva y palabras de aliento han sido un bálsamo en los momentos difíciles, impulsándome siempre a seguir adelante en mi camino de crecimiento y desarrollo personal.

Finalmente agradezco infinitamente a un ser de paz como lo fue mi perrito Thor, que durante mucho tiempo estuvo a mi lado viéndome crecer y dándome su apoyo de la forma más pura y sincera que esos seres brindan, volviéndose mi nuevo ángel que me seguirá cuidando y apoyando desde lo más alto.

## DEDICATORIA

Este proyecto de titulación está dedicado en primer lugar a mis padres, cuyo incansable esfuerzo y dedicación me han brindado las oportunidades necesarias para forjar mi propio futuro. Después de años de perseverancia y superación de adversidades, han sido testigos de mi graduación del colegio y, finalmente, de la culminación de este proyecto de titulación que marca el inicio de mi carrera profesional. A ellos les debo un profundo agradecimiento y reconocimiento por su constante apoyo y sacrificio.

Asimismo, dedico este proyecto a mí mismo, como resultado de mi entrega total y dedicación para llevarlo a cabo de la mejor manera posible. Mi compromiso fue dar lo mejor de mí para entregar un trabajo de calidad y sentirme orgulloso de mis logros.

También quiero dedicar este proyecto a mi tutor y amigo, MSc. Fabrizio Villasis, cuya guía experta y apoyo incondicional fueron fundamentales durante los cinco semestres de mi formación en el Instituto Superior Tecnológico Sudamericano. Sin su paciencia, sabios consejos y dedicación, este logro no habría sido posible. Su intervención oportuna y persistencia fueron cruciales para proporcionarme las herramientas necesarias para llevar a cabo este proyecto con éxito.

Por último, dedico este trabajo al Instituto Superior Tecnológico Sudamericano, un lugar que ha sido fundamental en mi formación académica y personal. Agradezco a todos los compañeros, profesores y personal administrativo que han contribuido a mi desarrollo durante este tiempo. Me llevo de este lugar no solo conocimientos académicos, sino también grandes amistades que valoro enormemente.

## RESUMEN

El proyecto consiste en el desarrollo de una aplicación móvil con temática de red social académica para combatir la problemática de la poca comunicación e interacción entre los miembros del Instituto Superior Tecnológico Sudamericano, esta app tiene por objetivo principal el proporcionar una plataforma digital donde los usuarios puedan interactuar, compartir información personal o institucional, y establecer vínculos significativos tanto en lo académico como en lo personal.

Para la creación del aplicativo se optó por usar herramientas como Python, FlutterFlow, Firebase y OneSignal, las cuales proporcionan un amplio catálogo de posibilidades a la hora de programar una aplicación móvil.

Se siguió una metodología de investigación ágil y eficaz para iniciar con el desarrollo del proyecto, con continuas pruebas para garantizar la funcionalidad y la experiencia que el usuario final. Asegurando una app intuitiva y fácil de usar para los usuarios finales.

Los resultados principales de esta aplicación fueron el desarrollo de perfiles de usuario, el buscador y seguimiento de distintos perfiles, sistema de publicaciones, así como tres tipos de chats dedicados a la conversación 1 a 1 y grupales de forma general y grupales de forma simplemente institucional separadas por carreras, niveles y secciones.

En conclusión, el proyecto fue un éxito al desarrollar una red social que cumplió con el objetivo general del proyecto de esta manera completando cada uno de los objetivos específicos que la aplicación debía garantizar y sustentar.

Es recámenle el seguir desarrollando nuevas actualizaciones para las funciones ya existentes e incorporar nuevas en base a las futuras necesidades de la institución.

Palabras clave: FlutterFlow, Firebase, OneSignal, Python, Red Social, aplicación, perfiles de usuario.

## ABSTRACT

The project consists of the development of a mobile application with an academic social network theme to combat the problem of poor communication and interaction between members of the Instituto Superior Tecnológico Sudamericano. The main objective of this app is to provide a digital platform where users can interact, share personal or institutional information, and establish meaningful links both academically and personally.

For the creation of the application, it was decided to use tools such as Python, FlutterFlow, Firebase and OneSignal, which provide a wide range of possibilities when programming a mobile application.

An agile and effective research methodology was followed to begin the development of the project, with continuous testing to ensure the functionality and experience of the end user. Ensuring an intuitive and easy-to-use app for end users.

The main results of this application were the development of user profiles, the search engine and monitoring of different profiles, a publication system, as well as three types of chats dedicated to 1-on-1 conversations and general group conversations and simply institutional group conversations separated by careers, levels and sections.

In conclusion, the project was a success in developing a social network that met the general objective of the project, thus completing each of the specific objectives that the application had to guarantee and support.

It is recommended to continue developing new updates for the existing functions and to incorporate new ones based on the future needs of the institution.

Keywords: FlutterFlow, Firebase, OneSignal, Python, Social Network, application, user profiles.

## ÍNDICE

1.	Introducción	1
2.	Justificación	2
3.	Antecedentes	3
4.	Objetivos	4
4.1.	Objetivo General	4
4.2.	Objetivos Específicos	4
5.	Marco Teórico	5
5.1	Fundamentos teóricos y herramientas esenciales para el diseño e implementación de la red social académica.	5
5.1.1	Selección de base de datos	5
5.1.1.1	MySQL base de datos relacional.	6
5.1.1.2	PostgreSQL base de datos relacional.	6
5.1.1.3	Firebase base de datos NOSQL o no relacional.	7
5.1.1.4	MongoDB base de datos NOSQL o no relacional.	7
5.1.2	Selección de creador de aplicación.	7
5.1.2.1	Java	8
5.1.2.2	Python	8
5.1.2.3	Dart	8
5.1.2.4	Bubble	9
5.1.2.5	FlutterFlow	9
5.2	Selección de las herramientas para la implementación de la red social	10
5.2.1	Selección de FlutterFlow	10

5.2.2	Selección de Firebase	11
6	Desarrollo del Proyecto de Titulación	13
6.1	Análisis de redes sociales populares entre la comunidad	13
6.1.1	Facebook	14
6.1.2	Instagram	14
6.1.3	Tinder	15
6.2	Diseño de la red social que se ajuste a la realidad del Instituto Superior Tecnológico Sudamericano de Quito y de instituciones educativas superiores en general.	18
6.2.1	Entorno cerrado y seguridad	18
6.2.2	Escalabilidad y Base de Datos	18
6.2.3	Publicaciones y comunicación masiva:	19
6.2.4	Perfiles de Usuario:	19
6.2.5	Interacciones y seguimientos:	19
6.2.6	Búsqueda de Usuarios:	20
6.2.7	Sistema de Chat:	20
6.3	Diagrama de comportamiento, estructural y mapa conceptual	21
6.4	Creación de un proyecto dentro de Firebase.	24
6.5	Creación de un proyecto en FlutterFlow	26
6.6	Método de registro de usuarios	29
6.6.1	Creación del archivo Python para registrar usuarios.	30
6.7	Creación archivo ejecutable para creación de perfiles en un solo paso.	34
6.8	Ciclos y funciones de las pantallas de la aplicación	36
6.8.1	Inicio de sesión y/o Registro dentro de la aplicación	36

6.8.2	Perfil del usuario autenticado	39
6.8.3	Pantalla de edición de perfil	40
6.8.4	Pantalla de información del usuario autenticado	41
6.8.4.1	Colección de “users” de Firebase	42
6.8.5	Página para subir publicaciones a la red social académica	43
6.8.5.1	Colección de “publicaciones” de Firebase	44
6.8.6	Pantalla de inicio o home	45
6.8.6.1	Función dar me gusta a una publicación	47
6.8.6.2	Función comentar a una publicación	49
6.8.6.3	Función compartir una publicación	52
6.8.7	Pantalla Buscador de perfiles	54
6.8.8	Pantalla de perfil de otro usuario	56
6.8.8.1	Conteo de seguidores y seguidos	58
6.8.9	Pantallas de chat 1 a 1	61
6.8.9.1	Colecciones dentro de Firebase para los chats 1 a 1	61
6.8.9.2	Pantallas de inicio de chat 1 a 1	63
6.8.9.3	Pantalla de espacio de chat	64
6.8.9.4	Pantalla de listado de chats 1 a 1	67
6.8.10	Pantallas de chat grupal general	67
6.8.10.1	Colecciones dentro de Firebase para los chats grupales generales	67
6.8.10.2	Pantalla de listado de grupos generales.	69
6.8.10.3	Pantalla para agregar miembros al chat grupal general	71
6.8.10.4	Pantalla espacio del chat grupal general	74
6.8.10.5	Menú de configuración de chat grupal general	78
6.8.10.6	Agregar administrador al grupo	79

6.8.11	Pantallas de chat grupal institucional _____	80
6.8.11.1	Colecciones dentro de Firebase para los chats grupales institucionales. __	81
6.8.11.2	Pantalla de creación de grupo institucional _____	82
6.8.11.3	Pantalla de listado de chats grupales institucionales _____	83
6.8.11.4	Pantalla de espacio de chat institucional _____	84
6.8.11.5	Menú de configuración de grupos institucionales _____	86
6.8.12	Realizar test de la aplicación para detectar posibles errores. _____	90
6.9	Sistema de envío de notificaciones “push” e “in app” _____	91
6.9.1	Implementación de la API _____	91
6.9.2	Conectar API dentro de FlutterFlow _____	96
6.9.3	Funcionalidad de las notificaciones _____	97
7	Publicación de la aplicación dentro de Google Play _____	98
8	Conclusiones y Recomendaciones _____	100
8.1	Conclusiones _____	100
8.2	Recomendaciones _____	102
9	Referencias _____	105
	ANEXOS _____	108

## ÍNDICE DE FIGURAS

Figura 1. Logo Flutterflow.	11
Figura 2. Logo Firebase.	11
Figura 3. Usuarios mensuales en las redes sociales populares.	15
Figura 4. Valores calculados anualmente de los usuarios de las redes sociales populares.	16
Figura 5. Gráfico de barras valores anuales de los usuarios de las redes sociales populares.	16
Figura 6. Cantidad de cuentas de negocio en Facebook.	17
Figura 7. Gráfico cuentas de negocios.	17
Figura 8. Diagrama de comportamiento de la Red Social Académica.	21
Figura 9. Diagrama estructural de la Red Social Académica.	22
Figura 10. Mapa conceptual Red Social Académica.	23
Figura 11. mapa conceptual de las funciones permitidas a cada tipo de usuario.	24
Figura 12. Creación proyecto en Firebase.	25
Figura 13. Agregar a FlutterFlow como editor de proyecto.	25
Figura 14. Autenticación con correo y contraseña.	26
Figura 15. Conexión de firebase con FlutterFlow.	27
Figura 16. Perfiles creados en el archivo Excel.	30
Figura 17. Python instalado.	30
Figura 18. Credenciales del proyecto Firebase en formato JSON.	31
Figura 19. Instalación librerías Python.	31
Figura 20. Llamar librerías e inicializar Firebase con credenciales.	32
Figura 21. Llamar al archivo Excel con la librería openpyxl.	32
Figura 22. Leer el archivo Excel empezando por la segunda línea.	33
Figura 23. Verificar si el campo email no está vacío y crear usuario.	33
Figura 24. Guardar datos adicionales del usuario.	34

Figura 25. Alertas y proceso culminado.	34
Figura 26. Ejecutables proceso registro.	35
Figura 27. Firebase Autenticación.	35
Figura 28. Información de perfil de usuario exportada desde Excel.	36
Figura 29. Inicio de sesión.	37
Figura 30. PIN para la sección de registro.	38
Figura 31. Sección de registros dentro de la aplicación.	38
Figura 32. Pantalla del perfil de usuario.	39
Figura 33 Pantalla de edición de perfil.	41
Figura 34 Pantalla de información del usuario.	42
Figura 35. Colección de usuarios dentro de FlutterFlow y Firebase.	43
Figura 36. Pantalla Publicaciones.	44
Figura 37. Colección de publicaciones dentro de FlutterFlow y Firebase.	45
Figura 38. Pantalla de inicio o home.	46
Figura 39. Estados icono me gusta.	47
Figura 40 Consulta listado de usuarios que dan me gusta.	48
Figura 41 Pantalla de usuarios que les gusto la publicación.	48
Figura 42. Sección de comentarios.	49
Figura 43. Variables de estado.	50
Figura 44. Sección de comentarios funcional.	51
Figura 45. Colección de comentarios dentro de FlutterFlow y Firebase.	52
Figura 46. Publicación repostada.	53
Figura 47. Variable del buscador.	54
Figura 48. Construcción pantalla buscador.	55
Figura 49. Pantalla del Buscador de perfiles.	56

Figura 50. Funcionalidad de bloqueo de usuario. _____	57
Figura 51. Pantalla de perfil de usuario diferente. _____	58
Figura 52. Pantalla de seguidores o seguidos. _____	59
Figura 53. Vista de seguidores y seguidos perfiles de usuario diferente. _____	61
Figura 54. Vista de seguidores y seguidos perfil usuario autenticado. _____	61
Figura 55. Colección de chats 1 a 1 dentro de FlutterFlow y Firebase. _____	62
Figura 56. Colección de mensajes de chats 1 a 1 dentro de FlutterFlow y Firebase. _____	63
Figura 57. Pantalla de chat inicial. _____	64
Figura 58. Configuración burbujas de chat 1 a 1. _____	65
Figura 59. Espacio de chat 1 a 1. _____	66
Figura 60. Condiciones en el listado de chat 1 a 1. _____	67
Figura 61. Colección de chats grupales general dentro de FlutterFlow y Firebase. _____	68
Figura 62. Colección de mensajes de chats grupales general dentro de FlutterFlow y Firebase. _____	69
Figura 63. Pantalla de listado de chats grupales generales. _____	71
Figura 64. Tipos de búsqueda en los grupos generales. _____	72
Figura 65. Pantalla Agregar miembros a grupos generales. _____	74
Figura 66. Componente inicial espacio de chat. _____	75
Figura 67. Componente de bienvenida al grupo. _____	75
Figura 68. Componente burbujas de chat grupales. _____	76
Figura 69. Componente final pantalla espacio de chat. _____	77
Figura 70. Pantalla de espacio de chat. _____	78
Figura 71. Menú de configuración grupos generales. _____	79
Figura 72. Agregar administradores al grupo. _____	80
Figura 73. Colección de chats grupales institucionales dentro de FlutterFlow y Firebase. _____	81

Figura 74. Colección de mensajes de chats grupales institucionales dentro de FlutterFlow y Firebase. _____	82
Figura 75. Pantalla de creación de grupo institucional. _____	83
Figura 76. Pantalla de listado de chats grupales institucionales. _____	84
Figura 77. Pantalla de espacio de chat institucional. _____	85
Figura 78. Cambio de nombre grupo institucional. _____	86
Figura 79. Zona de miembros chat institucionales. _____	87
Figura 80. Miembros que coinciden con la selección. _____	87
Figura 81. Selecciones carrera, nivel, sección. _____	89
Figura 82. Actualización de datos de los grupos institucionales. _____	90
Figura 83. Planes OneSignal. _____	92
Figura 84. Formulario inicial en OneSignal. _____	92
Figura 85. Nombrar aplicación de OneSignal. _____	93
Figura 86. Configuración OneSignal. _____	93
Figura 87. Elección de canal de mensajería. _____	94
Figura 88. Descarga de nueva llave privada. _____	95
Figura 89. Importar clave privada. _____	95
Figura 90. Selección SDK de destino. _____	96
Figura 91. Agregar API al proyecto. _____	96
Figura 92. Notificación exitosa. _____	97
Figura 93. App subida dentro de Google Play Console para su revisión. _____	99

## 1. Introducción

El proyecto se enfoca en abordar una problemática dentro del Instituto Superior Tecnológico Sudamericano en Quito, que consiste en el distanciamiento significativo entre las diversas carreras que ofrece la institución. Este problema ha generado un entorno académico fragmentado donde la comunicación, interacción y colaboración entre los miembros se ve altamente comprometida.

El Instituto Superior Tecnológico Sudamericano se ve en la necesidad de implementar una plataforma dedicada a fortalecer los vínculos dentro de la institución, mejorando la comunicación entre estudiantes, personal administrativo y maestros. Debido a estas dificultades, se opta por la creación de una herramienta que fomente la unificación y la cooperación entre todos los miembros de la comunidad educativa.

El objetivo principal de esta plataforma es incrementar la interacción entre los usuarios, facilitando la formación de relaciones sólidas y colaborativas entre compañeros y profesores de diferentes áreas de estudio. A través de esta red social, se busca crear un espacio virtual donde los miembros de la institución puedan compartir información, discutir ideas, colaborar en proyectos y establecer conexiones que trasciendan las fronteras de sus respectivas carreras.

En resumen, la creación de una red social académica para el Instituto Superior Tecnológico Sudamericano en Quito representa una respuesta innovadora y necesaria ante la problemática del distanciamiento entre las carreras dentro de la institución. Esta iniciativa no solo busca mejorar la comunicación y la interacción entre los miembros de la comunidad educativa, sino que también aspira a promover una cultura de colaboración y aprendizaje mutuo que beneficie a todos los involucrados.

## 2. Justificación

La concepción de la Red Social parte de la idea de facilitar la conexión y la integración entre los miembros de la Institución además de compartir información relevante dirigida a la comunidad institucional, tomando en consideración la diversidad de edificios que conforman su infraestructura. En un entorno donde las plataformas digitales adquieren una creciente relevancia en la construcción de relaciones personales, esta iniciativa busca aprovechar dicha tendencia para potenciar los lazos entre los individuos que comparten el mismo ámbito educativo.

El impacto de la tecnología en la comunidad institucional es innegable puesto que podemos observar a los estudiantes usar sus teléfonos en los pasillos de la institución, generando un escenario propicio para la concepción de una red social diseñada para robustecer la interacción en este entorno. La interconexión digital está impregnando en diversos aspectos de la vida moderna y educativa, lo que subraya la necesidad de explorar nuevas vías para promover la comunicación y la conexión entre los miembros de la institución, además es una buena forma de promocionar los eventos de la institución u otra información importante.

El propósito de esta red social es abordar la carencia de plataformas específicas que estimulen la conexión entre individuos pertenecientes a la misma institución. Esta propuesta apunta a colmar un vacío en la interacción digital dentro de la comunidad. El objetivo esencial es permitir que las personas que comparten un mismo espacio físico puedan encontrarse y conectar en línea, enriqueciendo así su experiencia institucional. Otro de los objetivos fundamentales de esta plataforma es promover y facilitar la difusión de información importante que el instituto necesite socializar, fortaleciendo de esta manera la comunicación interna y asegurando que todos los miembros estén al tanto de los acontecimientos relevantes.

### **3. Antecedentes**

Actualmente, la comunicación y la interacción entre los miembros de la comunidad institucional del Instituto Superior Tecnológico Sudamericano de Quito se han visto limitadas por la ausencia de una plataforma digital dedicada que facilite estas interacciones. Esta realidad plantea dificultades para promover la convivencia entre estudiantes, docentes y personal administrativo, así como para la difusión efectiva de información relevante y la participación en eventos institucionales. Ante esta situación, se hace necesario desarrollar una red social que dé respuesta a estas necesidades específicas y promueva una mayor integración y colaboración dentro de la comunidad educativa.

Además, esta iniciativa de desarrollar una red social interna representa una oportunidad única para romper barreras existentes entre los miembros de la comunidad educativa del Instituto Superior Tecnológico Sudamericano Quito. Al proporcionar una forma rápida y accesible de compartir información relevante, además de facilitar las interacciones entre estudiantes, profesores y personal administrativo, crea un entorno propicio para el intercambio de ideas, el aprendizaje colaborativo y la creación de redes de apoyo. Esta idea innovadora no sólo promueve una mayor integración y colaboración dentro de la institución, sino que también contribuye significativamente a mejorar las experiencias educativas de todos sus miembros. Al permitir que la comunidad educativa se mantenga conectada e informada de los eventos, proyectos y actividades institucionales, la red social se convierte en un motor de convivencia, participación y enriquecimiento mutuo, fortaleciendo así el tejido social y académico de la institución.

## **4. Objetivos**

### **4.1. Objetivo General**

Desarrollar e implementar una red social para el Instituto Superior Tecnológico Sudamericano de Quito, para promover las relaciones entre los estudiantes y socializar información relevante a toda la comunidad institucional.

### **4.2. Objetivos Específicos**

- Desarrollar los diversos conceptos, definiciones, tecnologías y herramientas necesarias para el diseño e implementación del proyecto a manera de marco teórico.
- Definir las tecnologías y herramientas clave que se utilizarán para la implementación de la red social.
- Analizar detalladamente las características y funciones de plataformas sociales populares como Instagram, Facebook y Tinder.
- Diseño de la red social que se ajuste a la realidad del Instituto Superior Tecnológico Sudamericano de Quito y de Instituciones Educativas Superiores en general.
- Desarrollar la aplicación con las características y funcionalidades definidas.
- Realizar test de la aplicación para detectar posibles errores.
- Publicación de la red social académica en la tienda de Android.

## **5. Marco Teórico**

El objetivo de este proyecto se basa en la construcción de una red social, que satisfaga las necesidades de comunicación e interacción entre miembros de la comunidad institucional del Instituto Superior Tecnológico Sudamericano Quito, para la construcción de dicha aplicación se debe desarrollar conceptos, definiciones y barajar diversas herramientas que ayuden con este proyecto.

### **5.1 Fundamentos teóricos y herramientas esenciales para el diseño e implementación de la red social académica.**

Para el desarrollo de una red social funcional es crucial comprender y aplicar una serie de fundamentos teóricos y herramientas tecnológicas las cuales brinden soluciones efectivas a la hora de reducir tiempo y recursos para la creación de una aplicación.

Para el proyecto, fue esencial seleccionar cuidadosamente las herramientas adecuadas para el proceso de desarrollo. Se concluyó que era necesario utilizar una base de datos que sirviera como almacenamiento de información del aplicativo. Además, el uso de editores o creadores de código resultó fundamental para desarrollar la parte visual y funcional de la red social. Finalmente, para realizar pruebas de funcionalidad y diseño, se implementaron emuladores de celular y dispositivos físicos.

#### **5.1.1 Selección de base de datos**

Una base de datos es un conjunto de información almacenada de forma sistemática, lo que permite la recuperación, el análisis y la transmisión de los datos. Existen diferentes tipos de bases de datos, siendo las más comunes las relacionales y las no relacionales (NoSQL) (Equipo editorial, Etecé., 2023).

Las bases de datos relacionales organizan la información en filas y columnas, creando tablas que se relacionan entre sí mediante claves primarias y secundarias. Este tipo de base de datos es ideal para combinar y resumir grandes cantidades de datos. Entre las opciones destacadas se encuentran MySQL y PostgreSQL (IBM, s.f.).

Por otro lado, las bases de datos no relacionales, también conocidas como NoSQL, no utilizan tablas ni filas. En su lugar, almacenan la información en forma de documentos, similar a hojas de papel. Este tipo de base de datos es adecuado para manejar grandes volúmenes de información, ofreciendo escalabilidad, flexibilidad y velocidad mejorada. Las opciones principales en este ámbito son Firebase y MongoDB (Mosca, 2023).

#### **5.1.1.1 MySQL base de datos relacional.**

MySQL es un sistema de base de datos relacional de código abierto que almacena información en forma de filas y columnas. Se basa en una arquitectura cliente-servidor, permitiendo una comunicación eficiente entre clientes y servidores para consultas, modificaciones y gestión de datos. Permite configurar vistas personalizadas y utilizar procedimientos almacenados para incrementar la eficacia en la gestión de bases de datos. Aunque es personalizable, su naturaleza relacional limita su escalabilidad comparada con las bases de datos NoSQL (Robledano, 2019).

#### **5.1.1.2 PostgreSQL base de datos relacional.**

PostgreSQL es un gestor de base de datos relacional de código abierto, gratuito y disponible para Linux, Unix y Windows. Es la segunda base de datos más utilizada y se distingue por su gran escalabilidad y fiabilidad. Ofrece soporte completo para claves foráneas, uniones, vistas,

disparadores y procedimientos almacenados. Soporta la mayoría de los tipos de datos de SQL y el almacenamiento de objetos binarios grandes (Kinsta, 2019).

#### **5.1.1.3 Firebase base de datos NOSQL o no relacional.**

Firebase es una plataforma desarrollada por Google que permite a los desarrolladores crear aplicaciones escalables y flexibles, aumentando la cantidad de usuarios sin afectar la calidad de la aplicación. Una de sus principales herramientas es Cloud Storage, que permite almacenar y gestionar medios generados por los usuarios, como imágenes, documentos y videos (López Mora, 2020).

#### **5.1.1.4 MongoDB base de datos NOSQL o no relacional.**

MongoDB es una base de datos NoSQL orientada a documentos, destacada por su flexibilidad y rendimiento. A diferencia de las bases de datos relacionales tradicionales, no utiliza tablas y columnas, sino que almacena datos en colecciones y documentos, que son pares clave/valor. Esta estructura permite manejar grandes volúmenes de datos de manera eficiente. MongoDB es una herramienta imprescindible en la ingeniería de datos, ideal para aplicaciones que requieren escalabilidad y gestión de datos masivos (DataScientest, 2022).

### **5.1.2 Selección de creador de aplicación.**

Para la creación de aplicaciones existen diversas herramientas que facilitan este proceso. Se pueden utilizar herramientas que se basan completamente en código, empleando lenguajes de programación como Java, Python o Dart, junto con algún editor de código. Alternativamente, se pueden emplear herramientas de bajo código, como FlutterFlow o Bubble, que permiten

desarrollar aplicaciones de manera más eficiente y con menos necesidad de programación manual.

#### **5.1.2.1 Java**

Java es un lenguaje de programación empleado para desarrollar software compatible con una amplia variedad de sistemas operativos. Su característica distintiva es la capacidad de ser compilado e interpretado simultáneamente, lo que simplifica el proceso al convertir automáticamente el código en instrucciones de máquina. Java es ideal para construir aplicaciones complejas con múltiples interfaces, ya que permite la compilación en diferentes dispositivos y sistemas operativos (Coppola, 2023).

#### **5.1.2.2 Python**

Python es un lenguaje de programación orientado a objetos, caracterizado por su fácil interpretación y una sintaxis similar al inglés, lo que facilita su legibilidad. Este lenguaje permite el desarrollo de aplicaciones móviles, web y software para computadoras, lo que lo convierte en una herramienta versátil para diversas necesidades de desarrollo, tales como la operación de sistemas, la creación de juegos y el desarrollo de aplicaciones móviles (Londoño, 2023).

#### **5.1.2.3 Dart**

Dart es un lenguaje de programación desarrollado por Google, diseñado como una alternativa a JavaScript para los navegadores web. Como estándar Ecma, Dart se caracteriza por su sintaxis similar a C, lo que facilita su aprendizaje. Es un lenguaje orientado a objetos que soporta herencia y programación genérica, y puede ser compilado a JavaScript mediante el

compilador Dart2js. Dart es compatible con variables, operadores, enunciados condicionales, bucles, funciones, clases, objetos y listas (Equipo editorial de IONOS, 2020).

#### **5.1.2.4 Bubble**

Bubble.io es una herramienta que permite crear aplicaciones, CRM y redes sociales sin necesidad de programar. Es especialmente útil para startups y empresas que buscan validar ideas rápidamente y a bajo costo. Aunque su interfaz es sencilla, requiere formación para su uso efectivo. Bubble.io ofrece diseño responsive y la opción de adquirir plantillas ya creadas con un precio desde los 100 dólares o euros, facilitando el desarrollo de software sin conocimientos de código (Lima, 2022).

#### **5.1.2.5 FlutterFlow**

Flutterflow es una herramienta no-code para el desarrollo de aplicaciones móviles y web que permite crear interfaces de usuario y funcionalidades interactivas sin necesidad de escribir código. Utiliza Flutter, el framework de código abierto de Google, permitiendo la creación de aplicaciones nativas para móviles, web y escritorio. La plataforma ofrece la capacidad de exportar el código generado para mayor personalización, y facilita el trabajo colaborativo en tiempo real. Además, se integra con diversos servicios externos como bases de datos y APIs, proporcionando una solución robusta y versátil para proyectos de desarrollo de aplicaciones (Greaves & Mengistu, s.f.).

## **5.2 Selección de las herramientas para la implementación de la red social**

Para la implementación de la red social académica, se ha seleccionado FlutterFlow como herramienta principal para el desarrollo visual y funcional de la aplicación, y Firebase como base de datos no relacional. A continuación, se detallan las razones y ventajas de estas elecciones.

### **5.2.1 Selección de FlutterFlow**

FlutterFlow ha sido elegida para construir la interfaz visual y funcional de la aplicación debido a varias ventajas clave. A pesar de estar completamente en inglés, FlutterFlow proporciona una amplia gama de widgets que facilitan el proceso de desarrollo, haciéndolo más intuitivo y accesible. Comparado con otras plataformas como Bubble, que tiene un objetivo similar de crear aplicaciones, FlutterFlow se destaca por ser más completo y eficiente.

Una de las características más apreciadas de FlutterFlow es la capacidad de descargar el código fuente o el archivo APK de la aplicación una vez finalizada, sin la necesidad de seguir suscribiéndose a un plan profesional. Esto permite a los desarrolladores mantener el código y el control total sobre la aplicación sin incurrir en costos adicionales a largo plazo.

Además, FlutterFlow ofrece una mejor integración entre pantallas y una mayor personalización en la interfaz de usuario, lo cual es crucial para crear una experiencia de usuario coherente y fluida.

La siguiente figura muestra el logo de FlutterFlow:



**Figura 1. Logo Flutterflow.**  
**Fuente:** (FlutterFlow, s.f.)

### 5.2.2 Selección de Firebase

Firebase ha sido seleccionada como la base de datos principal debido a su capacidad de escalabilidad. A diferencia de las bases de datos relacionales, que pueden volverse lentas o colapsar bajo múltiples consultas simultáneas, Firebase maneja eficientemente grandes volúmenes de datos sin sacrificar rendimiento.

Otra razón importante para elegir Firebase es su integración fluida con FlutterFlow. Gracias a las opciones de generación de código y configuración de Firebase dentro de FlutterFlow, la conectividad entre ambos sistemas se logra de manera sencilla. Esto permite que la aplicación acceda a servicios esenciales como Cloud Storage y Firestore Database, facilitando el almacenamiento y la gestión de datos de usuario.

La siguiente figura muestra el logo de Firebase.



**Figura 2. Logo Firebase.**  
**Fuente:** (Firebase, s.f.)

Finalmente, en caso de descargar el código fuente, se utilizará el lenguaje de programación Dart, que es el lenguaje nativo de Flutter. Esta elección asegura que el desarrollo y la personalización del código se realicen de manera eficiente y coherente con las herramientas utilizadas.

En resumen, la combinación de FlutterFlow y Firebase proporciona una solución robusta y escalable para el desarrollo de la red social académica, optimizando tanto el proceso de creación como la funcionalidad y el rendimiento de la aplicación.

## **6 Desarrollo del Proyecto de Titulación**

### **6.1 Análisis de redes sociales populares entre la comunidad**

En la actualidad, las redes sociales son un medio cotidiano en la vida de la mayoría de las personas a nivel mundial, ya que estas plataformas han transformado drásticamente la manera en que nos comunicamos y conectamos en la sociedad.

Estas herramientas han logrado romper y derribar barreras geográficas, permitiendo interactuar con individuos de otros países o, sin ir más lejos, de la misma ciudad, independientemente de la distancia. Esto las convierte en medios atractivos para compartir información de manera rápida y eficaz, dado que, gracias a estas plataformas, tenemos acceso a una cantidad creciente de datos, historias, noticias y conocimientos que se revelan por millones cada día, manteniéndonos actualizados en todo momento sin depender exclusivamente de otros medios tradicionales.

Además, las redes sociales también funcionan como medios de expresión ciudadana, permitiendo la manifestación de opiniones, la participación en debates y la creación de nuevos movimientos de interés social. De esta manera, se participa activamente en áreas como el marketing y la promoción, ofreciendo una amplia variedad de formas para promocionar contenido, ya sea un producto o un servicio, facilitando la interacción con los clientes.

Dentro de este mundo de las redes sociales tenemos ejemplos muy notables en el mercado como por ejemplo Facebook, Instagram, o más recientemente Tinder entre muchas otras.

Principalmente el proyecto se basa en algunas funcionalidades de estas plataformas, aquí se analiza que funcionalidades son las más importantes.

### 6.1.1 Facebook

Facebook es una red social que nació por primera vez el 4 de febrero de 2004 por Mark Zuckerberg, Eduardo Saverin, Andrew McCollum, Dustin Moskovitz y Chris Hughes. Esta aplicación se lanzó originalmente con el nombre “TheFacebook” Posteriormente, en agosto de 2005, cambió su nombre a “Facebook” (Wikipedia, s.f.).

Las funcionalidades de Facebook más necesarias para el proyecto son:

- **Perfiles de usuarios:** Los usuarios pueden crear perfiles con fotos, e información personal de su interés.
- **Amigos y conexiones:** Permite a las personas conectarse con sus amigos enviando solicitudes de amistad.
- **Publicaciones:** Los usuarios pueden publicar fotos y pueden ser interactuadas por el resto de la comunidad.

### 6.1.2 Instagram

Es una plataforma que fue lanzada por primera vez en 2010 por los cofundadores Kevin Systrom y Mike Krieger, originalmente se llamaba Burbn, es una plataforma que permite compartir fotos y videos.

Algunas funciones interesantes de esta plataforma son:

- **Compartir fotos:** La aplicación permite subir fotografías en forma de publicación.
- **Seguir y ser seguido:** Los usuarios pueden seguir a otros usuarios sin necesidad de aceptar solicitudes.
- **Explorador:** Los usuarios pueden buscar usuarios de la plataforma.

### 6.1.3 Tinder

Es una aplicación de citas en línea, lanzada en 2012, la principal función de esta aplicación es conocer gente y conectar, esto con ayuda de perfiles de usuario que muestre fotos, biografías y algunos intereses en común.

Las plataformas mencionadas son utilizadas anualmente por millones de personas, lo que indica un uso significativo de las funciones que estas ofrecen. Facebook encabeza la lista de redes sociales más populares, seguida por YouTube, WhatsApp e Instagram. A pesar de haber perdido protagonismo en los últimos años, Facebook continúa siendo líder y modelo para muchas plataformas con objetivos similares (Statista, 2024).

Por otro lado, Tinder se distingue como una red social enfocada en generar relaciones sentimentales entre sus usuarios, lo que la hace menos utilizada en comparación con Facebook e Instagram. Estas últimas son plataformas más orientadas a compartir el día a día y relacionarse con amigos (Roast, 2024).

Los datos correspondientes se presentan en las siguientes figuras.



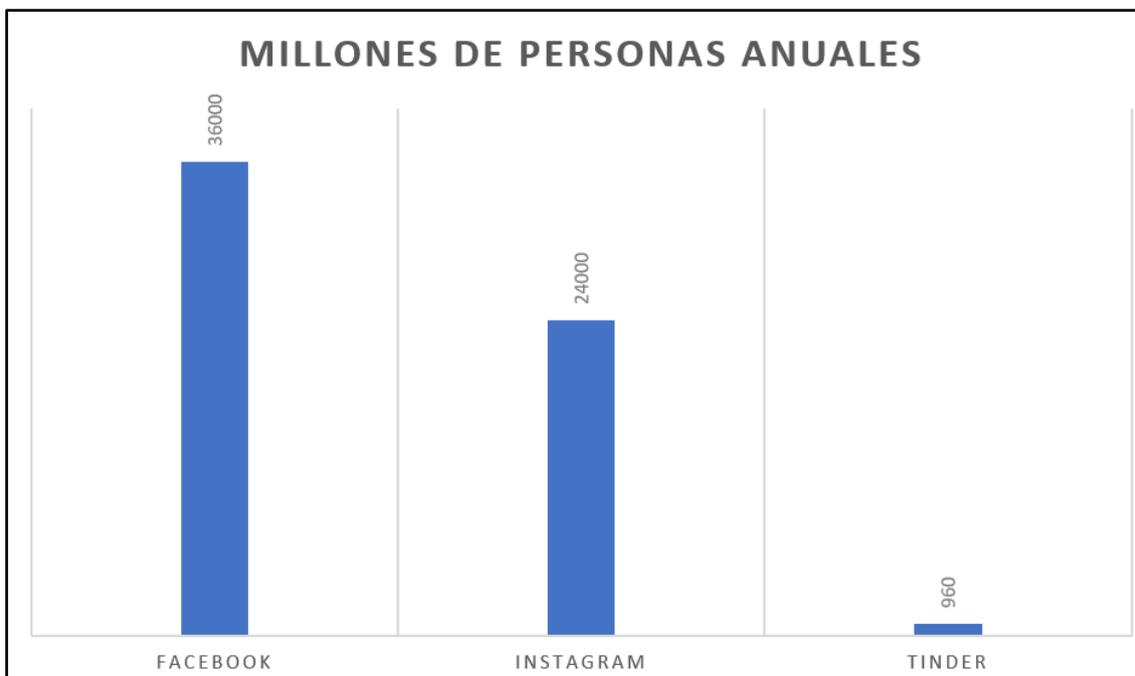
**Figura 3. Usuarios mensuales en las redes sociales populares.**

**Fuente:** (Statista, 2024)

Redes Sociales	Millones de Personas Mensuales	Millones de Personas Anuales
Facebook	3000	36000
Instagram	2000	24000
Tinder	80	960

**Figura 4. Valores calculados anualmente de los usuarios de las redes sociales populares.**

**Fuente: El Autor.**



**Figura 5. Gráfico de barras valores anuales de los usuarios de las redes sociales populares.**

**Fuente: El Autor.**

El gráfico adjunto proporciona una perspectiva clara sobre la popularidad y el uso de las redes sociales en nuestro entorno, destacando especialmente a Facebook. Esta plataforma, a pesar de perder protagonismo con el paso del tiempo, sigue siendo una de las más utilizadas. Se estima que los usuarios invierten aproximadamente 19 horas al mes en Facebook. Además, es la plataforma más empleada para la búsqueda de marcas y personas, siendo superada únicamente por YouTube, según datos del sitio web de Kolsquare (Kolsquare, 2024).

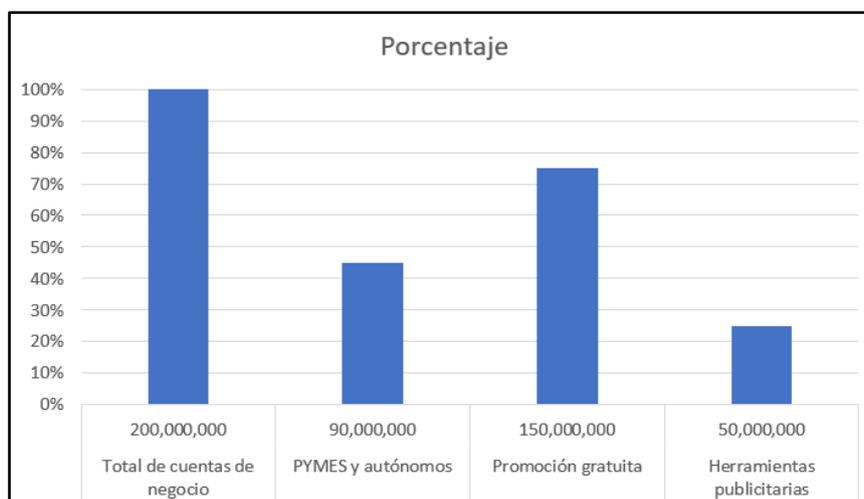
En Facebook, más de 200 millones de cuentas están dedicadas a negocios que buscan conectar con su audiencia. Este dato refleja la extensa utilización de las funcionalidades

gratuitas dentro de la plataforma, como la publicación de contenido y el uso del servicio de mensajería dentro de la aplicación. 90 millones de estas cuentas están dedicadas exclusivamente a PYMES (empresa mercantil, industrial que se conforma por un número pequeño de trabajadores) y autónomos. Aproximadamente el 75% de estas cuentas aprovechan la publicidad gratuita, mientras que solo el 25% optan por las opciones de pago (Pérez, 2024).

Esta distribución se ilustra en las siguientes figuras:

<b>Categoría</b>	<b>Cantidad de cuentas</b>	<b>Porcentaje</b>
Total de cuentas de negocio	200,000,000	100%
PYMES y autónomos	90,000,000	45%
Promoción gratuita	150,000,000	75%
Herramientas publicitarias	50,000,000	25%

**Figura 6. Cantidad de cuentas de negocio en Facebook.**  
Fuente: El Autor.



**Figura 7. Gráfico cuentas de negocios.**  
Fuente: El Autor.

## **6.2 Diseño de la red social que se ajuste a la realidad del Instituto Superior Tecnológico Sudamericano de Quito y de instituciones educativas superiores en general.**

En el diseño de una red social académica adaptada a la realidad del Instituto Superior Tecnológico Sudamericano de Quito, así como de otras instituciones educativas superiores, se plantea una solución que facilite la interacción y comunicación entre todos los miembros de la comunidad educativa. La intención es crear un entorno cerrado y seguro, en el que únicamente puedan participar personas vinculadas a la institución, incluyendo estudiantes, docentes y personal administrativo.

### **6.2.1 Entorno cerrado y seguridad**

Dado que la red social debe ser exclusiva para los miembros de la institución, se decidió implementar un sistema de cuentas pre-creadas. Estas cuentas serán generadas mediante un script en Python que tomará los datos de un archivo Excel y los ingresará automáticamente en el sistema de autenticación de Firebase. De esta forma, se garantiza que solo los individuos autorizados tengan acceso a la plataforma, fortaleciendo así la seguridad y privacidad de los datos personales.

### **6.2.2 Escalabilidad y Base de Datos**

Considerando la necesidad de escalabilidad, especialmente para integrar tanto el edificio matriz como las sucursales del instituto, se optó por utilizar Firebase como base de datos. Firebase no solo proporciona una gestión eficiente y segura de los datos, sino que también ofrece una escalabilidad adecuada para manejar un gran número de usuarios simultáneamente sin comprometer el rendimiento de la aplicación.

Una vez planteado este inicio, hay que plantear algunas de las funcionalidades clave que posea la aplicación, pues debe reunir varias de las ideas de otras apps del mercado y agregar algo diferente.

### **6.2.3 Publicaciones y comunicación masiva:**

Para facilitar la comunicación rápida y efectiva, se implementará una funcionalidad que permita realizar publicaciones que lleguen a toda la comunidad educativa. Esto elimina la necesidad de enviar comunicados a cada coordinador para su posterior distribución, permitiendo informar de manera directa y eficiente sobre eventos y programas. Sumado que las personas también podrán subir su día a día para darse a conocer.

### **6.2.4 Perfiles de Usuario:**

El diseño de los perfiles de usuario combinará elementos de Instagram y Facebook. La presentación inicial de la información será ordenada y visualmente atractiva, similar a Instagram, mientras que las publicaciones se mostrarán en una lista completa, al estilo de Facebook. Además, se incluirá una sección para descripciones personales y otra para indicar el estado sentimental, inspirado en Tinder pues lo que se busca es generar relaciones de amistad dentro de la institución.

### **6.2.5 Interacciones y seguimientos:**

Los usuarios tendrán la opción de seguir a otros miembros para mantenerse informados de sus publicaciones. Esta funcionalidad busca fomentar la interacción y el seguimiento de actividades dentro de la comunidad académica.

### 6.2.6 Búsqueda de Usuarios:

Se incorporará un buscador que permita localizar a todos los usuarios de la aplicación y visualizar sus perfiles. Este buscador funcionará de manera similar a los de Facebook e Instagram, facilitando la conexión entre los miembros de la comunidad.

### 6.2.7 Sistema de Chat:

Para abordar el principal problema de comunicación en la institución, se incluirán varias modalidades de chat:

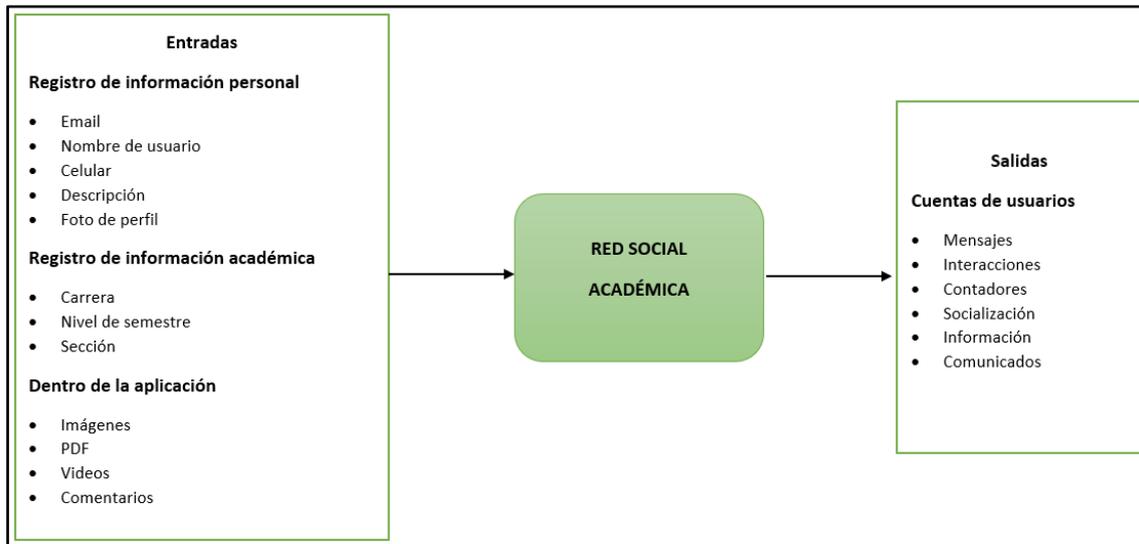
- **Chat 1 a 1:** Para conversaciones privadas entre dos usuarios.
- **Chat Grupal General:** Creado por cualquier usuario para formar grupos de amigos y compartir información dentro de esos grupos.
- **Chat Grupal Administrativo:** Limitado a la creación por parte del personal administrativo, estos grupos estarán diseñados para incluir a miembros de una carrera específica, nivel de semestre y sección, facilitando la rápida incorporación de estudiantes que cumplan con la selección especificada de esta manera minimizando el tiempo de configuración de los grupos.

El diseño de esta red social académica se orienta a mejorar significativamente la interacción y comunicación dentro de la comunidad educativa del Instituto Superior Tecnológico Sudamericano de Quito y otras instituciones similares. Al combinar funcionalidades clave de redes sociales populares con un entorno seguro y escalable, se busca crear una plataforma eficiente y fácil de usar que responda a las necesidades específicas de los usuarios académicos. Este diseño previo sienta las bases para la creación de una aplicación robusta que contribuirá al desarrollo y cohesión de la comunidad educativa.

### 6.3 Diagrama de comportamiento, estructural y mapa conceptual

Para la realización de este proyecto, es fundamental comprender el funcionamiento de la red social académica. Por lo tanto, se ha creado un diagrama de comportamiento que mejora la visión del funcionamiento del sistema.

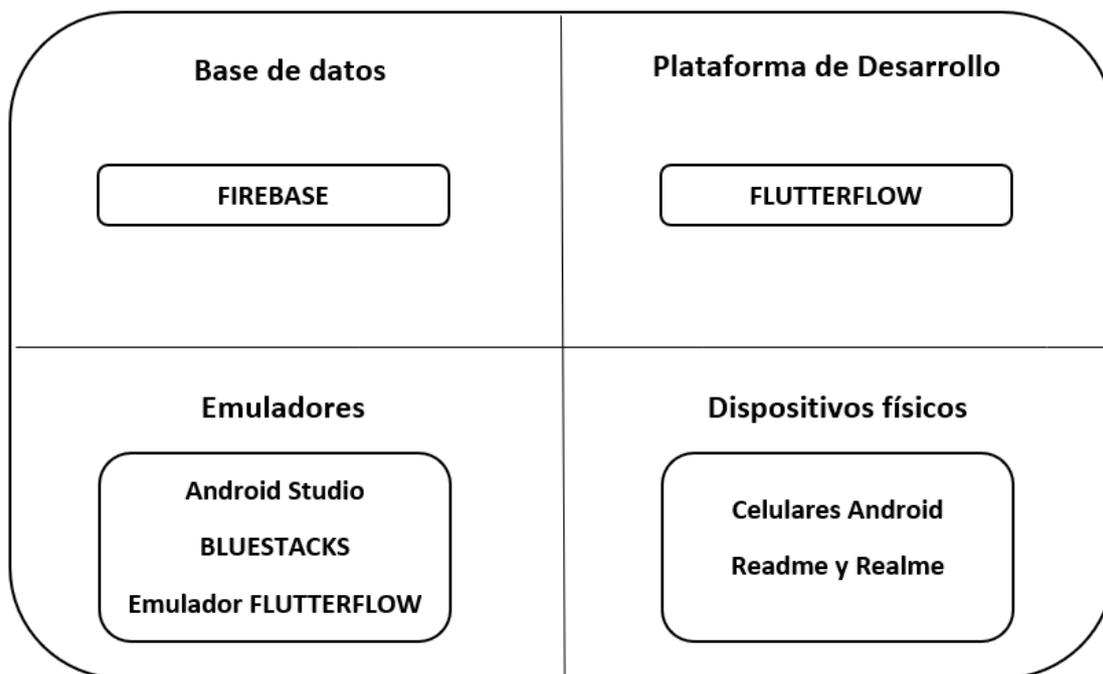
En la siguiente figura se muestra el diagrama de comportamiento:



**Figura 8. Diagrama de comportamiento de la Red Social Académica.**  
Fuente: El Autor.

Como se observa en el gráfico, el recuadro izquierdo contiene las entradas de la aplicación. Aquí se muestra la información de registro de alumnos, profesores y personal administrativo que ingresarán a la plataforma. Asimismo, se detalla el tipo de archivos que los usuarios podrán subir, como imágenes y PDF. Por otro lado, el recuadro derecho del diagrama muestra las salidas, donde se describen las acciones a realizar una vez que los usuarios se han registrado. En este caso, los usuarios tendrán un perfil desde el cual podrán interactuar mediante mensajes, comunicados y la información compartida en sus perfiles.

A continuación, en la siguiente figura, se muestra el diagrama estructural



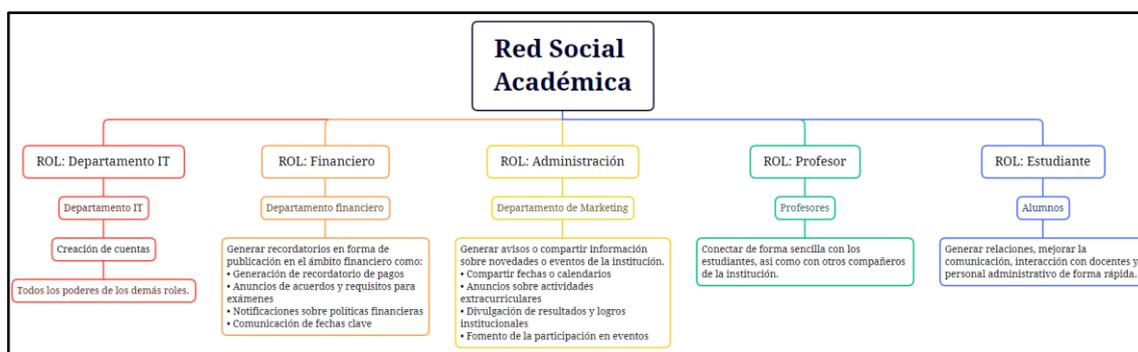
**Figura 9. Diagrama estructural de la Red Social Académica.**  
Fuente: El Autor.

La figura muestra una aplicación diseñada para funcionar como una red social académica, creada con diversas herramientas para facilitar la interacción entre el personal de la institución. A continuación, se detalla la contribución de cada componente del sistema.

1. **Base de datos:** Es el espacio de almacenamiento de la información, permitiendo su administración dentro de la aplicación.
  - **Firestore:** Es una base de datos eficaz para almacenar la información del personal, facilitando su manejo.
2. **Plataforma de desarrollo:** Es el entorno donde se construye el proyecto, generando las pantallas que conformarán la aplicación.
  - **FlutterFlow:** Es una plataforma de bajo código que permite diseñar la aplicación y dotarla de funcionalidad, conectándose directamente a la base de datos.

- 3. Emuladores:** Son entornos que simulan la interfaz de un dispositivo móvil, permitiendo descargar aplicaciones exclusivas para dispositivos móviles desde una PC.
- **Android Studio:** Es un editor de código que también cuenta con una serie de emuladores de diferentes tipos de teléfonos, utilizado para probar la aplicación en diversos sistemas operativos.
  - **BlueStacks:** Es un emulador exclusivamente, con una sola pantalla, que permite probar la aplicación en un segundo plano.
  - **Emulador FlutterFlow:** FlutterFlow proporciona su propio entorno de pruebas con un emulador incorporado para revisar actualizaciones.
- 4. Dispositivos físicos:** Los dispositivos físicos son esenciales para la aplicación, ya que no es lo mismo probar una aplicación en un emulador que en un dispositivo real.
- **Celulares Android:** La aplicación ha sido probada en dispositivos con sistemas operativos Android 13 y 14.

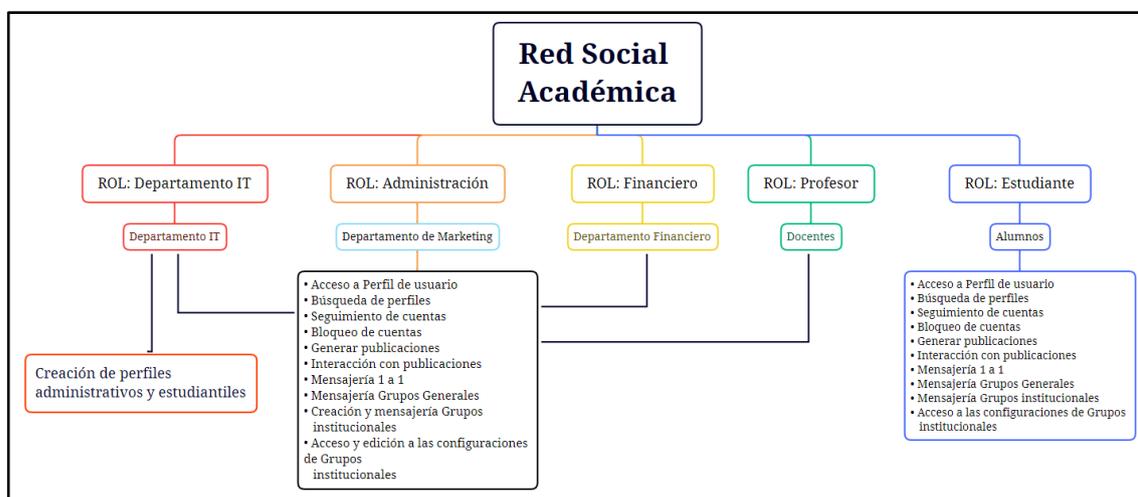
A continuación, se presenta el mapa conceptual de los departamentos y personal que tendrá acceso a la red social académica.



**Figura 10. Mapa conceptual Red Social Académica.**  
Fuente: El Autor.

La aplicación está diseñada para ser utilizada por todos los miembros de la institución en el edificio matriz y sucursales; sin embargo, algunas funciones estarán limitadas exclusivamente para el personal administrativo. Por ejemplo, la creación de cuentas será una función habilitada únicamente para el departamento de IT del Instituto Superior Tecnológico Sudamericano de Quito. En cambio, la creación de grupos institucionales estará disponible para todos los usuarios, exceptuando a aquellos con un rol estudiantil.

A continuación, en la siguiente figura se muestra el mapa conceptual de las funciones permitidas a cada tipo de usuario.



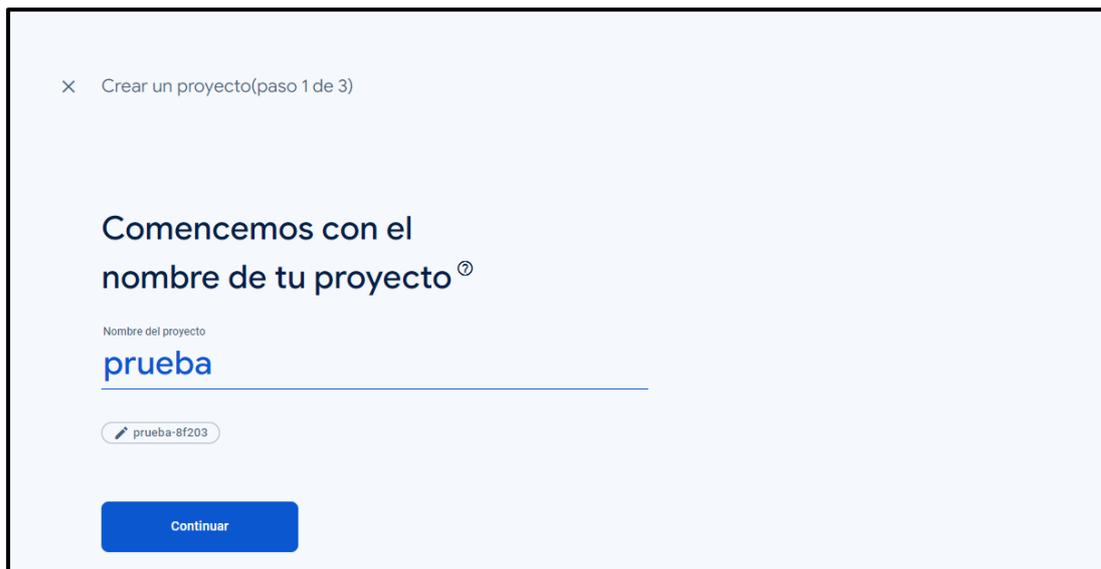
**Figura 11. mapa conceptual de las funciones permitidas a cada tipo de usuario.**

**Fuente: El Autor.**

## 6.4 Creación de un proyecto dentro de Firebase.

Para la red social académica es indispensable el iniciar un proyecto de Firebase. Para configurar el proyecto en Firebase, se procede a otorgar un nombre a la base de datos y a desactivar la función de análisis de Google al inicio del proceso

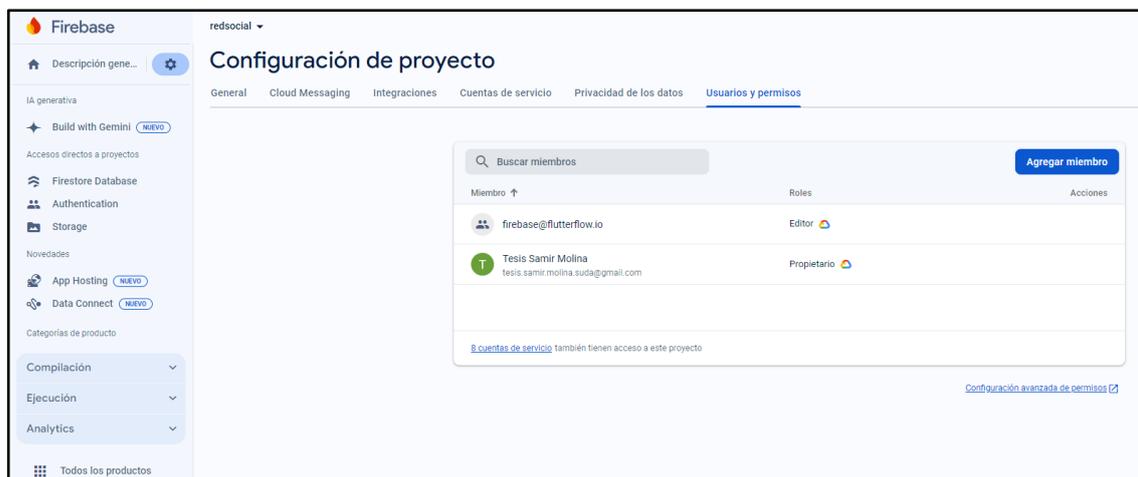
Como se muestra en la siguiente figura.



**Figura 12. Creación proyecto en Firebase.**  
Fuente: (Firebase, s.f.)

Una vez dentro del proyecto, se accede al apartado "Descripción general del proyecto" para gestionar los usuarios y los permisos. En este contexto, se agrega a FlutterFlow como editor, utilizando la dirección de correo electrónico ([firebase@flutterflow.io](mailto:firebase@flutterflow.io))

Se visualiza en la siguiente figura.



**Figura 13. Agregar a FlutterFlow como editor de proyecto.**  
Fuente: (Firebase, s.f.)

Posteriormente, en la sección de "Compilación", se accede a Firestore Database y se crea una base de datos en modo de prueba. Este modo otorga un período de 30 días para utilizar las funciones de lectura y escritura, con la opción de actualizarlo más adelante. Simultáneamente,

dentro del mismo apartado, se ingresa a Authentication y se selecciona la opción "Correo y contraseña", habilitándola en el proceso.

La siguiente figura muestra este proceso.



**Figura 14. Autenticación con correo y contraseña.**

**Fuente:** (Firebase, s.f.)

Finalmente, dentro del apartado de Storage, se elige el modo de prueba para continuar con la configuración del proyecto.

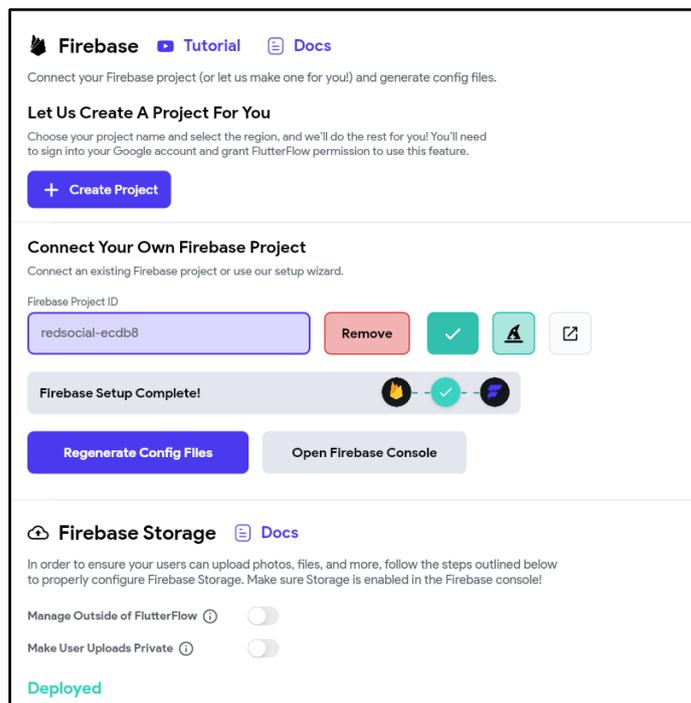
## 6.5 Creación de un proyecto en FlutterFlow

Dentro de FlutterFlow, se selecciona la opción "Create New" y se asigna un nombre al proyecto. Luego, se elige "Create Blank" para crear un lienzo desde cero, evitando la utilización de plantillas predefinidas. Es importante verificar que la opción "Setup Firebase" se encuentre activada y, de forma opcional, se puede habilitar la opción para la plataforma web antes de seleccionar "Next Step".

En el cuadro de texto designado, se ingresa el ID del proyecto de Firebase, el cual se encuentra en la sección "Descripción general del proyecto" dentro de la consola de Firebase. Una vez ingresado el ID, se selecciona la opción "Connect".

A continuación, se elige la opción para generar archivos y, finalmente, se habilita la opción de autenticación, junto con la creación de la colección de usuarios. Además, se selecciona una página de entrada (registro) y una página de inicio para completar el proceso de configuración inicial del proyecto en FlutterFlow.

En la siguiente figura se observa el proyecto configurado.



**Figura 15. Conexión de firebase con FlutterFlow.**  
Fuente: (FlutterFlow, s.f.)

Una vez dentro del proyecto, se procedió a definir las pantallas que conformarían la aplicación, así como sus nombres, funciones y el ciclo de navegación entre ellas. Para ello, se elaboraron bocetos a mano de cada pantalla, donde se especificaron las funciones que debían incluirse, así como la base de datos correspondiente a cada una. Se determinó si era necesario realizar consultas a la base de datos y se identificaron los datos fundamentales que debían mostrarse en primer plano.

Además, se definieron las acciones que ejecutarían los componentes de la aplicación, como la creación, lectura, actualización y borrado de documentos. Este proceso fue meticuloso y requirió tiempo, ya que era crucial definir con precisión cada función para evitar trabajar a ciegas durante el desarrollo de la aplicación. Se reconoció que los obstáculos surgirían en el

camino, y se buscó anticiparlos para abordarlos con éxito y corregir cualquier problema que pudiera surgir durante el desarrollo.

De este proceso, surgieron las siguientes pantallas principales:

- **Página de Inicio:** En esta pantalla, se mostrarán las publicaciones realizadas por los usuarios de la aplicación.
- **Página del Buscador:** Su función principal será realizar consultas a la base de datos para mostrar los usuarios autenticados dentro de la aplicación.
- **Página de Creación de Publicación:** En esta pantalla, los usuarios podrán crear una nueva publicación, incluyendo una imagen, título y descripción correspondiente.
- **Página de Chats:** En esta pantalla, se presentarán en forma de listado los chats en los que participa el usuario.
- **Página de Perfil de Usuario:** Aquí se almacenará toda la información del usuario, incluyendo el número de publicaciones, seguidores, seguidos, nombre de usuario, email de registro, descripción, etc. Además, se mostrarán las publicaciones creadas por el usuario en cuestión.

Adicionalmente a las pantallas principales, se crearon "Extras" que incluyen componentes adicionales destinados a funciones específicas para el usuario autenticado. Estos elementos pueden servir como sub pantallas del perfil del usuario, proporcionar información detallada sobre distintos usuarios, presentar datos complementarios o gestionar ciclos de conversación en los chats. Entre estas pantallas adicionales se encuentran:

- **Pantalla de perfil de usuario:** Permite visualizar la información detallada del usuario autenticado.

- Pantalla de edición de perfil: Proporciona la funcionalidad necesaria para que el usuario modifique su información personal.
- Pantalla de información extra del usuario: Presenta datos adicionales sobre el usuario, más allá de los mostrados en el perfil.
- Pantalla para visualizar seguidores: Muestra la lista de usuarios que siguen al usuario autenticado.
- Pantalla de Likes por publicación: Muestra la cantidad de "Me gusta" que ha recibido cada publicación del usuario.
- Pantalla de Chat inicial: Inicia un nuevo chat entre usuarios.
- Pantalla de Chat comenzado: Permite continuar una conversación iniciada previamente.
- Componente de comentarios: Permite a los usuarios comentar en las publicaciones de otros usuarios.
- Componente de eliminar publicación: Ofrece la opción de eliminar una publicación creada por el usuario.

Una vez definidas las pantallas en papel, el siguiente paso fue crear cada una dentro de la aplicación y comenzar a dotarlas de funcionalidad.

## **6.6 Método de registro de usuarios**

Para el registro de usuarios dentro de la aplicación se decidió utilizar un método externo al proceso normal de registro el cual constaría que el usuario sea quien se registre, sin embargo, no era algo práctico pues la aplicación estaría dedicada exclusivamente a miembros de la institución, por esta razón se implementó un código Python externo el cual al ser ejecutado autenticaría y crearía los perfiles de usuario los cuales estarían definidos dentro de un archivo Excel con los siguientes campos: email, contraseña, nombre de usuario, rol, carrera, nivel de semestre y sección.

Podemos observar el ejemplo en la siguiente figura.

	A	B	C	D	E	F	G
1	email	password	display_name	role	carrera	NivelSemestre	Seccion
2	<a href="mailto:peñaranda@gmail.com">peñaranda@gmail.com</a>	p1726569635	Peñaranda	Estudiante	Desarrollo de Software	Primer Nivel	Diurno
3	<a href="mailto:cruz@gmail.com">cruz@gmail.com</a>	c1727859874	Cruz	Estudiante	Gastronomía	Tercer Nivel	Fin de Semana
4	<a href="mailto:ramos@gmail.com">ramos@gmail.com</a>	r1726847859	Ramos	Estudiante	Desarrollo de Software	Segundo Nivel	Diurno
5	<a href="mailto:sainz@gmail.com">sainz@gmail.com</a>	s1754879869	Sainz	Estudiante	Gastronomía	Quinto Nivel	Diurno
6	<a href="mailto:fabricio@gmail.com">fabricio@gmail.com</a>	f1726369878	Fabricio	Estudiante	Desarrollo de Software	Primer Nivel	Diurno

**Figura 16. Perfiles creados en el archivo Excel.**  
Fuente: El Autor.

### 6.6.1 Creación del archivo Python para registrar usuarios.

Para crear el Script o ejecutable de registro de usuarios lo primero fue ingresar en la página oficial de Python donde se descargó la última versión para Windows del programa, y procederíamos con la instalación respectiva, aquí se otorgó permisos de administrador y se añadió al PATH directamente Python, de esta forma teníamos instalado Python y PIP dentro del sistema operativo.

Se muestra en la siguiente figura:

```

C:\Windows\system32\cmd.exe - python
Microsoft Windows [Versión 10.0.19045.4412]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\SAMIR>python
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
  
```

**Figura 17. Python instalado.**  
Fuente: El Autor.

Posteriormente, se creó una carpeta en el escritorio denominada "Red Social". Dentro de esta carpeta, se generó un archivo Excel que contiene la información básica de los perfiles de usuario. Además, se añadieron las credenciales del proyecto de Firebase, obtenidas a través de Google Cloud Platform. Para ello, se seleccionó el proyecto correspondiente y se accedió a la opción "Cuentas de servicio". En esta sección, se localizó el proyecto y se seleccionó "firebase-adminsdk". Al hacer clic en los tres puntos junto a "firebase-adminsdk", se eligió la opción

"Administrar claves". Finalmente, se creó la clave y se descargó en formato JSON, añadiéndose esta descarga a la carpeta mencionada.



**Figura 18. Credenciales del proyecto Firebase en formato JSON.**  
Fuente: El Autor.

Una vez generado el archivo Excel y obtenidas las credenciales, el siguiente paso consistió en utilizar un editor de código, como Visual Studio Code. Dentro de este entorno, se arrastró la carpeta "Red Social" y se creó un archivo en formato Python (.py). En este archivo, se escribió el proceso que debe seguir el programa al ser ejecutado para agregar y autenticar todos los usuarios en Firebase.

Para que el proceso funcione correctamente, el sistema debe contar con las librerías firebase-admin y openpyxl. Para instalarlas, se abrió una terminal CMD o símbolo del sistema, y se ejecutó el siguiente comando: 'pip install firebase-admin openpyxl'.

La ejecución del código se muestra en la siguiente figura

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.19045.4412]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\SAMIR>pip install firebase-admin openpyxl
```

**Figura 19. Instalación librerías Python.**  
Fuente: El Autor.

En el archivo Python, se comenzó importando las librerías necesarias para interactuar con Firebase y manejar archivos Excel. Las librerías importadas fueron `firebase_admin`, `credentials`, `firestore`, y `auth` del paquete `firebase_admin`, además de `openpyxl` para trabajar con archivos Excel y `datetime` para gestionar fechas y horas.

Inicialmente, se configuró Firebase utilizando un archivo de credenciales JSON específico (`redsocal-ecdb8-cb5f7815857c.json`). Este archivo contiene la información requerida para autenticar la aplicación con el servicio de Firebase.

Dicho proceso se visualiza en la figura a continuación:

```
subir_new_users.py
1  import firebase_admin
2  from firebase_admin import credentials, firestore, auth
3  import openpyxl
4  from datetime import datetime
5
6  # Inicializar Firebase
7  cred = credentials.Certificate("redsocal-ecdb8-cb5f7815857c.json")
8  firebase_admin.initialize_app(cred)
9  db = firestore.client()
```

**Figura 20. Llamar librerías e inicializar Firebase con credenciales.**  
Fuente: El Autor.

A continuación, se cargó el archivo Excel llamado `redsocalFirebase.xlsx` utilizando “`openpyxl`”

Las líneas de código se especifican en la siguiente figura:

```
11  # Leer el archivo Excel
12  file_path = 'redsocalFirebase.xlsx' # Nombre del archivo Excel
13  workbook = openpyxl.load_workbook(file_path)
14  sheet = workbook.active
```

**Figura 21. Llamar al archivo Excel con la librería openpyxl.**  
Fuente: El Autor.

El script continuó iterando sobre las filas del archivo Excel, iniciando desde la segunda fila para omitir los encabezados. Para cada fila, se extrajeron los valores correspondientes a "email", "password", "display\_name", "role", "carrera", "NivelSemestre", y "Sección".

```

16 # Iterar sobre las filas del archivo Excel
17 for row in sheet.iter_rows(min_row=2, values_only=True): # Comienza en la fila 2
18     email, password, display_name, role, carrera, NivelSemestre, Seccion = row

```

**Figura 22. Leer el archivo Excel empezando por la segunda línea.**  
Fuente: El Autor.

Antes de crear un usuario, se verificó si el campo email no estaba vacío. Si había un correo electrónico, se crea un usuario en Firebase Authentication con el método `auth.create_user`, proporcionando el correo electrónico, la contraseña y el nombre de usuario.

Este proceso se muestra en la siguiente figura:

```

20 # Verificar si el campo de email está vacío
21 if email:
22     # Crear el usuario en Firebase Authentication
23     try:
24         user = auth.create_user(
25             email=email,
26             password=password,
27             display_name=display_name,
28         )

```

**Figura 23. Verificar si el campo email no está vacío y crear usuario.**  
Fuente: El Autor.

Después de crear el usuario, se almacenaron los datos adicionales en Firestore utilizando el "uid" del usuario recién creado. Los datos registrados incluyeron "uid", "email", "display\_name", "role", "carrera", "NivelSemestre", "Sección", y el tiempo de creación (`created_time`), que se estableció utilizando la hora actual en formato UTC (Tiempo Universal Coordinado).

Como se observa en la siguiente figura:

```

30 # Guardar los datos adicionales en Firestore con el uid
31 db.collection('users').document(user.uid).set({
32     'uid': user.uid,
33     'email': email,
34     'display_name': display_name,
35     'role': role,
36     'carrera': carrera,
37     'NivelSemestre': NivelSemestre,
38     'Seccion': Seccion,
39     'created_time': datetime.utcnow()
40 })

```

**Figura 24. Guardar datos adicionales del usuario.**  
Fuente: El Autor.

Si se producía algún error durante la creación del usuario, se capturaba y se imprimía un mensaje de error, esto en caso que el usuario ya haya sido creado, finalmente, al completar la iteración sobre todas las filas del archivo Excel, se imprimía un mensaje indicando que el proceso había finalizado.

El método de error se muestra en la siguiente figura:

```

42     print(f"Usuario {email} creado exitosamente.")
43     except Exception as e:
44         print(f"Error al crear el usuario {email}: {e}")
45
46     print("Proceso completado.")

```

**Figura 25. Alertas y proceso culminado.**  
Fuente: El Autor.

Para la creación de cuentas administrativas se replicó el proceso con la diferencia que estas cuentas no contarían con carrera, nivel y sección.

## 6.7 Creación archivo ejecutable para creación de perfiles en un solo paso.

Una vez finalizado el código del archivo Python, se crearon dos ejecutables denominados "1 InstalarDependencias" y "2 EjecutarScript". El primer ejecutable, "1 InstalarDependencias",

se utiliza exclusivamente para instalar las librerías necesarias en un nuevo sistema operativo y solo debe ejecutarse una vez por cada S.O. En caso de que las librerías ya estén instaladas, se procede a ejecutar el segundo ejecutable, "2 EjecutarScript", el cual inicia el proceso de creación de usuarios. Los ejecutables se muestran en la siguiente figura:



**Figura 26. Ejecutables proceso registro.**  
Fuente: El Autor.

Una vez ejecutado el programa, las cuentas se agregarán a Firebase Authentication. Dentro de Firebase, es posible visualizar en la sección de "Authentication" los correos electrónicos registrados, junto con su fecha de creación, la fecha de su último acceso a la aplicación y el ID único generado automáticamente para cada registro.

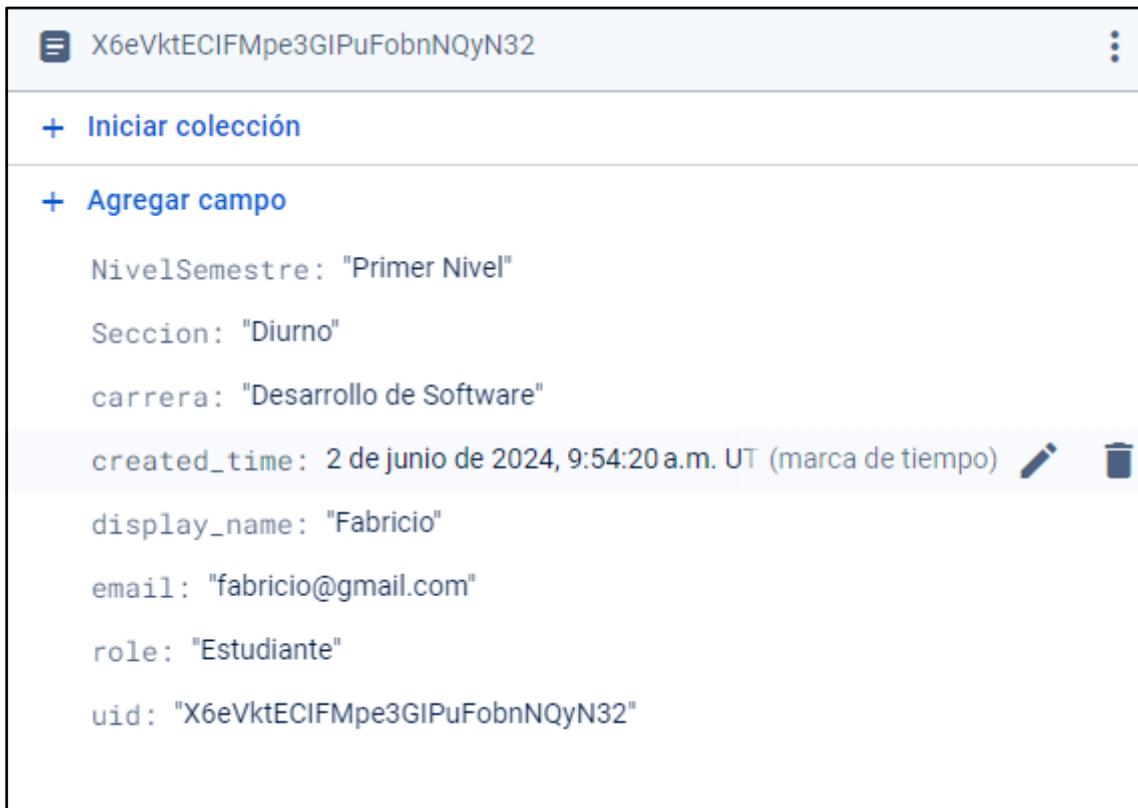
Los usuarios creados se muestran en la siguiente figura:

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
prueba@gmail.com	📧	12 abr 2024	12 abr 2024	mwe5a75aV09F3nLxXGvEHb...
goku@gmail.com	📧	7 abr 2024	7 abr 2024	1YwafHQJAERDCh-gL4tpn...
sami@gmail.com	📧	24 mar 2024	10 abr 2024	MhYDkLU7Y8h8CN5a2C7e...
firebase@flutterflow.io	📧	25 feb 2024	25 feb 2024	N8RavUq04B0t8st2hG2P1Z...
ariana@gmail.com	📧	12 feb 2024	8 abr 2024	b9MFE09a701IA066uT1ey...
valeria@gmail.com	📧	12 feb 2024	12 feb 2024	u3z2mk73MqZ7rv3MUSCh...
wily@gmail.com	📧	12 feb 2024	12 feb 2024	XITCsz5kgTpmNB0uLstVIA...
elizabeth@gmail.com	📧	12 feb 2024	12 feb 2024	GhHF7512FhQpZzqHqXBU...
molina@gmail.com	📧	12 feb 2024	8 abr 2024	XICZJ8R2mTW9HHTVppJ0sz...
sami@gmail.com	📧	10 feb 2024	2 abr 2024	43yzEzjzpbthQyvsTk00Mj...
molinasami4712@gmail.com	📧	10 feb 2024	10 feb 2024	ggfDZVqGyWV1RwT9qyV0G...

**Figura 27. Firebase Autentication.**  
Fuente: El Autor.

Por su parte, en Firestore Database se puede observar la creación de una colección llamada "Users", la cual contiene documentos separados por su ID. Dentro de estos documentos se encuentra la información del usuario la cual viene del archivo Excel

El ejemplo del documento se visualiza en la siguiente figura:



**Figura 28. Información de perfil de usuario exportada desde Excel.**  
Fuente: El Autor.

## 6.8 Ciclos y funciones de las pantallas de la aplicación

### 6.8.1 Inicio de sesión y/o Registro dentro de la aplicación

Al abrir la aplicación, el usuario será recibido con la pantalla de bienvenida a la red social. Al seleccionar la flecha de continuar, se procederá a la página de inicio de sesión, donde se solicitará el correo electrónico y la contraseña asociados al perfil del usuario.

El inicio de sesión se muestra en la siguiente figura:



Figura 29. Inicio de sesión.  
Fuente: El Autor.

El personal administrativo de la institución tendrá acceso al apartado de registro de cuentas institucionales. Para ingresar a esta sección, será necesario introducir un PIN que otorgará acceso a la creación de cuentas desde este lugar.

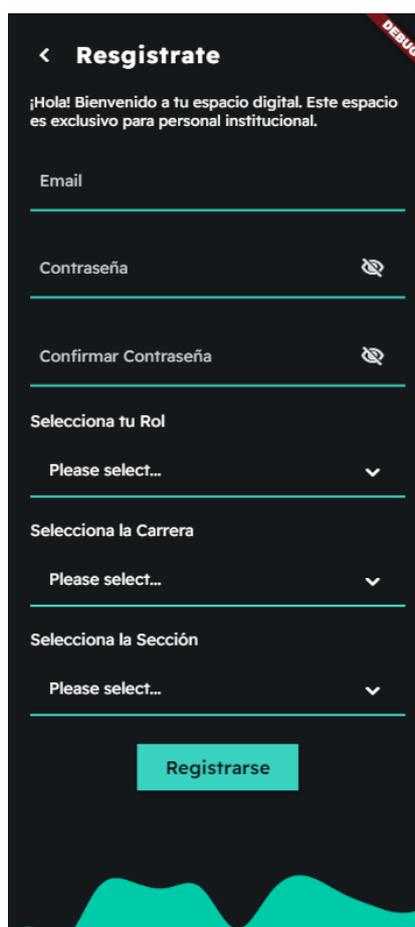
El ingreso a esta sección se muestra en la siguiente figura:



**Figura 30. PIN para la sección de registro.**  
Fuente: El Autor.

En la pestaña de registro, se solicitará diversa información, como el correo electrónico, la contraseña, la carrera, el nivel y la sección correspondiente. Una vez completados estos datos, la cuenta podrá ser registrada exitosamente.

Este tipo de registro se muestra en la siguiente figura:



**Figura 31. Sección de registros dentro de la aplicación.**  
Fuente: El Autor.

## 6.8.2 Perfil del usuario autenticado

Una vez autenticado, el usuario será redirigido automáticamente a su propio perfil. Aquí se exhibirá la información que el usuario decida publicar. Además, se incluirán tres tipos de contadores: el primero indicará el número de publicaciones del perfil, el segundo mostrará la cantidad de seguidores que tiene la persona, y el tercero reflejará el número de cuentas que el usuario sigue. Finalmente, se presentarán las publicaciones realizadas por el usuario en forma de lista.

La página de perfil se muestra en la siguiente figura:



Figura 32. Pantalla del perfil de usuario.  
Fuente: El Autor.

Como se puede observar, esta pantalla incluye una serie de botones que permiten navegar a diferentes secciones relacionadas con el perfil del usuario. Entre estos botones se encuentra el de "Más información", el cual redirige a una página detallando la información pública que el usuario desee mostrar. Además de este botón, están disponibles los botones para activar el modo oscuro o claro, así como uno para editar la información del perfil del usuario y otro para cerrar la sesión. Las funcionalidades de estas páginas se explorarán más adelante.

### **6.8.3 Pantalla de edición de perfil**

La pantalla de edición de perfil está diseñada para permitir al usuario editar y mostrar la información que desee hacer pública. Entre los elementos editables se encuentran la foto de perfil, el nombre de usuario, la descripción personal, el número de teléfono, la fecha de cumpleaños, el estado sentimental, el género y el semestre al que pertenece.

Además de los campos editables, la página incluye un botón para cancelar y salir sin guardar los cambios, así como otro botón dedicado a guardar la información actualizada.

La página de edición se muestra en la siguiente figura:

Agrega una foto de perfil

Nombre  
Jorge

Describete 5/15  
Lorem Ipsum es simplemente el texto de relleno de las imprentas y archivos de texto. Lorem Ipsum ha sido el texto de relleno

Celular  
0995167874

Cumpleaños dd/mm/yy  
27-11-2003

Situación Sentimental  
No sólo sobrevivió 500 años, sino que tambie

Selecciona tu Genero... ▼

Selecciona tu Semestre... ▼

**Guardar Cambios**

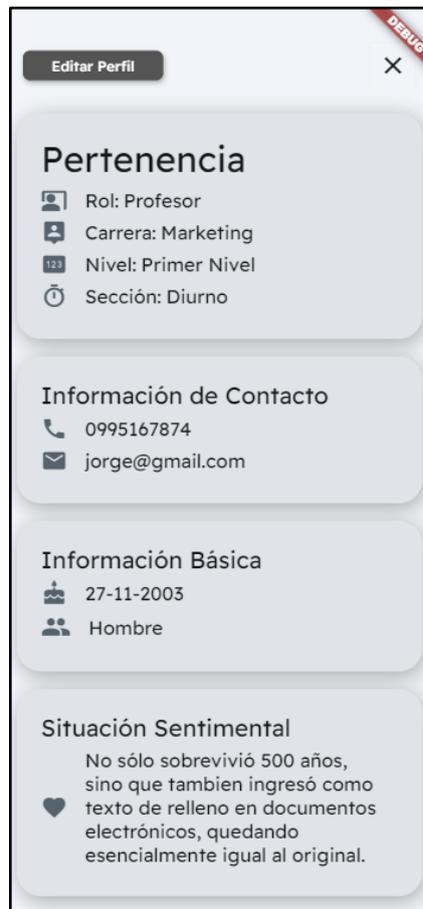
**Figura 33** Pantalla de edición de perfil.  
Fuente: El Autor.

#### 6.8.4 Pantalla de información del usuario autenticado

La pantalla de información del usuario está compuesta por varios cuadros informativos, organizados en diferentes secciones. La primera sección muestra la pertenencia académica, donde se visualizan el rol, la carrera, el nivel y la sección del usuario. La segunda sección se centra en la información de contacto, mostrando el número de teléfono y el correo electrónico con el que el usuario está autenticado. También se incluye información básica como el género y la fecha de cumpleaños, así como un espacio dedicado a la situación sentimental del usuario.

Adicionalmente, se ha incorporado un botón que redirige a la pantalla de edición de perfil, facilitando así la actualización de la información personal.

Esta pantalla se visualiza en la siguiente figura:

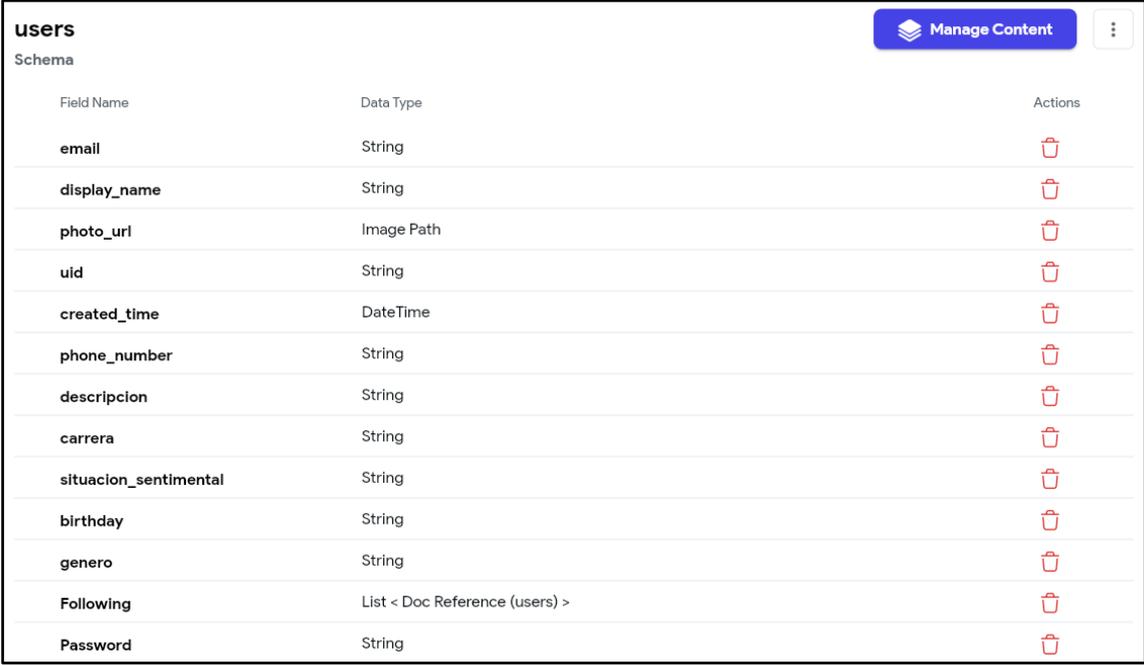


**Figura 34** Pantalla de información del usuario.  
**Fuente:** El Autor.

#### 6.8.4.1 Colección de “users” de Firebase

La colección de usuarios es la más importante dentro de la aplicación, ya que almacena toda la información relevante de cada persona, guardando cada cuenta como un documento único. En esta colección se almacenan datos como el rol, la carrera, el nivel, la sección y toda la información que el usuario desee hacer pública. Además, esta colección guarda las referencias de las cuentas que el usuario sigue y las que lo siguen, así como las cuentas bloqueadas, en caso de ser necesario.

La colección se muestra en la siguiente figura:



Field Name	Data Type	Actions
email	String	
display_name	String	
photo_url	Image Path	
uid	String	
created_time	DateTime	
phone_number	String	
descripcion	String	
carrera	String	
situacion_sentimental	String	
birthday	String	
genero	String	
Following	List < Doc Reference (users) >	
Password	String	

**Figura 35. Colección de usuarios dentro de FlutterFlow y Firebase.**  
Fuente: El Autor.

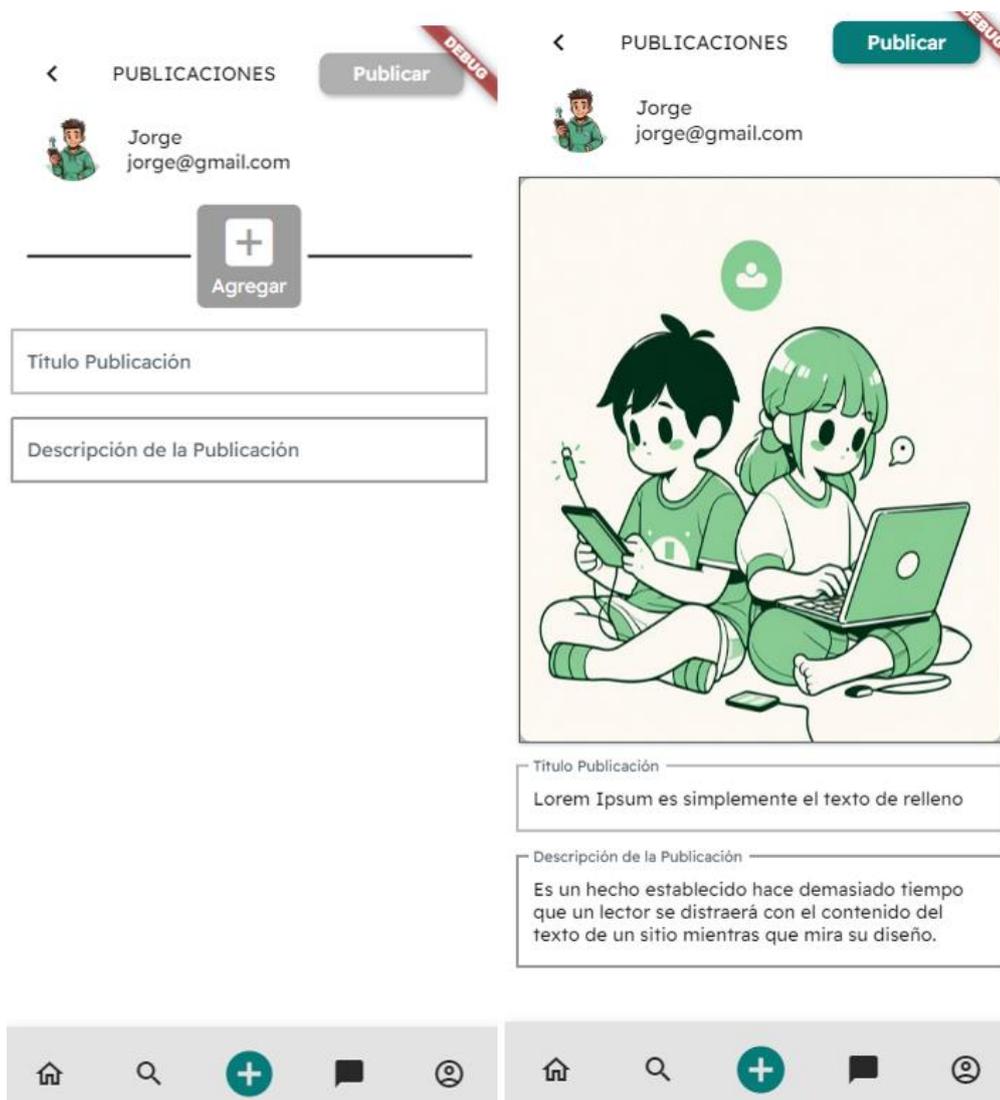
### 6.8.5 Pagina para subir publicaciones a la red social académica

Esta página está diseñada con el propósito de permitir la subida de contenido visual a la aplicación. Cuenta con una sección donde se puede subir una imagen que incluya un título y una descripción, permitiendo que otros usuarios comenten en las publicaciones. Esta pestaña se encarga de crear documentos dentro de una colección llamada "publicaciones", donde cada publicación tendrá un número de identificación único. Esto es necesario para que las publicaciones puedan ser comentadas, recibir "me gusta" o ser compartidas, como se explicará más adelante.

Además, se ha implementado una restricción que impide la creación de una publicación si no se adjunta una imagen. Es decir, si no hay una imagen para publicar, el documento no será creado.

Las publicaciones se harán visibles en el perfil de la persona encargada de generar el contenido y también en la página de inicio o home de la aplicación.

La visualización de esta pantalla se puede ver en la siguiente figura:



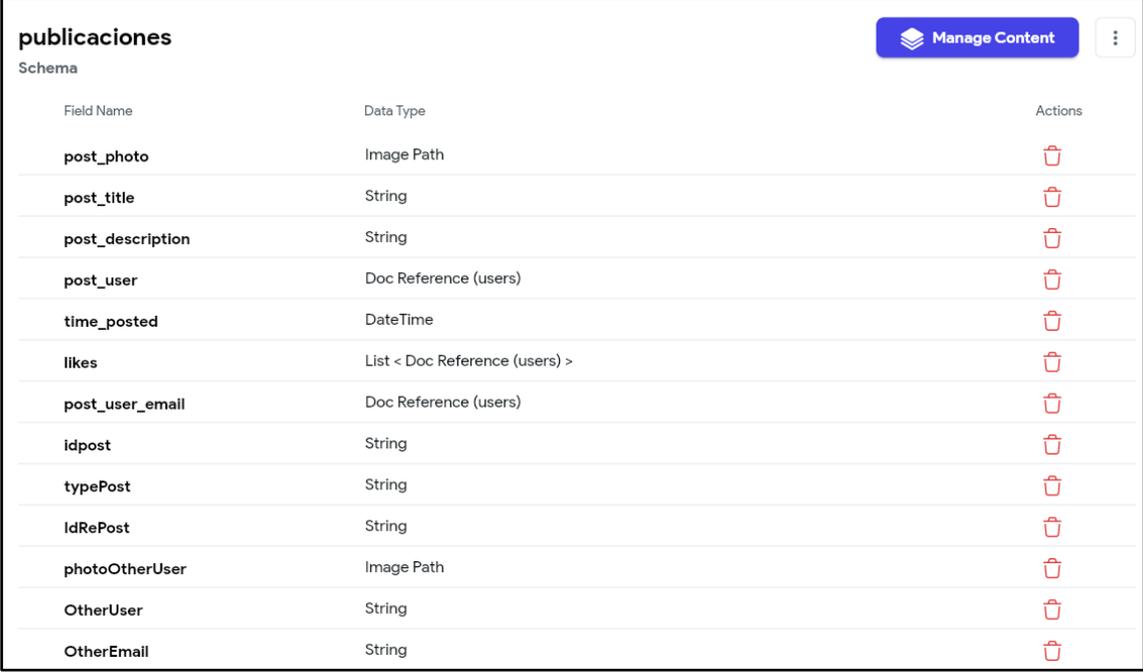
**Figura 36. Pantalla Publicaciones.**  
Fuente: El Autor.

#### 6.8.5.1 Colección de “publicaciones” de Firebase

La colección de publicaciones es fundamental para almacenar cada una de las entradas generadas por los usuarios. Esta colección incluye la imagen, el título y la descripción proporcionados por las personas, además de un campo en forma de lista para registrar a aquellos usuarios que den "me gusta" a la publicación. También cuenta con campos adicionales para almacenar la información de otros usuarios en caso de que decidan compartir la publicación.

Para gestionar las publicaciones, se utilizan dos tipos de ID: uno para publicaciones normales y otro secundario para las publicaciones compartidas. Esto garantiza un seguimiento preciso y una mejor organización de las interacciones.

La estructura de la colección se visualiza en la siguiente figura:



Field Name	Data Type	Actions
post_photo	Image Path	
post_title	String	
post_description	String	
post_user	Doc Reference (users)	
time_posted	DateTime	
likes	List < Doc Reference (users) >	
post_user_email	Doc Reference (users)	
idpost	String	
typePost	String	
IdRePost	String	
photoOtherUser	Image Path	
OtherUser	String	
OtherEmail	String	

**Figura 37. Colección de publicaciones dentro de FlutterFlow y Firebase.**  
Fuente: El Autor.

### 6.8.6 Pantalla de inicio o home

La pantalla de inicio de la aplicación está diseñada exclusivamente para mostrar las novedades recientes publicadas por los usuarios. Además de su diseño atractivo, incluye varias funcionalidades. En la esquina superior derecha, se encuentran dos botones que actúan como enlaces: el primero dirige a la página de publicaciones para generar una nueva publicación, y el segundo lleva rápidamente al listado de chats individuales.

Debajo de estos botones, se pueden observar las publicaciones realizadas por la comunidad. Cada publicación muestra la información del usuario que la creó, junto con la imagen, el título, la descripción y la fecha de publicación. Toda esta información se obtiene directamente de la

colección "publicaciones", donde se almacenan cada uno de estos posts. Las publicaciones ofrecen varias opciones de interacción, como dar "me gusta", comentar o compartir.

La pantalla y el formato de publicación se muestran en la siguiente figura:

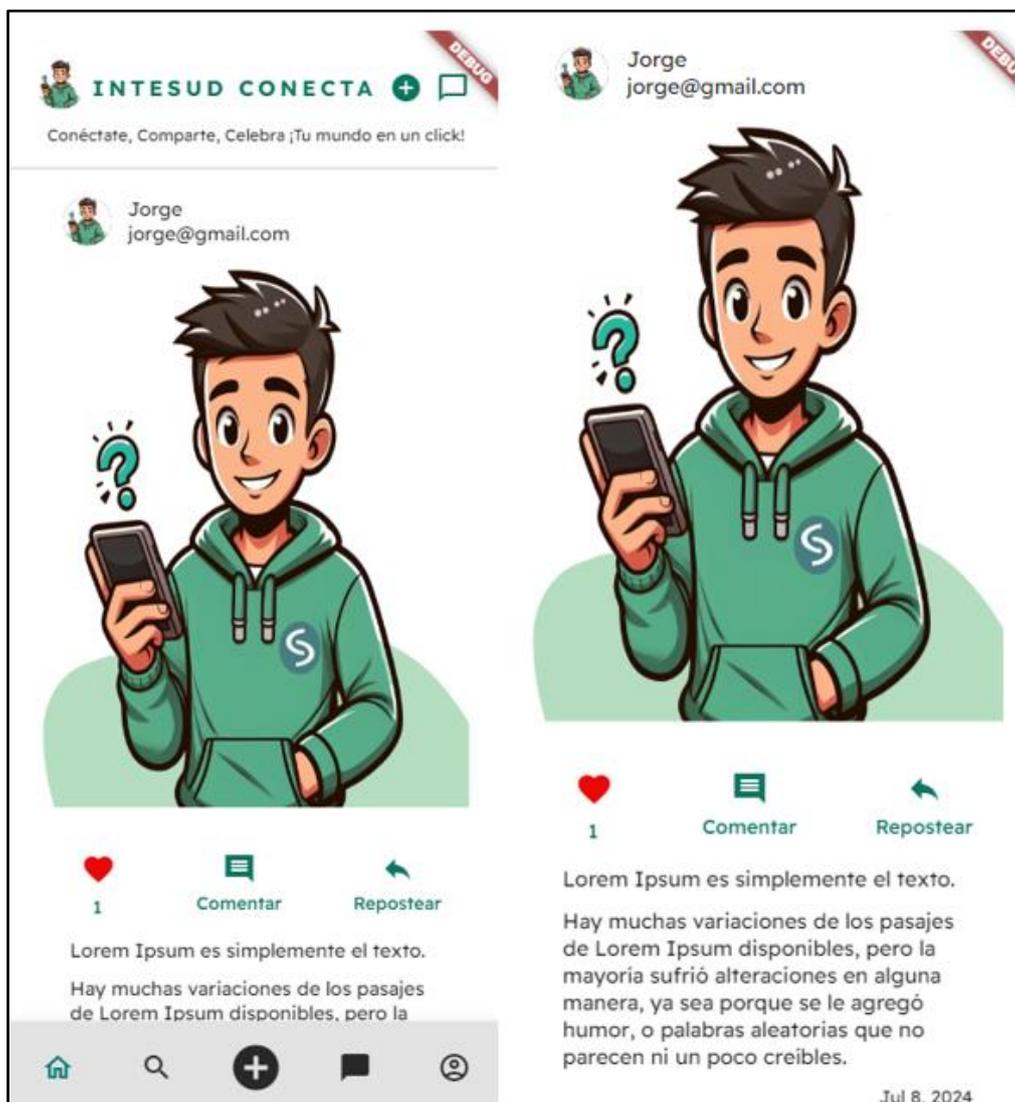


Figura 38. Pantalla de inicio o home.  
Fuente: El Autor.

### 6.8.6.1 Función dar me gusta a una publicación

Las interacciones en una publicación son fundamentales dentro de una red social, ya que fomentan la participación y generan interés dentro de la comunidad. Por ello, la función de dar "me gusta" es imprescindible. En este proyecto, se ha aprovechado las herramientas y facilidades que ofrece FlutterFlow para implementar esta funcionalidad.

Se decidió utilizar un ícono con cambio de estado. Al activarse, guarda la referencia del usuario que dio "me gusta" dentro de una lista en la misma colección de publicaciones. Esto permite luego contar los usuarios en dicha lista y mostrar cuántos "me gustas" tiene la publicación.

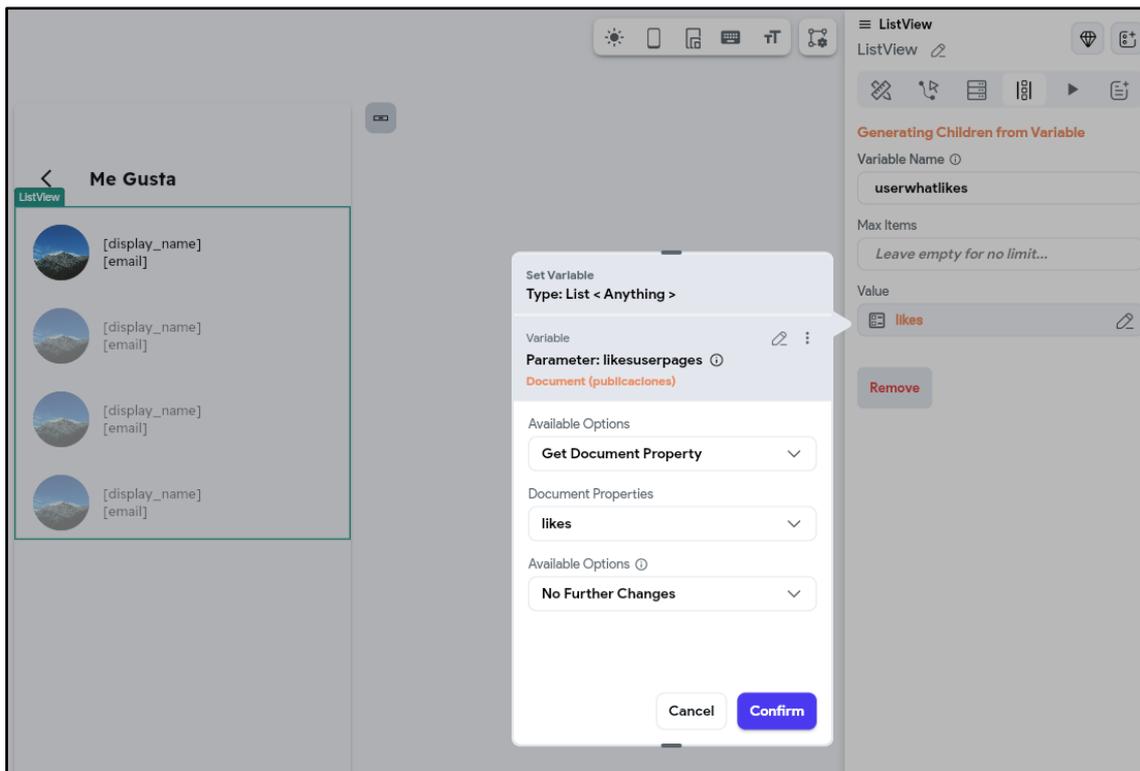
Los estados de este ícono se pueden ver en la siguiente figura:



**Figura 39. Estados icono me gusta.**  
**Fuente: El Autor.**

Esta función también tiene una característica adicional, ya que es importante conocer quiénes han dado "me gusta" a una publicación. Por ello, se decidió implementar una navegación hacia una página extra que muestra, en forma de listado, todos los usuarios que dejaron su "me gusta". Para lograr esto, se creó una página que realiza una consulta a la colección "users", la cual se relaciona con la colección "publicaciones" para listar los usuarios referenciados en la lista de "likes".

La consulta se muestra en la siguiente figura:



**Figura 40** Consulta listado de usuarios que dan me gusta.  
**Fuente:** El Autor.

La página en funcionamiento se muestra en la siguiente figura:

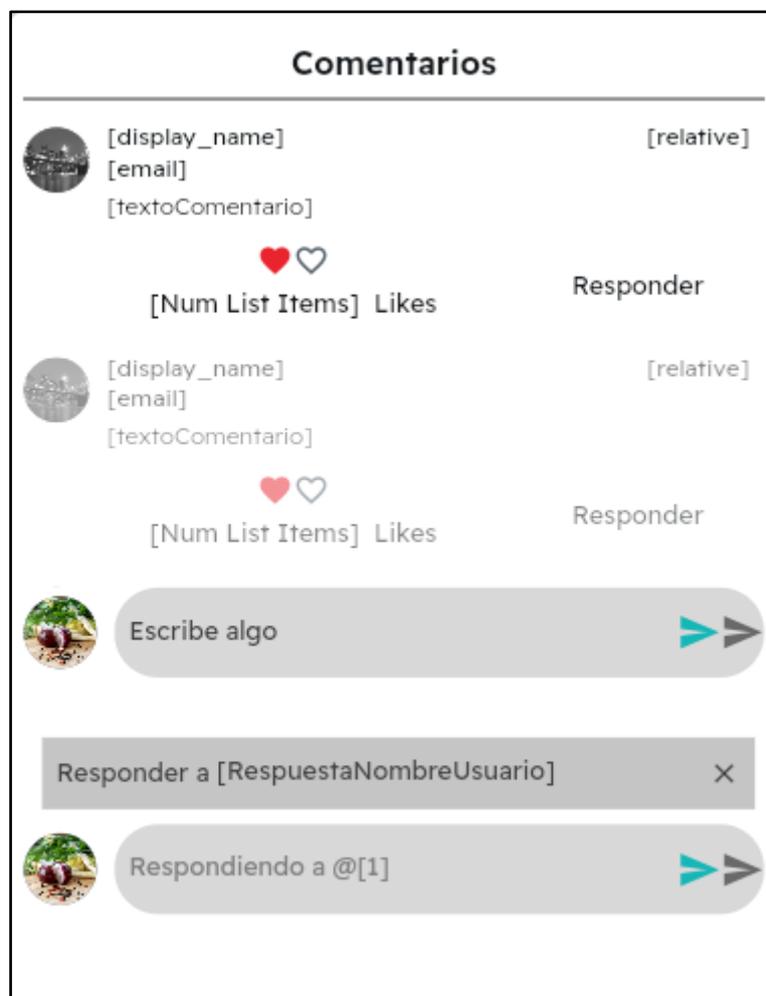


**Figura 41** Pantalla de usuarios que les gusto la publicación.  
**Fuente:** El Autor.

### 6.8.6.2 Función comentar a una publicación

La función de comentar es esencial para interactuar con otros usuarios dentro de una publicación, ya que permite expresar opiniones sobre el contenido publicado. Para implementar esta funcionalidad, se decidió crear una sub ventana que aparecerá al seleccionar el ícono de comentar. Esta pantalla está directamente relacionada con el ID de la publicación, lo que garantiza que los comentarios se muestren únicamente en la publicación correspondiente, evitando así la mezcla de comentarios en diferentes publicaciones.

Esta sub ventana consta de varias secciones, como se visualiza en la siguiente figura:



**Figura 42. Sección de comentarios.**  
**Fuente: El Autor.**

La zona principal de esta sección se centra en los comentarios. Cada comentario mostrará la información del usuario que realizó la publicación, incluyendo su foto de perfil, nombre de usuario y correo electrónico autenticado, junto con la hora del comentario, seguido del mensaje. Cada comentario se almacenará directamente en una nueva colección llamada "comentarios".

Cada comentario contará con la opción de dar "me gusta", replicando el proceso de "me gusta" descrito anteriormente, y también permitirá responder al comentario. Para implementar esto, se crearon dos variables de estado. La primera variable, denominada "EsComentarioRespuesta", de tipo booleano con un valor inicial de false, se utiliza para determinar si el comentario es una respuesta a otro comentario. La segunda variable, "RespuestaNombreUsuario", de tipo string, almacena temporalmente el nombre del usuario al que se está respondiendo.

Estas variables se visualizan en la siguiente figura:

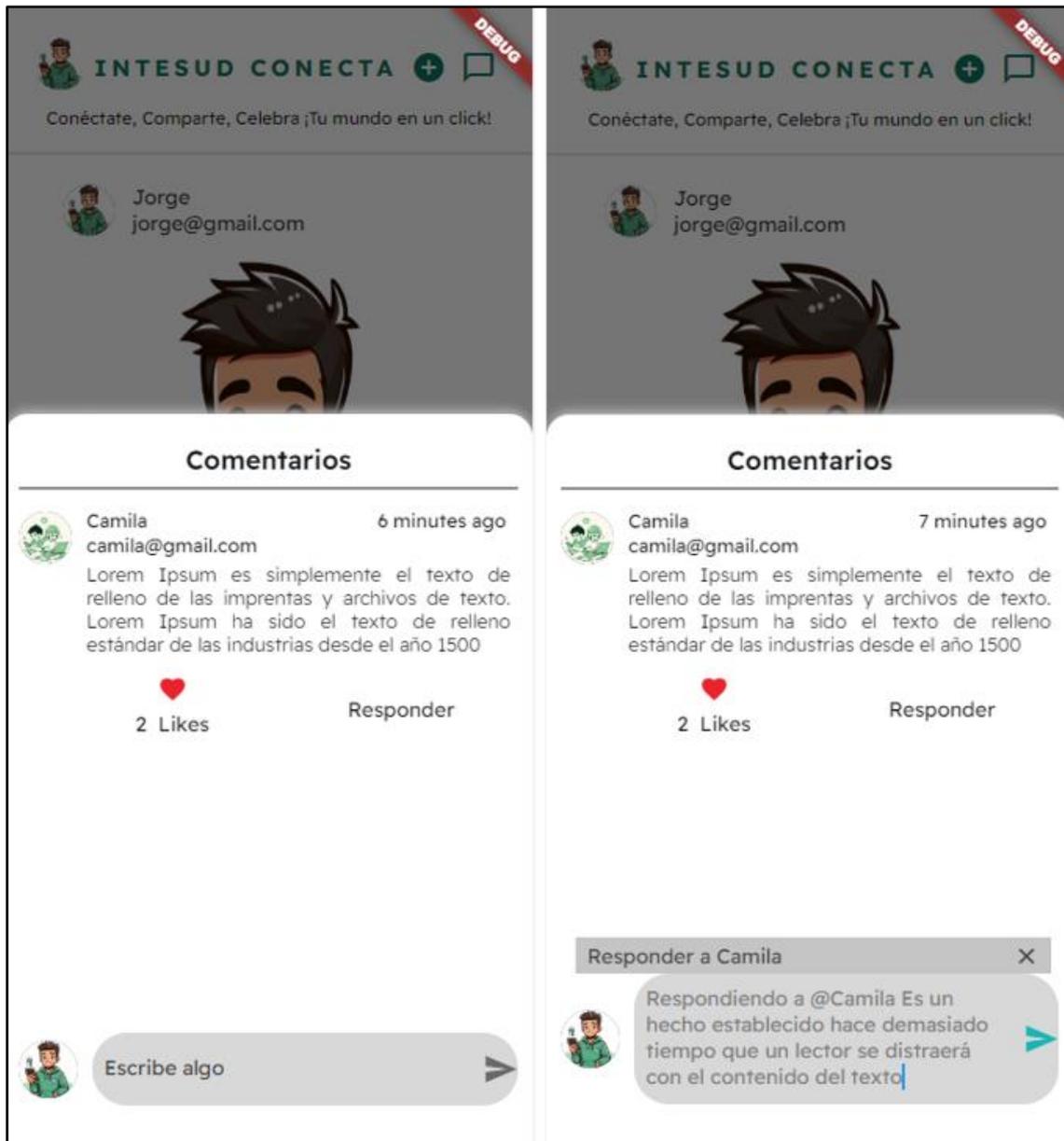
⋮	EsComentarioRespuesta	Boolean	false
⋮	RespuestaNombreUsuario	String	false

**Figura 43. Variables de estado.**  
Fuente: El Autor.

La siguiente sección se divide en dos partes, ya que es el área destinada a escribir comentarios. Esta área es dinámica y cambiará según el estado de la variable "EsComentarioRespuesta". Si esta variable mantiene su valor en false, se crearán comentarios normales. Sin embargo, al seleccionar la opción de responder, el valor de esta variable cambiará a true, lo que hará que el cuadro de comentar se reemplace por uno nuevo diseñado para responder.

Este nuevo cuadro permitirá almacenar temporalmente el nombre del usuario al que se está respondiendo, creando así un nuevo comentario con un ID de respuesta. Esto facilita la interacción y organización de las respuestas dentro de la misma publicación.

La sección de comentarios funcional se muestra en la siguiente figura:

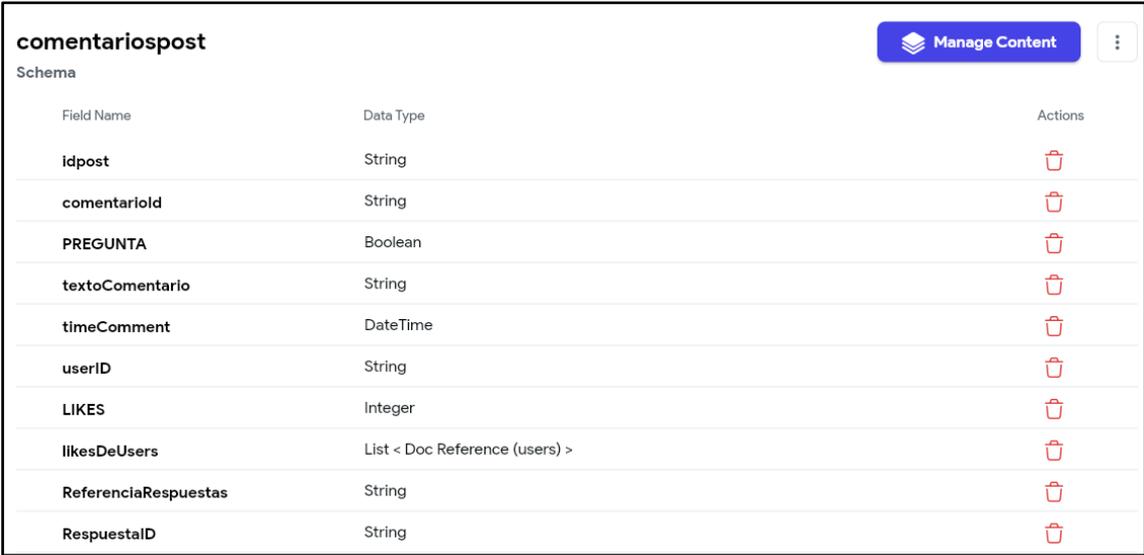


**Figura 44. Sección de comentarios funcional.**  
Fuente: El Autor.

### 6.8.6.2.1 Colección de “comentarios” de Firebase

La colección "comentarios" se encarga de almacenar los comentarios de cada publicación, y está relacionada con la colección de "publicaciones" mediante el ID de la publicación correspondiente. Cada comentario tendrá su propio identificador único. En esta colección se almacenará el texto del comentario, el usuario que lo realizó, los usuarios que dieron "like" al comentario para poder contabilizarlos, así como un identificador único en caso de que el comentario sea una respuesta a otro comentario.

La estructura de la colección se observa en la siguiente figura:



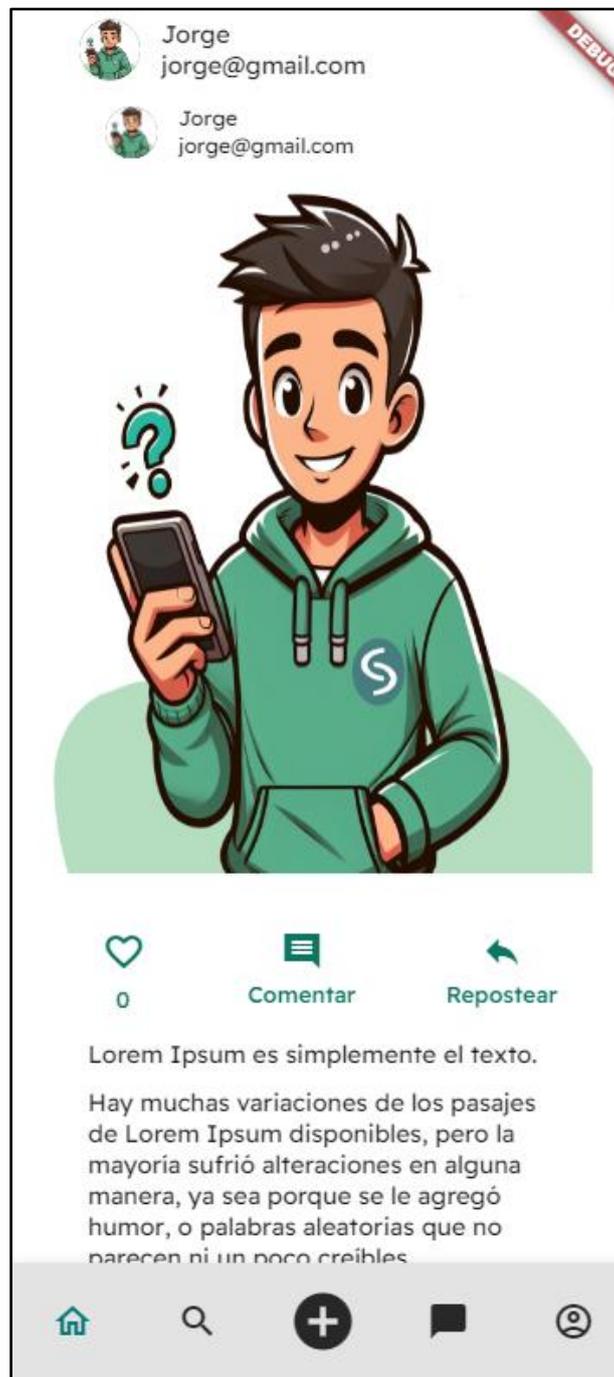
Field Name	Data Type	Actions
idpost	String	
comentarioid	String	
PREGUNTA	Boolean	
textoComentario	String	
timeComment	DateTime	
userID	String	
LIKES	Integer	
likesDeUsers	List < Doc Reference (users) >	
ReferenciaRespuestas	String	
RespuestaID	String	

**Figura 45. Colección de comentarios dentro de FlutterFlow y Firebase.  
Fuente: El Autor.**

### 6.8.6.3 Función compartir una publicación

La función de compartir publicaciones se asemeja al proceso de realizar una publicación propia. En este caso, la aplicación se encarga de almacenar la imagen, título, descripción, tiempo de publicación y la información del usuario original. Estos elementos se integran automáticamente en nuestro perfil como una nueva publicación. Es importante destacar que, en este proceso, los comentarios se conservan, mientras que los "me gusta" se registrarán como una nueva interacción.

Para ilustrar el ejemplo de reposteo, se presenta la siguiente figura:



**Figura 46. Publicación repostada.**  
Fuente: El Autor.

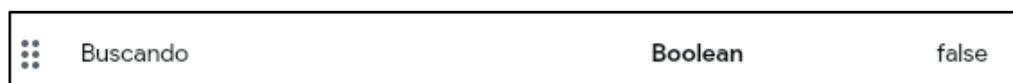
### 6.8.7 Pantalla Buscador de perfiles

Una de las funciones cruciales en cualquier red social es la capacidad de buscar y conectar con otros usuarios. Por eso, es esencial contar con un buscador de perfiles bien diseñado. Esta página incluirá un campo de búsqueda donde los usuarios podrán ingresar texto para encontrar perfiles específicos. Justo debajo, habrá un listado general que mostrará todos los perfiles existentes en la plataforma.

El buscador de perfiles funciona mediante un filtro en la colección "users", aprovechando la funcionalidad de búsqueda simple ofrecida por FlutterFlow. Esta acción inspecciona varios campos de la colección, como nombre de usuario, correo electrónico, número de teléfono, carrera, nivel de semestre o sección. Esto permite a los usuarios encontrar perfiles de manera rápida y eficiente, adaptando dinámicamente la lista de resultados según la búsqueda realizada.

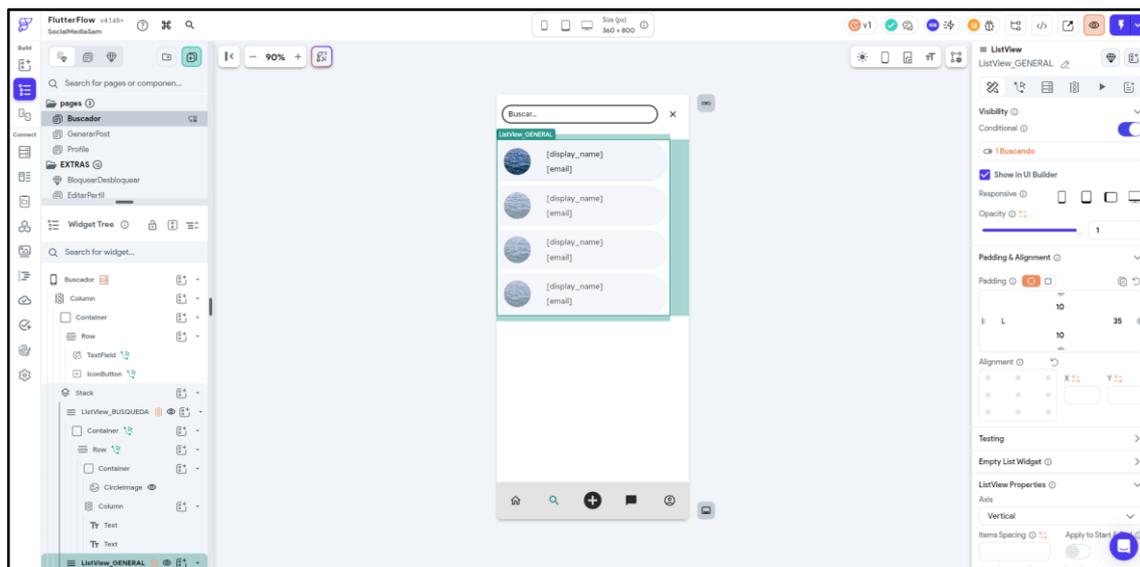
Para lograr esta dinámica de actualización de la lista, se han implementado dos tipos de listados: uno general que muestra todos los usuarios y otro que se ajusta automáticamente según los términos de búsqueda ingresados, sumado a esto se implementó una nueva variable de estado, la cual tiene por nombre "Buscando", al ser de tipo booleano esta variable inicia en false, y al ingresar texto en la barra del buscador esta cambia automáticamente a true haciendo que la lista cambie automáticamente.

La variable creada se muestra en la figura a continuación:



**Figura 47. Variable del buscador.**  
**Fuente: El Autor.**

La estructura y funcionalidad de esta pantalla se detallan en la siguiente figura:

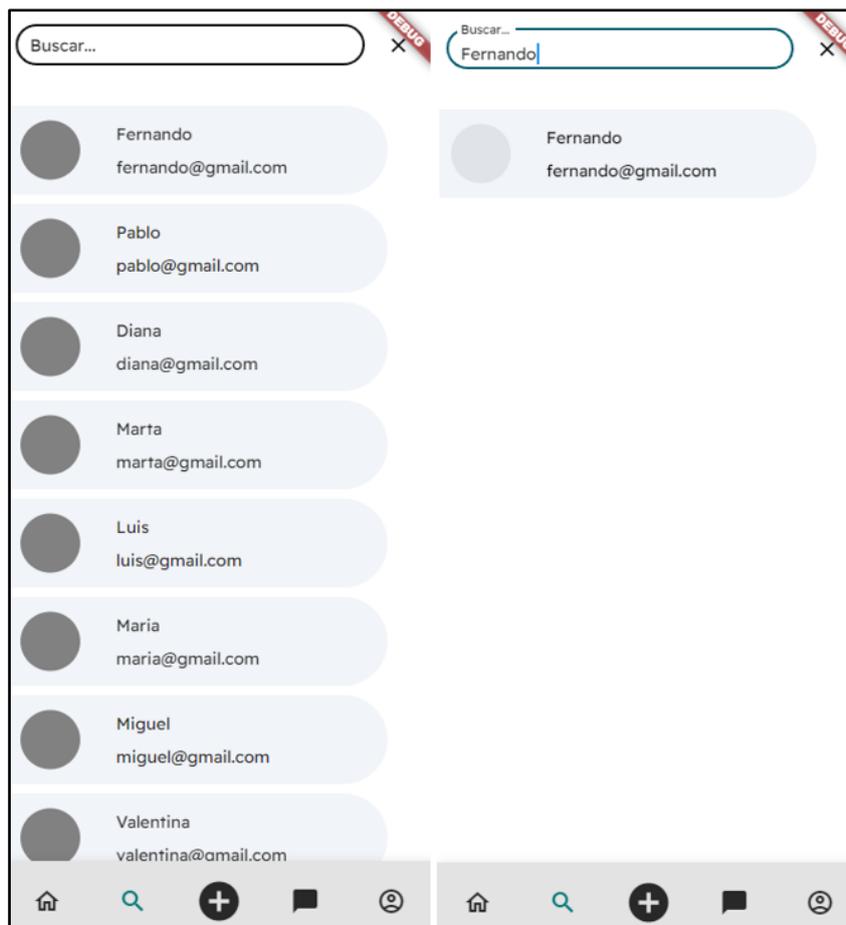


**Figura 48. Construcción pantalla buscador.**  
Fuente: El Autor.

En esta figura se observa la lista general como la lista de búsqueda, como se puede observar tiene una condición según la variable “Buscando”, pues como ya lo dijimos la lista se actualiza en caso de esta variable ser verdadera.

El buscador tiene integrado una navegación directa al perfil de una persona, al seleccionar el perfil nos enviara al perfil de dicha persona.

En la siguiente figura se muestra el buscador siendo utilizado dentro de la aplicación:



**Figura 49. Pantalla del Buscador de perfiles.**  
**Fuente: El Autor.**

### 6.8.8 Pantalla de perfil de otro usuario

La pantalla de perfil de un usuario se asemejará a la de nuestro propio perfil, destacando inicialmente la información que el usuario haya hecho pública. Esto elimina la necesidad de crear una pantalla adicional para detalles adicionales, como ocurre en el perfil del usuario autenticado. Además, incluirá un icono de candado que al presionarlo abrirá una sub pantalla para bloquear al usuario respectivo, evitando así la recepción o envío de mensajes entre ambos.

La funcionalidad del botón de bloqueo se ilustra en la siguiente figura:



**Figura 50. Funcionalidad de bloqueo de usuario.**  
**Fuente: El Autor.**

Dentro de la colección de usuarios, se gestionan dos campos: "Mebloquean" y "UsuariosBloqueados". En estos campos, en nuestros documentos de Firebase, se guardan los IDs de los usuarios bloqueados. Asimismo, nuestro ID se almacena en el campo "me bloquean" del documento del otro usuario. Esto asegura que la comunicación bidireccional entre ambos usuarios sea bloqueada efectivamente.

En lugar de los botones tradicionales de editar perfil, cerrar sesión y cambiar modo, esta pantalla contendrá dos botones específicos: uno para seguir o dejar de seguir al usuario y otro para enviar mensajes directos.

Para gestionar el seguimiento o dejar de seguir a un usuario, utilizamos una lógica similar a la del botón de bloqueo. Utilizamos los campos "Follow" y "Following": cuando seguimos a un usuario, nuestro ID se agrega a su lista de seguidores, mientras que el ID de ese usuario se agrega a nuestra lista de seguidos. Esta estructura nos permite mantener un conteo preciso de los seguidores y seguidos.

La estructura y funcionalidad detallada de esta pantalla de perfil de usuario se presenta en la siguiente figura:



**Figura 51. Pantalla de perfil de usuario diferente.**  
Fuente: El Autor.

#### 6.8.8.1 Conteo de seguidores y seguidos

En una red social, resulta fundamental poder ver quién nos sigue y a quién seguimos, así como obtener esta misma información de otros perfiles. Esta funcionalidad facilita el contacto rápido con otros usuarios de la plataforma y permite seguir a personas conocidas a través de conexiones comunes. Por ello, se decidió implementar una única pantalla que, según el perfil

consultado, accede a su lista de seguidores (followers) o seguidos (follows). Si se trata de nuestro propio perfil, la página utilizará nuestro identificador, haciéndola también útil para nosotros.

La pantalla está diseñada para funcionar en dos etapas: la primera muestra exclusivamente a los seguidores, mientras que la segunda etapa está dedicada a las personas seguidas.

Esta funcionalidad se detalla en la siguiente figura:



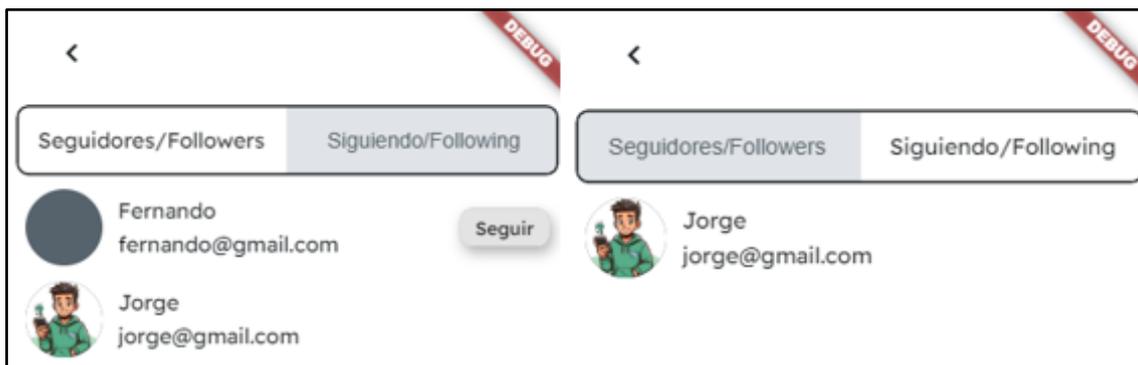
**Figura 52. Pantalla de seguidores o seguidos.**  
Fuente: El Autor.

Esta pantalla incorpora una lógica meticulosamente diseñada, ya que debe permitir a los usuarios seguir o dejar de seguir a otras cuentas. La pantalla puede mostrar usuarios con los que ya exista una relación dentro de la aplicación, por lo cual el botón, en lugar de "seguir", deberá ofrecer la opción de "dejar de seguir". Esta funcionalidad debe estar presente en ambas instancias: seguidores y seguidos. Sin embargo, existe una condición adicional: si el perfil que aparece en seguidores o seguidos es el propio, no debería ser posible seguirse o dejar de seguirse a uno mismo, ya que carece de sentido.

Por otro lado, si el perfil consultado es el propio, las funcionalidades cambian. En este caso, se debe agregar un botón adicional, visible únicamente desde la consulta del propio perfil. Este botón servirá exclusivamente para eliminar seguidores en caso de ser necesario, ya que, en ocasiones, puede no ser deseable que ciertas personas sigan el perfil. De esta manera, el usuario tiene la opción de eliminarlos.

La implementación de toda esta lógica convierte a la pantalla en una herramienta realmente compleja, ya que debe ser única para ambos casos mencionados. Para lograr esto, se replica la acción de seguir o dejar de seguir utilizada en los perfiles, en resumen, añadiendo o eliminando una identificación de usuario en el campo follow o following. Se le añade una condición que detecta el ID del perfil propio: si el ID aparece entre los seguidores o seguidos, entonces el botón no será visible. Adicionalmente, el botón para eliminar seguidores, exclusivo para el usuario autenticado, tendrá una condición que permitirá su aparición únicamente si el identificador de la página coincide con el del usuario autenticado. Esto permite eliminar seguidores propios, pero no los de otras personas.

Esta funcionalidad se ilustra en las siguientes figuras:



**Figura 53. Vista de seguidores y seguidos perfiles de usuario diferente.**  
Fuente: El Autor.



**Figura 54. Vista de seguidores y seguidos perfil usuario autenticado.**  
Fuente: El Autor.

## 6.8.9 Pantallas de chat 1 a 1

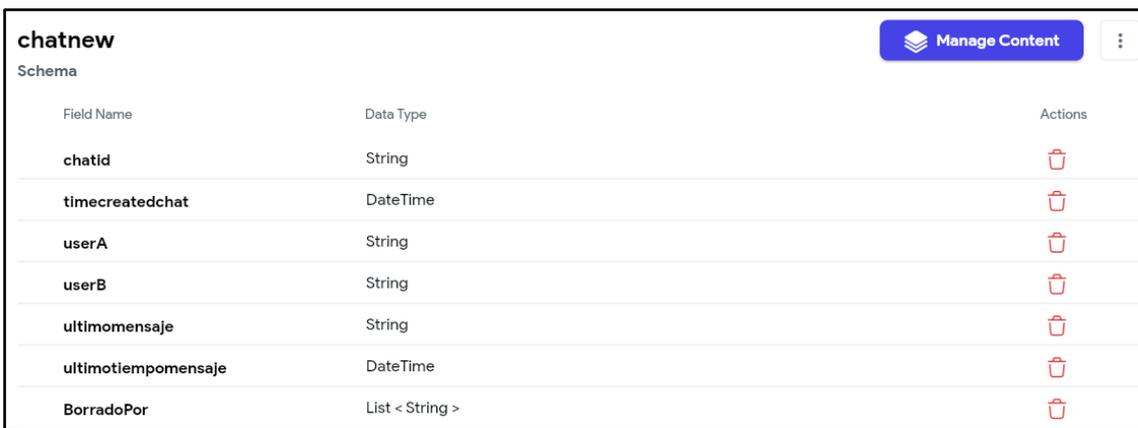
### 6.8.9.1 Colecciones dentro de Firebase para los chats 1 a 1

Para comenzar a diseñar la funcionalidad de los chats uno a uno, fue indispensable crear las colecciones necesarias para almacenar toda la información relevante. Se decidió establecer dos colecciones: una para los chats y otra para los mensajes correspondientes a cada chat.

La colección de chats debe registrar primero un identificador único para cada conversación, garantizando su singularidad. También debe guardar la fecha de creación, así como los identificadores de los dos usuarios involucrados: el usuario A, quien inicia la conversación, y

el usuario B, quien recibe el mensaje. Esta estructura es esencial para poder mostrar los chats en los que el usuario participa en la pantalla de listado de chats. Además de estos datos, la colección almacenará el último mensaje enviado y la hora de su envío. Asimismo, incluirá un campo adicional donde se registrará la identificación del usuario que elimine el chat, lo que permitirá que el chat desaparezca de la lista sin borrar el documento de la base de datos.

La estructura de la colección de chats se ilustra en la siguiente figura:



Field Name	Data Type	Actions
chatid	String	
timecreatedchat	DateTime	
userA	String	
userB	String	
ultimomensaje	String	
ultimotiempomensaje	DateTime	
BorradoPor	List < String >	

**Figura 55. Colección de chats 1 a 1 dentro de FlutterFlow y Firebase.**  
Fuente: El Autor.

Además de la colección de chats, fue necesario utilizar una segunda colección para almacenar los mensajes enviados dentro de cada conversación. Para ello, se creó la colección de mensajes. Esta colección se relacionará con la colección de chats mediante el identificador de la conversación, que será una combinación del ID del receptor del mensaje y el ID del remitente, garantizando así su unicidad.

Cada mensaje dentro de esta colección tendrá su propio identificador y un campo que registrará la identificación del remitente, sin importar si es el usuario A o B, ya que cada mensaje se almacena como un nuevo documento. Adicionalmente, se incluirá un campo para definir el tipo de mensaje, permitiendo el envío de textos, imágenes o archivos PDF. Esta diferenciación será útil para diseñar las burbujas de chat que mostrarán el contenido según su

tipo. Finalmente, se dispondrán campos específicos para almacenar el texto, la imagen o el PDF, según sea necesario.

La estructura de la colección de mensajes se ilustra en la siguiente figura:

Field Name	Data Type	Actions
chatid	String	
userID	String	
texto	String	
imagenes	List < Image Path >	
timemensaje	DateTime	
TipoMensaje	String	
idMensaje	String	
pdfUrl	String	

**Figura 56. Colección de mensajes de chats 1 a 1 dentro de FlutterFlow y Firebase.**  
Fuente: El Autor.

### 6.8.9.2 Pantallas de inicio de chat 1 a 1

Es el momento de adentrarse en la sección de chats. Esta área de la aplicación está compuesta por varias pantallas debido a su extensa funcionalidad. Las pantallas incluidas son:

- Pantalla de inicio de chat uno a uno
- Pantalla de espacio de chat
- Listado de conversaciones

En este apartado, se enfocará en la primera pantalla, encargada de iniciar una nueva conversación. Para acceder a esta pantalla, es necesario localizar el perfil de un usuario, ingresar a dicho perfil y seleccionar el botón de mensaje. Esta acción verificará si ya existe una referencia para el chat. Si la referencia no se encuentra creada, el usuario será dirigido a la pantalla de inicio de chat.

En la pantalla de inicio de chat, se mostrarán la fotografía y el nombre del usuario con quien se comenzará la conversación, así como los datos del usuario que inicia el chat. Esto asegura

que el usuario esté escribiendo a la persona correcta y mejora la presentación visual. Además, la pantalla incluirá un campo de texto para redactar el mensaje, acompañado de un botón que inicialmente estará inactivo y gris. Una vez que el botón cambie de color, permitirá enviar el mensaje.

Esta funcionalidad se detalla en la siguiente figura:



**Figura 57. Pantalla de chat inicial.**  
**Fuente: El Autor.**

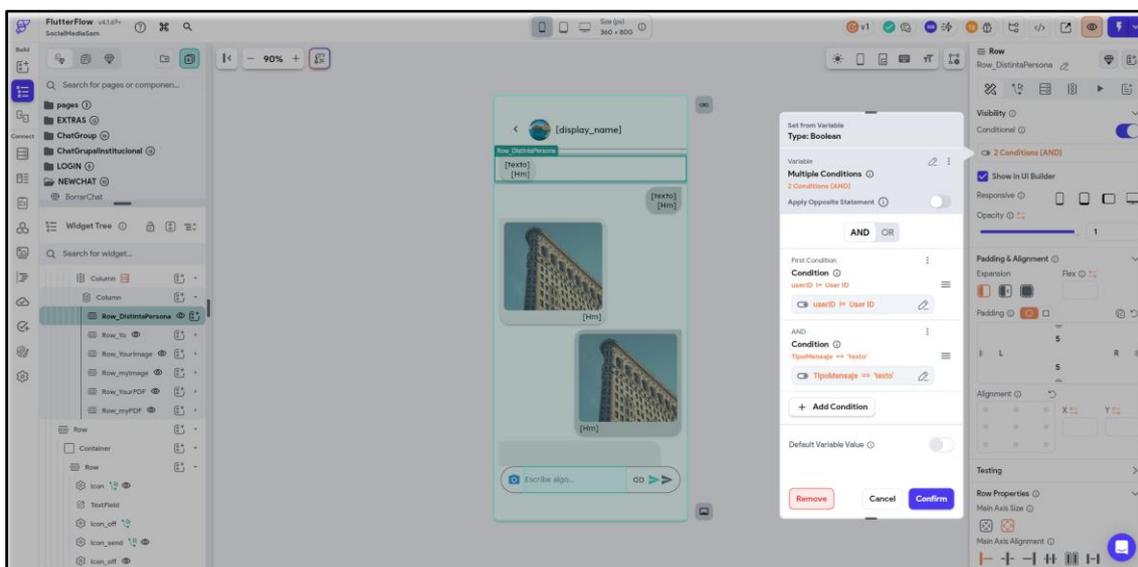
### 6.8.9.3 Pantalla de espacio de chat

Una vez se envíe el primer mensaje y se haya creado el chat, el usuario será redirigido a la pantalla de espacio de chat, donde se mostrarán los mensajes que pueden ser de tipo texto, imagen o PDF. Esta pantalla estará dividida en tres secciones: la primera mostrará el nombre

del usuario con el que se está entablando la conversación; la segunda sección será el espacio donde se visualizarán los mensajes; y la tercera será el campo de entrada para enviar nuevos mensajes.

Los mensajes aparecerán en la pantalla con ciertas condiciones. En primer lugar, se determinará quién envía el mensaje: los mensajes que se muestren a la izquierda serán enviados por el otro usuario, mientras que los mensajes a la derecha serán enviados por el usuario autenticado. Adicionalmente, cada burbuja de mensaje tendrá una condición que identificará el tipo de mensaje, es decir, si es un mensaje de texto, aparecerá en una burbuja de texto simple. Si se trata de una imagen o un archivo PDF, aparecerán en sus respectivas burbujas según corresponda.

Esta configuración se ilustra en la siguiente figura:



**Figura 58. Configuración burbujas de chat 1 a 1.**  
Fuente: El Autor.

Finalmente, la tercera sección de esta pantalla será el espacio destinado para escribir el mensaje. Esta sección incluirá un ícono de cámara, que permitirá enviar una imagen al chat seleccionándola de la galería y enviándola como mensaje. Este mensaje se almacenará como tipo imagen, mostrando la burbuja correspondiente.

Además, se podrá enviar un mensaje de texto, como se explicó anteriormente, con la diferencia de que, en lugar de crear un nuevo chat, se actualizará el último mensaje enviado dentro de la colección y se creará el documento correspondiente al mensaje.

También habrá un ícono de clip, que permitirá subir y enviar archivos PDF al chat. Estos documentos se enviarán como tipo PDF, permitiendo su visualización dentro del chat o en forma de enlace en el mismo mensaje. Al seleccionar el enlace, el usuario será dirigido directamente a su navegador preferido.

La estructura detallada de la pestaña de espacio de chat se ilustra en la siguiente figura:

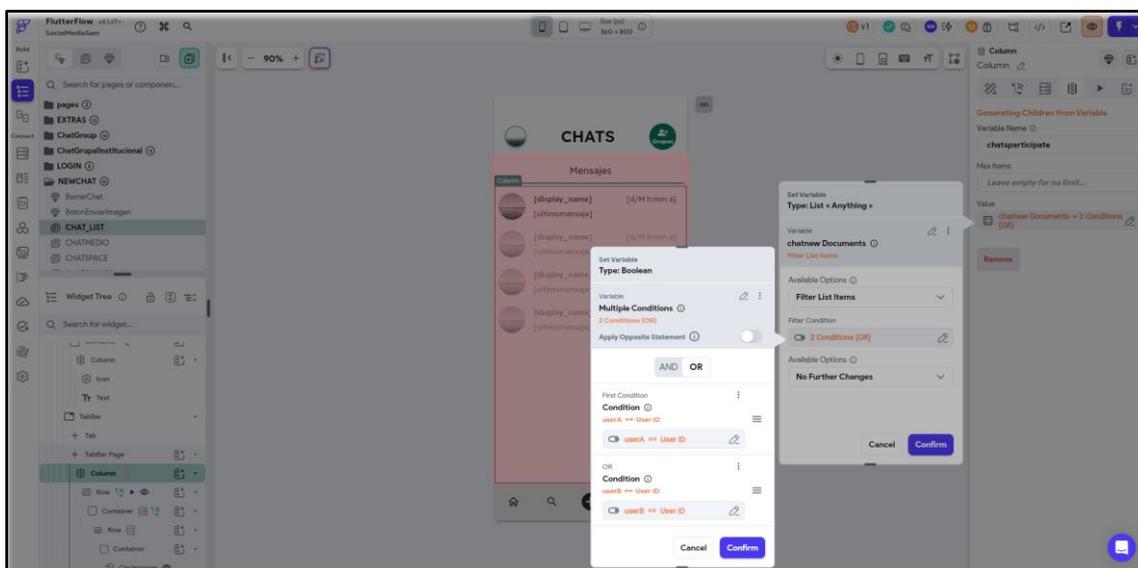


**Figura 59. Espacio de chat 1 a 1.**  
**Fuente: El Autor.**

#### 6.8.9.4 Pantalla de listado de chats 1 a 1

La pantalla de listado de chats es un componente crucial de la aplicación, ya que aquí se muestran los chats en los que el usuario participa. Para lograr esto, se realiza una consulta a la colección de chats, presentándolos en forma de lista. Los contenedores de cada chat estarán condicionados a aparecer únicamente si el usuario autenticado se encuentra registrado como usuario A o usuario B en la conversación. De esta manera, solo se mostrarán los chats en los que el usuario está involucrado. Al seleccionar un contenedor, el usuario será dirigido a la pantalla de espacio de chat para continuar con la conversación.

Esta configuración se ilustra en la siguiente figura:



**Figura 60. Condiciones en el listado de chat 1 a 1.**  
Fuente: El Autor.

### 6.8.10 Pantallas de chat grupal general

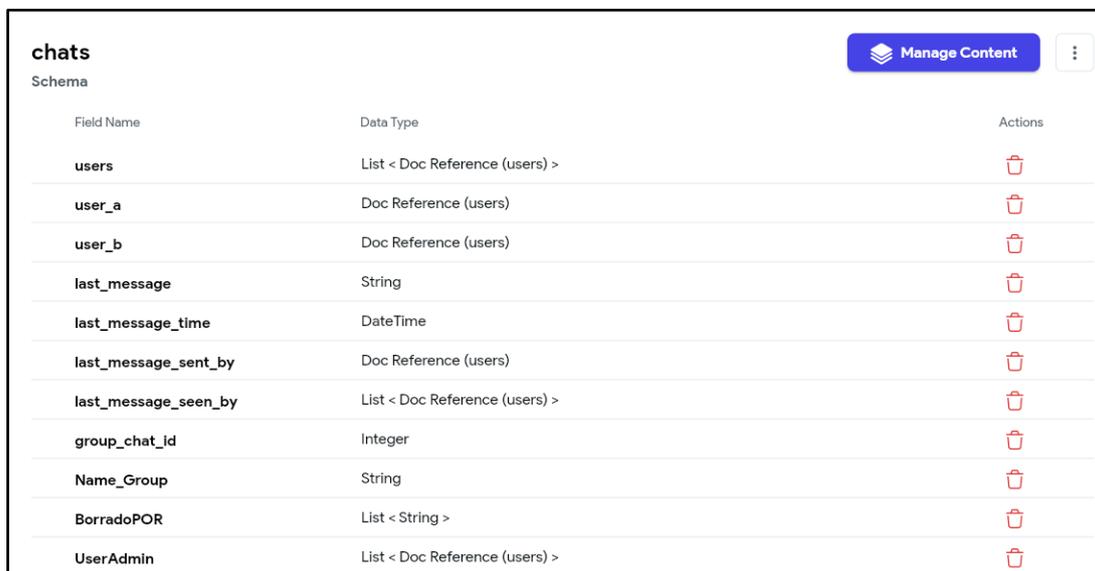
#### 6.8.10.1 Colecciones dentro de Firebase para los chats grupales generales

Al igual que en los chats uno a uno, se decidió implementar una lógica similar utilizando dos colecciones: una para almacenar los chats grupales y otra para los mensajes correspondientes.

La colección de chats grupales contendrá un identificador único, generado a partir de un número aleatorio, ya que combinar textos no sería factible debido al gran número de usuarios. Esta colección también incluirá una lista de usuarios referenciados desde la colección de "users", permitiendo así almacenar múltiples usuarios en un solo chat. Se repetirá el proceso de usuario A y usuario B, teniendo en cuenta que el usuario A será el creador del grupo y el usuario B el primer usuario invitado.

Además, se añadirá una lista de usuarios administradores, donde se ingresarán los identificadores de los usuarios que podrán ser nombrados como administradores del grupo. Habrá un campo que contendrá el nombre del grupo, y también una lista para almacenar los identificadores de los usuarios que decidan eliminar el grupo de su lista de chats, aparte contendrá campos para el último mensaje, para la hora de envió del último mensaje y para saber quién fue el que envió el mensaje.

La estructura de la colección de chats grupales se ilustra en la siguiente figura:



Field Name	Data Type	Actions
users	List < Doc Reference (users) >	
user_a	Doc Reference (users)	
user_b	Doc Reference (users)	
last_message	String	
last_message_time	DateTime	
last_message_sent_by	Doc Reference (users)	
last_message_seen_by	List < Doc Reference (users) >	
group_chat_id	Integer	
Name_Group	String	
BorradoPOR	List < String >	
UserAdmin	List < Doc Reference (users) >	

**Figura 61. Colección de chats grupales general dentro de FlutterFlow y Firebase.**  
**Fuente: El Autor.**

Para complementar la colección de chats grupales, se utilizará una colección dedicada a los mensajes. Esta colección se relacionará mediante una referencia directa a la colección de chats, ya que al utilizar un identificador aleatorio no es posible establecer una relación sin dicha referencia, a diferencia del chat uno a uno donde el identificador podía ser una combinación de textos que siempre sería igual.

En esta colección también habrá un campo para hacer referencia al usuario dentro del grupo que envíe el mensaje. Adicionalmente, se incluirá un campo para guardar el contenido del mensaje y el tiempo en que fue enviado. Asimismo, habrá campos dedicados para almacenar imágenes, videos o archivos PDF.

La estructura de la colección de mensajes se detalla en la siguiente imagen:

Field Name	Data Type	Actions
user	Doc Reference (users)	
chat	Doc Reference (chats)	
text	String	
timestamp	DateTime	
image	Image Path	
video	Video Path	
PDFURL	String	

**Figura 62. Colección de mensajes de chats grupales general dentro de FlutterFlow y Firebase.**  
Fuente: El Autor.

### 6.8.10.2 Pantalla de listado de grupos generales.

En este tipo de grupos, la primera pantalla será un listado que contendrá, en la esquina superior derecha, un ícono (+). Este ícono proporcionará navegación a la página de creación de grupos, que se explorará en el siguiente punto.

La pantalla de listado mostrará un catálogo de grupos y posee una característica especial: los contenedores de chats tienen una doble visualización. Un grupo puede crearse con dos o más personas, lo que implica que puede consistir únicamente en el creador y otra persona, similar a un chat uno a uno, o incluir varios miembros. Por esta razón, se decidió implementar una función que diferencie estos grupos. Si el grupo tiene solo un invitado, se mostrará únicamente el nombre de ese integrante y su foto de perfil, emulando un chat uno a uno. Sin embargo, si el grupo tiene más integrantes, se mostrará el nombre del grupo junto con las fotos de perfil de los dos últimos usuarios añadidos al grupo. Adicionalmente, se exhibirá el último mensaje enviado junto con su respectiva hora de envío.

Esta funcionalidad se logró mediante el uso de condicionales (si/entonces/si no), de manera que, si el chat contenía solo un invitado, se mostraría la información de dicho invitado; en caso contrario, se desplegaría la información general del grupo.

La pantalla de listado se presenta en la siguiente figura:



**Figura 63. Pantalla de listado de chats grupales generales.**  
Fuente: El Autor.

### 6.8.10.3 Pantalla para agregar miembros al chat grupal general

En la pantalla de listado, se encuentra un ícono (+) que navega directamente a la página de agregar miembros. Esta página desempeña varias funciones esenciales dentro de este tipo de grupos. Inicialmente, está diseñada para la creación de nuevos grupos, por lo que lo primero que se encuentra es un mensaje solicitando al creador que seleccione dos o más personas.

Además, la página incluye un apartado para la búsqueda de usuarios, con diversas opciones de filtrado, tales como:

- Nombre
- Correo electrónico
- Número de celular
- Carrera
- Nivel
- Roles

Como se muestra en la siguiente figura:



**Figura 64. Tipos de búsqueda en los grupos generales.**  
Fuente: El Autor.

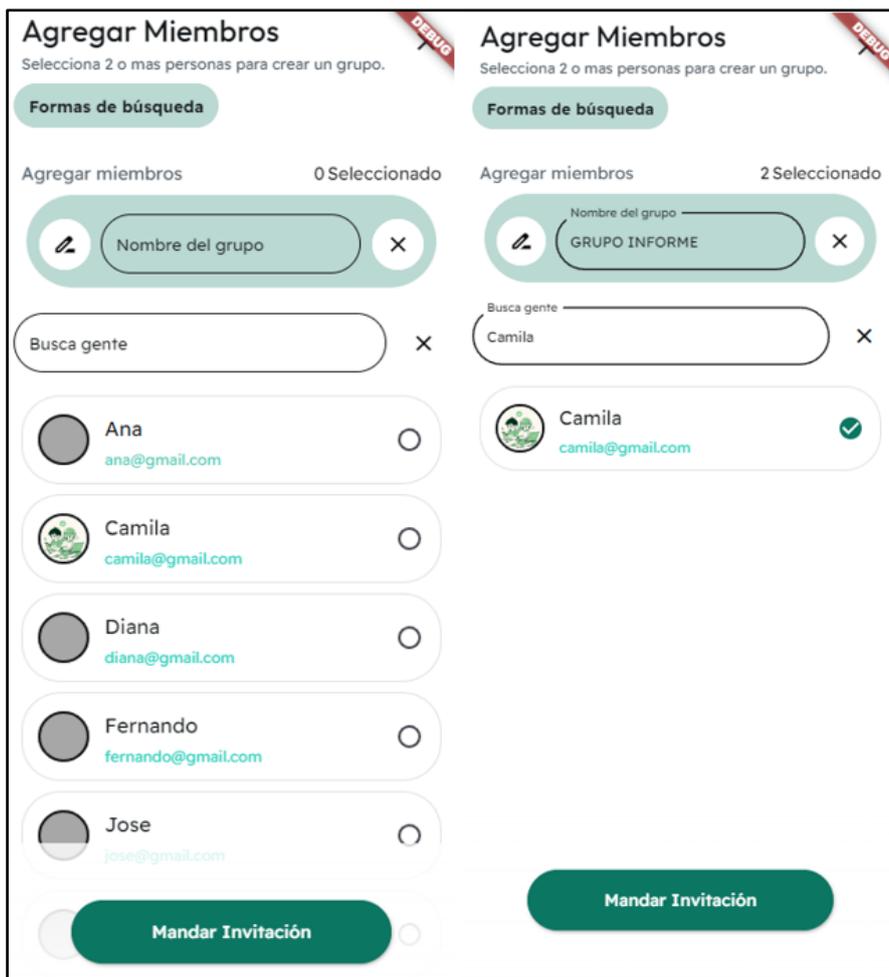
Seguido a esto, se encuentra un campo donde se puede ingresar un nombre para el grupo, acompañado de un ícono (x) que limpia automáticamente el campo de texto en caso de que se decida cambiar el nombre antes de crearlo. Luego, se presenta un buscador de perfiles cuya lógica es similar, por no decir la misma, a la utilizada en la pantalla de búsqueda de perfiles. La única diferencia es que, en lugar de redirigir al perfil de la persona, el usuario será seleccionado mediante una casilla de verificación, ya que se está eligiendo a este usuario para invitarlo al grupo.

El buscador incluye una lista general de usuarios y otra que responde al estado de la variable "Buscando", actualizando rápidamente la lista para mostrar a aquellos individuos que coincidan con la búsqueda.

Finalmente, se dispone de un botón con el mensaje "Mandar invitación", el cual crea un grupo en la colección de chats grupales y agrega a todos los usuarios invitados desde esta pantalla. Este botón también navega automáticamente al espacio del chat, donde se verán los mensajes enviados. Esta pantalla se explorará en el siguiente punto.

Para recapitular, esta pantalla tiene funciones adicionales destinadas a reutilizarla en caso de querer agregar más miembros a un grupo ya creado. En este caso, el botón, en lugar de mandar una invitación y crear un grupo, verificará si la referencia del grupo ya existe. Si esto es verdadero, solo actualizará la lista de miembros para agregar o quitar un integrante. Es decir, si se selecciona un usuario no marcado en la casilla de verificación, se agregará; si se desmarca la casilla de un usuario ya perteneciente, el botón entenderá que este usuario ha sido removido del grupo. De esta manera, se reutiliza la misma pantalla tanto para la creación como para la actualización del grupo. El funcionamiento se mostrará más adelante.

La pantalla de agregar miembros se muestra en la siguiente figura:



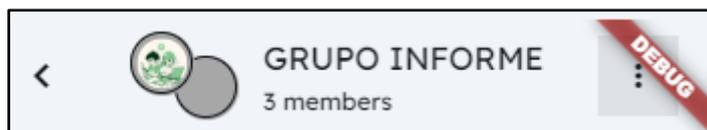
**Figura 65. Pantalla Agregar miembros a grupos generales.**  
Fuente: El Autor.

#### 6.8.10.4 Pantalla espacio del chat grupal general

Una vez creado el chat, se dirige a la pantalla donde se mostrarán los mensajes. Esta pantalla se compone de varios elementos, es decir, a diferencia de otras pantallas de la aplicación, esta se construye a partir de diferentes componentes que conforman una única vista.

Estos componentes se dividen en la sección de la app bar, que mostrará inicialmente el nombre del grupo junto a las fotos de perfil de los dos últimos usuarios añadidos. Asimismo, indicará el número de participantes del grupo. Además, contiene un ícono de tres puntos, cuya función es abrir un menú de configuración del grupo. Este menú se explorará más adelante.

El componente del app bar se muestra en la siguiente figura:



**Figura 66. Componente inicial espacio de chat.**  
Fuente: El Autor.

El siguiente componente será una portada de chat vacío, diseñada para dar la bienvenida a un chat de la institución. Al iniciar el grupo, este chat estará completamente vacío, y con esta portada se busca personalizar la aplicación, dándole un toque distintivo y propio del Instituto Superior Tecnológico Sudamericano. No obstante, esta área se ubicará debajo del componente que contiene las burbujas de mensajes, ya que una vez se envíe el primer mensaje, el componente de bienvenida está diseñado para ocultarse permanentemente, cediendo su espacio a los mensajes.

La bienvenida a los grupos se visualiza en la siguiente figura:

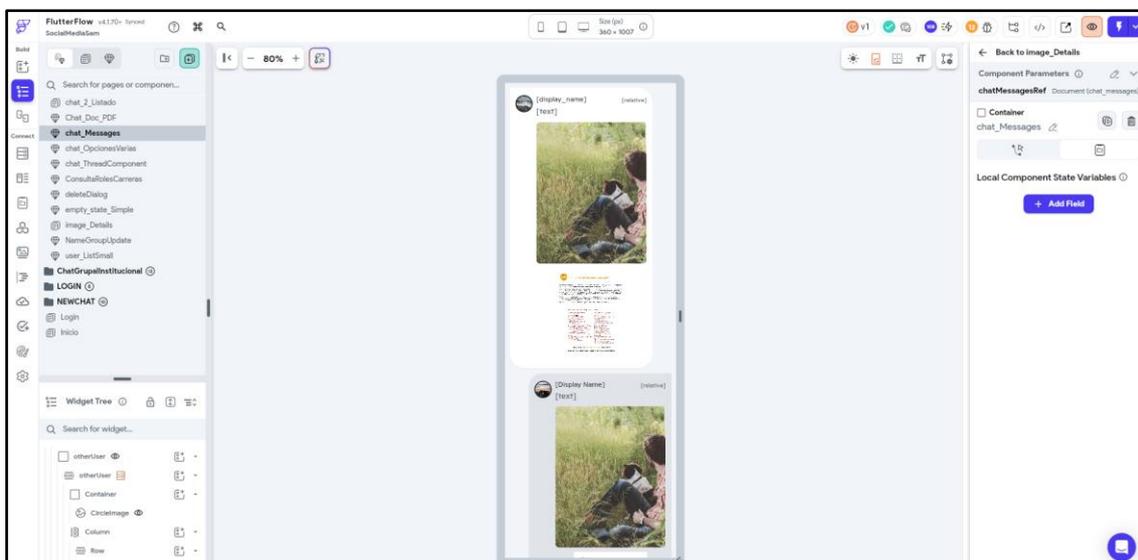


**Figura 67. Componente de bienvenida al grupo.**  
Fuente: El Autor.

Es momento de hablar del último componente que conforma esta página. Este componente, a su vez, estará compuesto por dos elementos adicionales. El primero será el componente que muestra los mensajes de los integrantes, diseñado para mostrar la información de la persona que envía el mensaje, incluso si es el mismo usuario autenticado. De esta manera, siempre se sabrá quién ha escrito o enviado contenido visual.

A diferencia del chat uno a uno, donde se hizo una burbuja separada para cada acción (texto, imagen y PDF), este componente simplifica las cosas al combinar los tres tipos de mensajes en una sola burbuja. Solo se mostrará el contenido enviado: si es texto, no aparecerán los contenedores de imágenes o PDF, y viceversa. Sin embargo, estas burbujas pueden contener tanto imágenes con texto como un enlace al PDF si se envía uno.

En resumen, se trata de un componente tres en uno que reemplazará la bienvenida inicial. Las burbujas se observan en la siguiente figura:



**Figura 68. Componente burbujas de chat grupales.**  
Fuente: El Autor.

El segundo componente contendrá el espacio para escribir el mensaje y enviarlo. Al igual que en el chat uno a uno, incluirá un ícono para adjuntar archivos PDF y otro para enviar

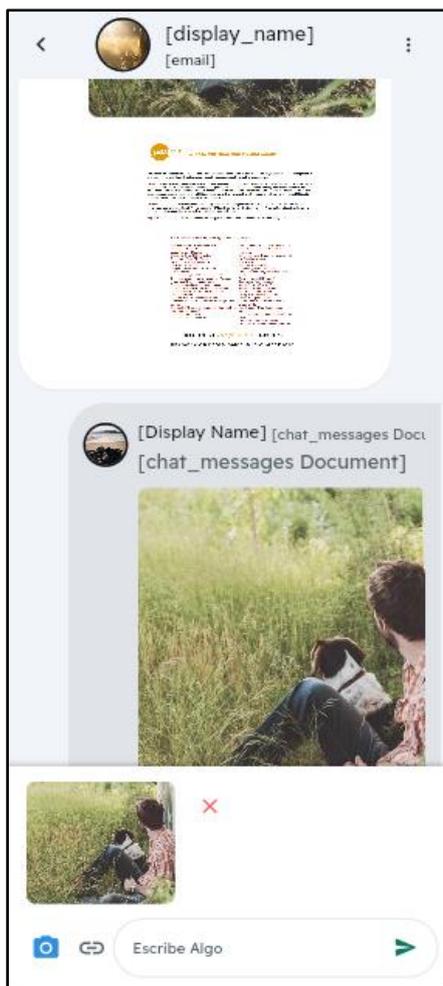
imágenes. Este componente también tendrá un botón de envío que solo aparecerá si el campo de texto contiene un mensaje; de lo contrario, no permitirá enviar mensajes. Este botón creará un nuevo documento en la colección de mensajes y actualizará la referencia del chat grupal, ya que debe actualizar tanto el último mensaje como la hora del último envío.

El último componente se muestra completo en la siguiente figura:



**Figura 69. Componente final pantalla espacio de chat.**  
**Fuente: El Autor.**

Finalmente, la pantalla de espacio de chat armada queda completamente se muestra en la siguiente figura:



**Figura 70. Pantalla de espacio de chat.**  
**Fuente: El Autor.**

#### 6.8.10.5 Menú de configuración de chat grupal general

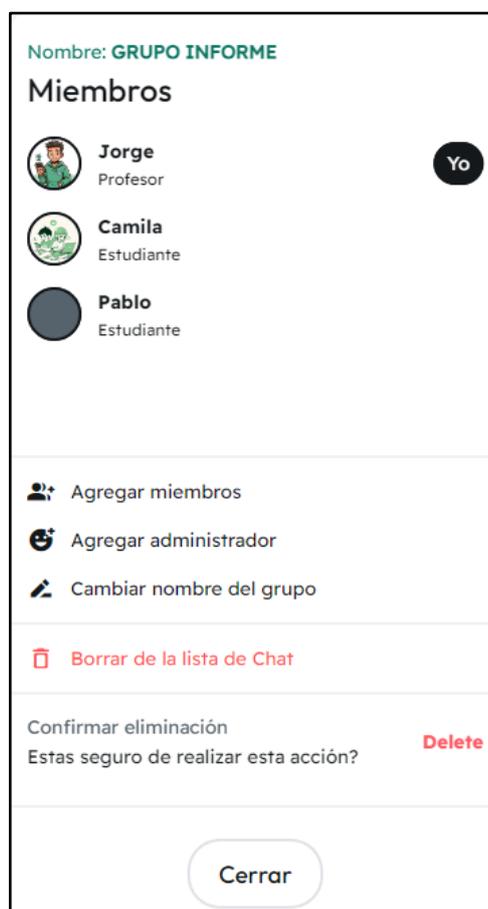
El menú de configuración tendrá la misma estructura dinámica que la pantalla del espacio de chat, ya que también se compone de varios elementos que conforman un componente general. El primer componente mostrará el nombre del grupo y los miembros. El segundo componente ofrecerá una serie de configuraciones para el grupo, entre ellas:

- **Agregar miembros:** Esta funcionalidad reutiliza la pantalla de agregar miembros.
- **Agregar administrador:** Esta opción permite añadir los identificadores de algunos usuarios seleccionados a una lista de administradores, otorgándoles acceso a las configuraciones del menú.
- **Cambiar nombre del grupo:** Permite modificar el nombre del grupo.

Cabe destacar que este segundo componente solo será visible si el usuario tiene el rol de administrador del grupo. En caso contrario, solo se mostrará el nombre del grupo y los miembros.

Finalmente, se encuentra un componente que permite borrar el chat de la lista de chats, solicitando confirmación del usuario antes de proceder.

El menú completo se muestra en la siguiente figura:



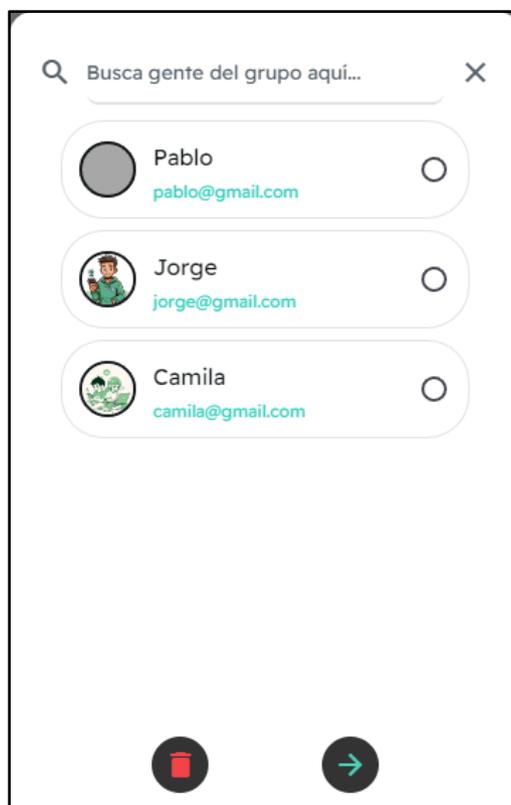
**Figura 71. Menú de configuración grupos generales.**  
**Fuente: El Autor.**

#### 6.8.10.6 Agregar administrador al grupo

Para agregar o quitar un administrador de grupo, se utiliza una pantalla similar al buscador de perfiles. Esta pantalla muestra una lista general de los miembros del grupo y otra lista que reacciona al estado de búsqueda. Se han agregado dos tipos de botones: el primero, de color

oscuro y con un ícono de basurero, para quitar administradores; y el segundo, con un ícono de flecha y de color verde, para agregar la identificación del nuevo administrador.

La sub pantalla se observa en la siguiente figura:



**Figura 72. Agregar administradores al grupo.**  
Fuente: El Autor.

### **6.8.11 Pantallas de chat grupal institucional**

El chat grupal institucional constituye un espacio destinado a agrupar usuarios que cumplan con criterios específicos, como carrera(s), nivel(es) y sección(es) seleccionadas. Esta funcionalidad agiliza la creación y búsqueda de grupos al reunir exclusivamente a los usuarios pertinentes. La creación de estos grupos está reservada a usuarios con roles distintos al de Estudiante. No obstante, todos los usuarios podrán acceder al listado de estos grupos, independientemente de su rol.

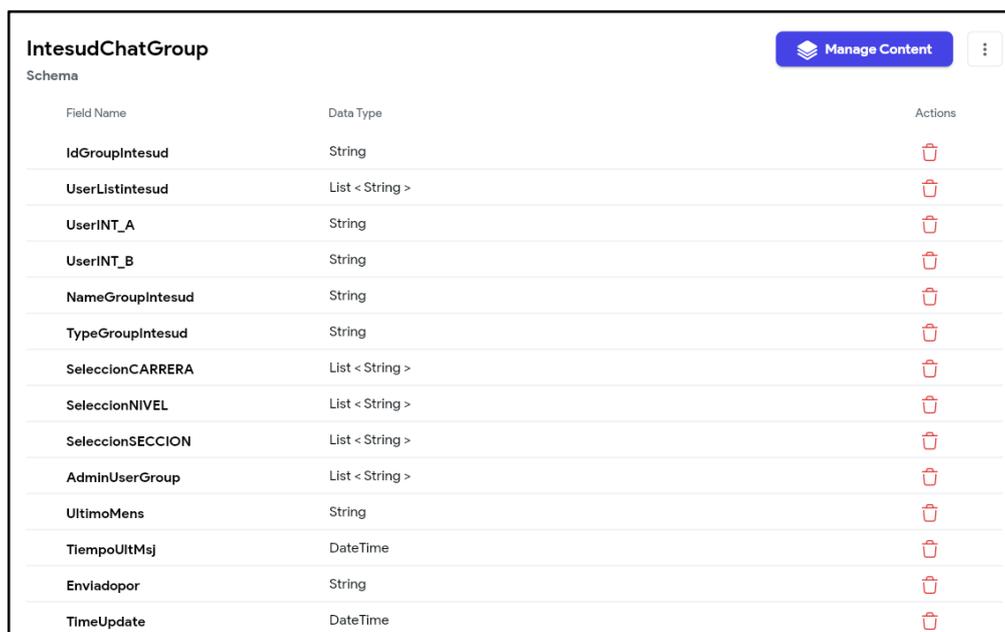
La lógica empleada en estos grupos combina características tanto de los chats 1 a 1 en estilo y funcionalidad, como de los chats grupales generales que permiten la participación de más de dos personas. Esta combinación se basa en las condiciones especificadas en la pantalla de creación del grupo, las cuales se detallarán en el siguiente punto.

#### 6.8.11.1 Colecciones dentro de Firebase para los chats grupales institucionales.

Las colecciones necesarias para implementar este tipo de chat institucional seguirán un patrón similar a las colecciones anteriores. Se tendrán dos colecciones: una para los chats y otra para los mensajes.

La colección de chats se encargará de guardar el identificador único del grupo, el nombre del grupo y una lista de usuarios participantes. También se incluirán los usuarios A y B, el último mensaje junto con la hora de envío y la información de quien envió el último mensaje. Además, se agregarán tres listas que almacenarán las carreras, los niveles de semestre y las secciones, lo que facilitará el filtrado de los usuarios que deben pertenecer al grupo.

La colección de chats de grupos institucionales se muestra en la siguiente figura:

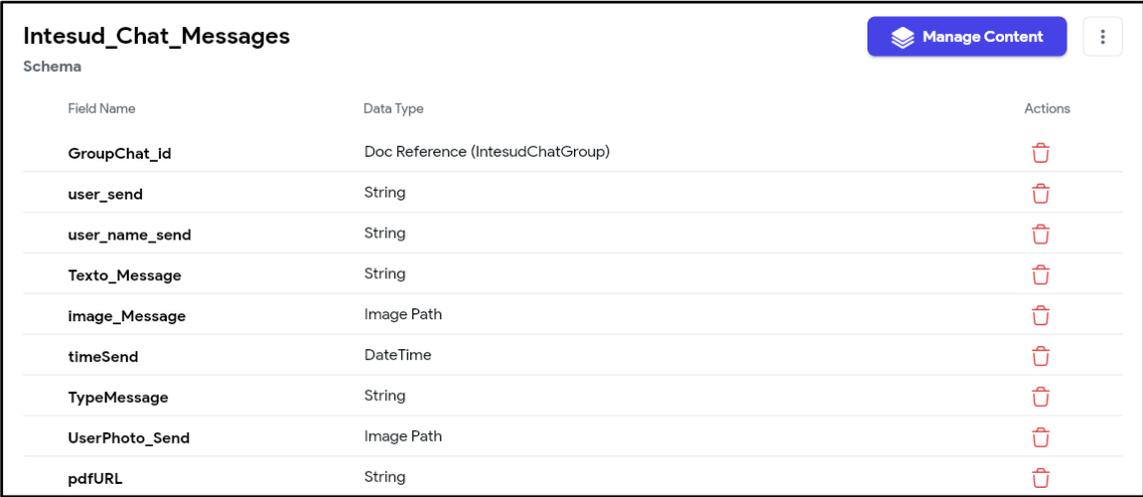


Field Name	Data Type	Actions
IdGroupIntesud	String	
UserListIntesud	List < String >	
UserINT_A	String	
UserINT_B	String	
NameGroupIntesud	String	
TypeGroupIntesud	String	
SeleccionCARRERA	List < String >	
SeleccionNIVEL	List < String >	
SeleccionSECCION	List < String >	
AdminUserGroup	List < String >	
UltimoMens	String	
TiempoUltMsj	DateTime	
EnviadoPor	String	
TimeUpdate	DateTime	

**Figura 73. Colección de chats grupales institucionales dentro de FlutterFlow y Firebase.**  
Fuente: El Autor.

La siguiente colección almacenará los mensajes del chat y se relacionará directamente con la colección de chats mediante una referencia. Además, guardará la referencia del usuario que envió el mensaje, su nombre de usuario, el texto del mensaje, y en caso de ser una imagen o un PDF, también se incluirán estos archivos. Asimismo, se registrará el tiempo de envío y la foto del usuario que envió el mensaje.

La colección de mensajes se muestra en la siguiente figura:



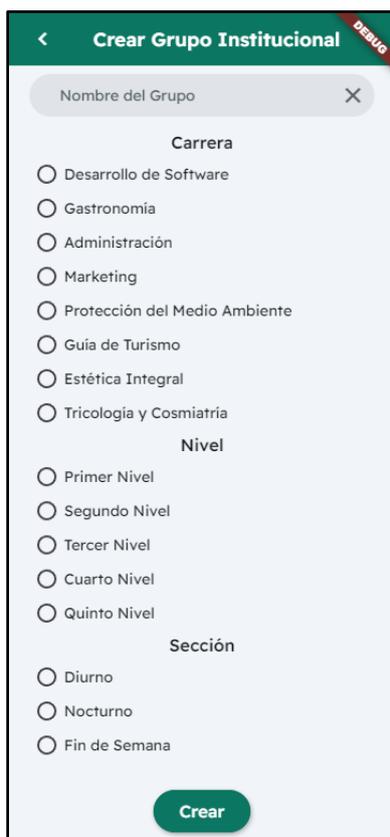
Field Name	Data Type	Actions
GroupChat_Id	Doc Reference (IntesudChatGroup)	
user_send	String	
user_name_send	String	
Texto_Message	String	
image_Message	Image Path	
timeSend	DateTime	
TypeMessage	String	
UserPhoto_Send	Image Path	
pdfURL	String	

**Figura 74. Colección de mensajes de chats grupales institucionales dentro de FlutterFlow y Firebase.**  
Fuente: El Autor.

### 6.8.11.2 Pantalla de creación de grupo institucional

La pantalla de creación del grupo incluye un apartado para nombrar el grupo y tres secciones para seleccionar la carrera, el nivel y la sección correspondientes. Esta pantalla utiliza casillas de verificación que contienen las ocho carreras ofrecidas por el instituto, incluyendo los cinco niveles, así como la opción "alumnis". También se incluyen las secciones: diurna, nocturna y fines de semana disponibles en la institución. Además de estas opciones seleccionables, se ha agregado un botón que permite crear un nuevo chat con estas características. Al hacer clic en este botón, el grupo se crea de inmediato y redirige al usuario creador al listado de chats, donde se puede visualizar el chat recién creado. La pestaña de listado se discutirá más adelante.

La pantalla de creación se observa en la siguiente imagen:



**Figura 75. Pantalla de creación de grupo institucional.**  
**Fuente: El Autor.**

### 6.8.11.3 Pantalla de listado de chats grupales institucionales

Al igual que las demás pantallas de listado, esta pantalla mostrará directamente el nombre del grupo, acompañado de una imagen predefinida del instituto para reforzar la identidad de la aplicación. Adicionalmente, se presentará el tiempo del último mensaje y el contenido de dicho mensaje. Al seleccionar un chat, el usuario será redirigido automáticamente al espacio de chat de ese grupo.

Asimismo, se ha implementado un buscador de chats que sigue la misma lógica de los buscadores anteriores, pero con la diferencia de que, en lugar de buscar perfiles, filtra los chats grupales. Esta funcionalidad permite una búsqueda más rápida e intuitiva. La pantalla también incluye una lista general de chats y una lista que reacciona a los resultados de la búsqueda.

La pantalla de listado se muestra en la siguiente figura:



**Figura 76. Pantalla de listado de chats grupales institucionales.**  
**Fuente: El Autor.**

#### 6.8.11.4 Pantalla de espacio de chat institucional

Esta pantalla es una extensión de la interfaz de chat 1 a 1, con funciones adicionales. Como se mencionó anteriormente, este tipo de grupos es una combinación de otros tipos de chats. Inicialmente, la pantalla muestra el nombre del grupo junto con la foto de perfil del mismo. En esta sección se encuentra un botón con el signo (+), cuyo uso se explicará más adelante.

A continuación, se encuentra el espacio de mensajes, donde aparecen las burbujas de chat. Estas burbujas siguen ciertas condiciones: las burbujas a la izquierda corresponden a mensajes de remitentes distintos al usuario autenticado, mientras que las burbujas a la derecha son mensajes enviados por el usuario autenticado. Se decidió utilizar burbujas separadas por tipo

de mensaje en lugar de agruparlas, para mayor comodidad. Estas burbujas muestran información del usuario que envió el mensaje; las burbujas a la izquierda incluyen la foto de perfil y el nombre del usuario remitente, combinando así la estética de los chats 1 a 1 con la funcionalidad de los chats grupales, lo cual es necesario para identificar al remitente de cada mensaje.

Finalmente, la zona para escribir mensajes contiene íconos para enviar imágenes, PDFs y mensajes de texto. El botón de enviar tiene la función de crear un nuevo documento en la colección de mensajes de los grupos institucionales y actualizar la referencia en la colección de chats.

Esta pantalla se visualiza en la siguiente figura:

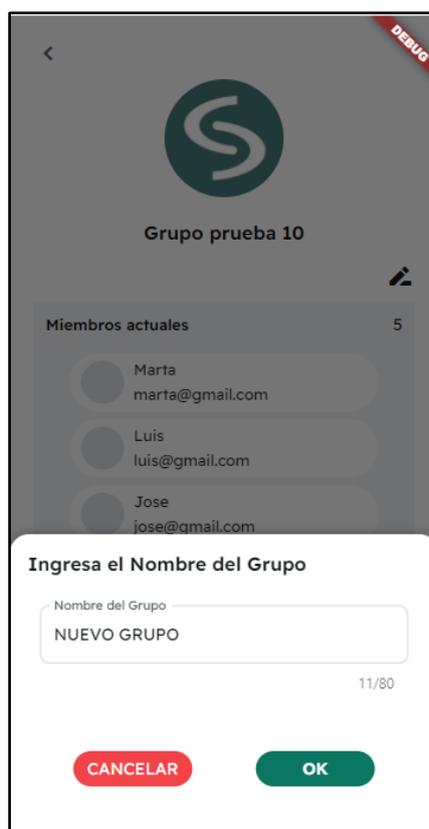


**Figura 77. Pantalla de espacio de chat institucional.**  
**Fuente: El Autor.**

### 6.8.11.5 Menú de configuración de grupos institucionales

Para acceder a esta página, es necesario seleccionar el nombre del grupo desde la pantalla principal del chat. Esto abrirá una nueva pantalla donde lo primero que se visualiza es el nombre del grupo, que incluye un ícono para cambiar el nombre. Al realizar este cambio, se actualiza la referencia del chat para guardar el nuevo nombre.

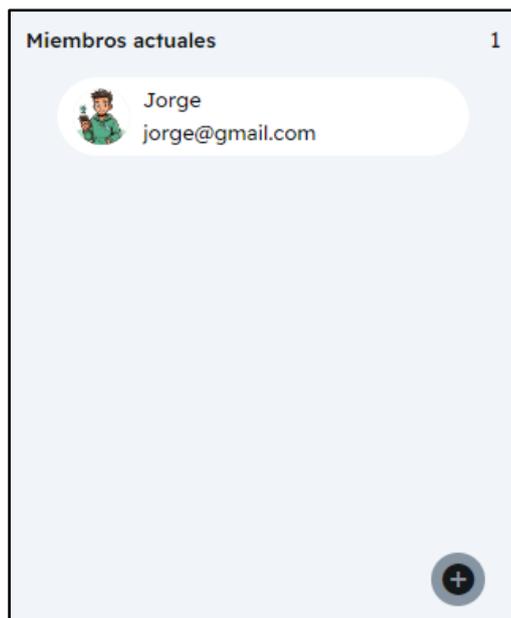
La sub pantalla de cambio de nombre se muestra en la siguiente figura:



**Figura 78. Cambio de nombre grupo institucional.**  
**Fuente: El Autor.**

Además, la página presenta un listado de los miembros del grupo. En caso de tratarse de un grupo nuevo, únicamente aparecerá el miembro creador. Esta sección incluye un ícono (+) similar al que se visualizó anteriormente en la página del espacio de chat.

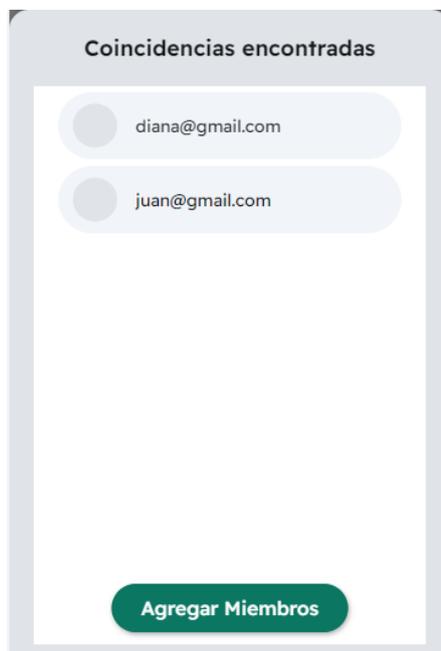
La zona de miembros se muestra en la siguiente figura:



**Figura 79. Zona de miembros chat institucionales.**  
**Fuente: El Autor.**

Este ícono tiene la función de buscar y encontrar a los usuarios que coincidan con las características del chat, tales como carrera, nivel y sección. Al presionar este botón, se abrirá una sub pantalla que filtrará a los usuarios según estos criterios.

La búsqueda de dichos perfiles se muestra en la siguiente figura:



**Figura 80. Miembros que coinciden con la selección.**  
**Fuente: El Autor.**

Como se puede observar, esta sub pantalla es similar a la interfaz para agregar o invitar miembros en otros chats grupales. En esta pantalla, se encuentra un botón cuya función es actualizar la lista de usuarios invitados mediante la inclusión de sus identificadores, añadiéndolos así al grupo. Además, este botón actualiza la fecha de adición de miembros.

La diferencia entre este ícono y el ícono presente en la página del espacio de chat radica en que el primero actualiza la fecha de ingreso de los miembros, mientras que el segundo no lo hace. Esto se debe a que los alumnos cambian de semestre continuamente, lo que requiere una actualización periódica de los integrantes del grupo para evitar que vean mensajes enviados en fechas anteriores.

Por ejemplo, si un grupo de personas utiliza el chat desde el 10 de julio de 2024 y se intercambian varios mensajes, estos mensajes desaparecerán en la siguiente actualización de fecha, por ejemplo, el 10 de enero de 2025. Esto permite que el chat pueda ser reutilizado por nuevos miembros, quienes iniciarán una nueva conversación.

Por otro lado, esta pantalla también presenta un listado de las carreras, niveles y secciones seleccionadas. Cada recuadro de estos elementos incluye un botón para actualizar los datos, permitiendo agregar o eliminar carreras, niveles o secciones del grupo. Esta funcionalidad se implementó para facilitar la inclusión de varias carreras en un solo grupo.

Estos recuadros se muestran en la siguiente figura:



**Figura 81. Selecciones carrera, nivel, sección.**  
**Fuente: El Autor.**

Las sub pantallas para actualizar estos campos son similares, diferenciándose únicamente en los datos que manejan. Cada una de estas pantallas contiene un botón que actualiza la referencia del chat, modificando las listas de los campos correspondientes. De esta manera, se pueden agregar o eliminar carreras, niveles o secciones. Una vez realizada la actualización, es necesario actualizar nuevamente los miembros del grupo, teniendo en cuenta si se desea actualizar la fecha de ingreso de los miembros para evitar la pérdida de mensajes anteriores.

Las pantallas de actualización de campos se muestran en la siguiente figura:

<input checked="" type="radio"/> Desarrollo de Software <input type="radio"/> Gastronomía <input type="radio"/> Administración <input type="radio"/> Marketing <input type="radio"/> Protección del Medio Ambiente <input type="radio"/> Guía de Turismo <input type="radio"/> Estética Integral <input type="radio"/> Tricología y Cosmiatría	<input checked="" type="radio"/> Primer Nivel <input checked="" type="radio"/> Segundo Nivel <input checked="" type="radio"/> Tercer Nivel <input type="radio"/> Cuarto Nivel <input type="radio"/> Quinto Nivel	<input checked="" type="radio"/> Diurno <input type="radio"/> Nocturno <input type="radio"/> Fin de Semana
<b>Actualizar Carreras</b>	<b>Actualizar Niveles</b>	<b>Actualizar Secciones</b>

**Figura 82. Actualización de datos de los grupos institucionales.**  
**Fuente: El Autor.**

### 6.8.12 Realizar test de la aplicación para detectar posibles errores.

Durante todo el desarrollo de la aplicación, se llevaron a cabo pruebas exhaustivas de falla y error para cada acción y funcionalidad implementada. Desde el inicio del proyecto, se realizaron testings minuciosos en cada etapa de implementación, asegurando que no se avanzara a la siguiente fase sin antes resolver cualquier problema detectado.

Estas pruebas se realizaron tanto en el tester de FlutterFlow como en dispositivos móviles reales, garantizando así que la aplicación funcione correctamente en diversos entornos y bajo diferentes condiciones. Este enfoque meticuloso permitió identificar y corregir fallos de manera oportuna, mejorando la estabilidad y la fiabilidad de la aplicación antes de su lanzamiento final.

## **6.9 Sistema de envío de notificaciones “push” e “in app”**

Las notificaciones son una forma de mantener a los usuarios activos dentro de una aplicación, pues sirven como recordatorios o alertas que te envía la propia red social, por lo que no pueden faltar.

Este proceso tuvo un sin número de desafíos que tuvieron que ser superados, pues el sistema de Firebase sufrió una actualización repentina el pasado día 20 de junio del 2024, lo que provocó que el proceso de implementación de notificaciones push fuera alterado y se volvió poco eficaz pues si bien estas notificaciones se guardaban en una nueva colección en la base de datos NO se mostraban en los dispositivos que descargaban la aplicación. Para resolver este problema se optó por implementar una API externa perteneciente a OneSignal.

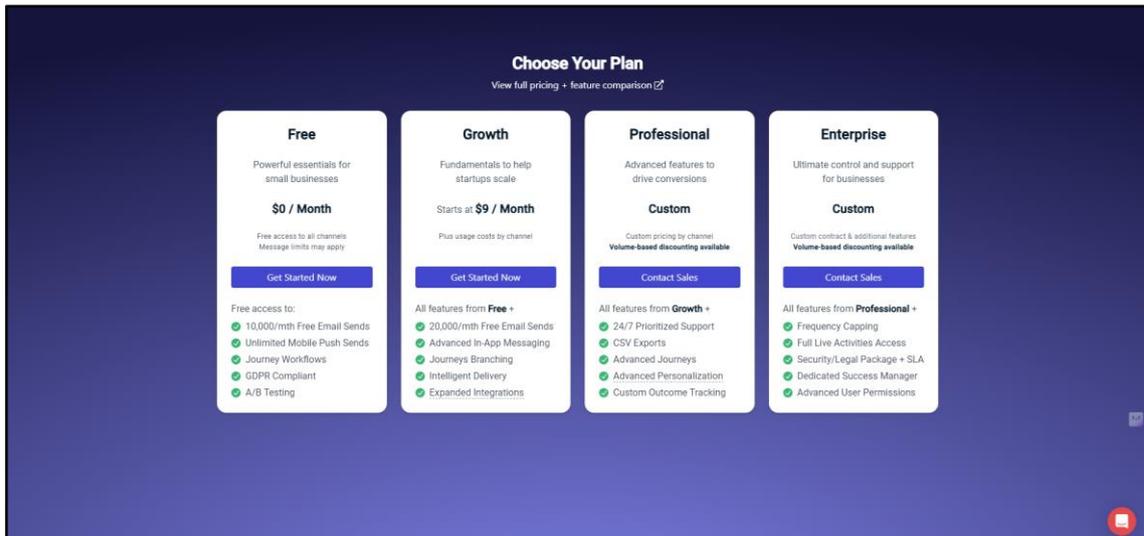
OneSignal es una plataforma de mensajería directa la cual permite el envío de notificaciones “push”, “in app”, correo electrónico y SMS, la cual permite a sus usuarios conectar esta API con su proyecto creado en Firebase y Flutter. Esta plataforma muestra las estadísticas de cada notificación que se envían, así otorgando un control total de estas notificaciones (OneSignal, s.f.).

### **6.9.1 Implementación de la API**

Para implementar este sistema, lo primero se debe ingresar en su página oficial y registrarse en su web se puede registrar con Google.

Pedirá elegir un plan para empezar a usar el servicio, en este caso se escoge el plan gratuito pues solo se usará para conectar la plataforma a la red social académica.

Los planes se muestran en la siguiente figura:



**Figura 83. Planes OneSignal.**  
Fuente: (OneSignal, s.f.)

Una vez escogido el plan, se solicita llenar un formulario para identificar la necesidad de ocupar esta plataforma.

El formulario se muestra en la siguiente figura:

**¡Bienvenido a OneSignal!**

**Cuéntanos acerca de ti**  
Responda algunas preguntas breves para que podamos personalizar su experiencia OneSignal.

Nombre de pila \*  Apellido \*

¿Qué categoría describe mejor su función? \*  
Seleccionar...

¿Para qué desea utilizar principalmente OneSignal? (Seleccione todas las opciones que correspondan) \*

- Impulsar las visitas al sitio web o la aplicación (por ejemplo, incorporación, actualizaciones y recordatorios)
- Enviar mensajes de marketing/promoción para generar ingresos
- Proporcionar actualizaciones de la empresa (por ejemplo, anuncios, actualizaciones de productos)
- Recuperar la participación de los usuarios inactivos / reducir la pérdida
- Enviar notificaciones transaccionales (por ejemplo, confirmaciones de pedidos)
- Otro

¿Para qué canales planes utilizar OneSignal? (Seleccione todas las opciones que correspondan) \*

- Correo electrónico
- Notificaciones push
- SMS
- Mensajes en la aplicación
- Viajes

Numero de teléfono (opcional)

[Continuar](#)

0 / 3 Pasos completados

- 1 **Acercas de ti**
- 2 Crear organización
- 3 Crear aplicación

**Figura 84. Formulario inicial en OneSignal.**  
Fuente: (OneSignal, s.f.)

Una vez completado el formulario en el punto 1 y 2 del proceso, es momento de crear la aplicación de OneSignal, para lo cual hay que darle un nombre que la identifique.

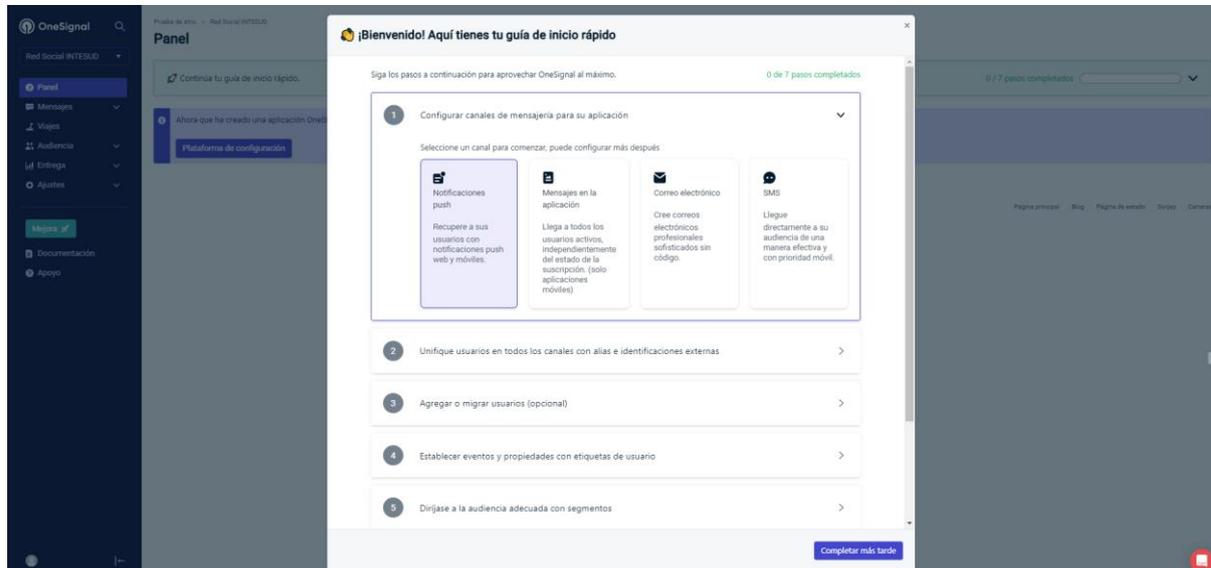
Este punto se visualiza en la figura a continuación:



**Figura 85. Nombrar aplicación de OneSignal.**  
Fuente: (OneSignal, s.f.)

Una vez dentro de nuestra cuenta de OneSignal aparecerá un recuadro, donde se elige la primera opción “Configurar canales de mensajería para tu aplicación” y se selecciona la opción de notificaciones push.

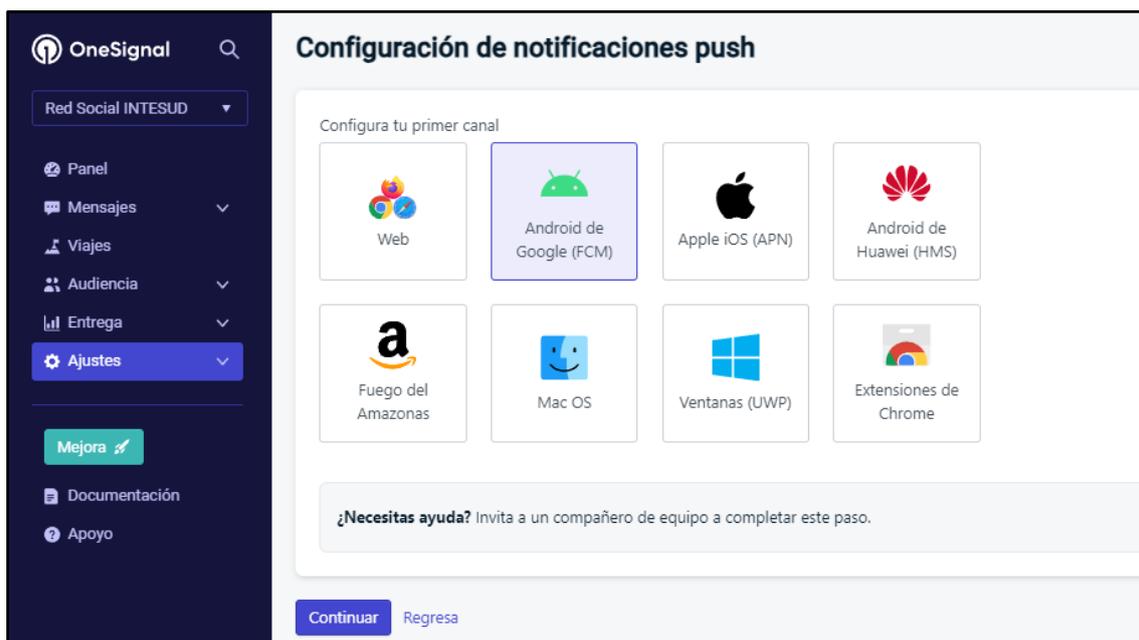
El recuadro se ilustra en la siguiente figura:



**Figura 86. Configuración OneSignal.**  
Fuente: (OneSignal, s.f.)

Redirige al usuario a configurar las notificaciones push, aquí se selecciona el canal para enviar las notificaciones, en este caso Android.

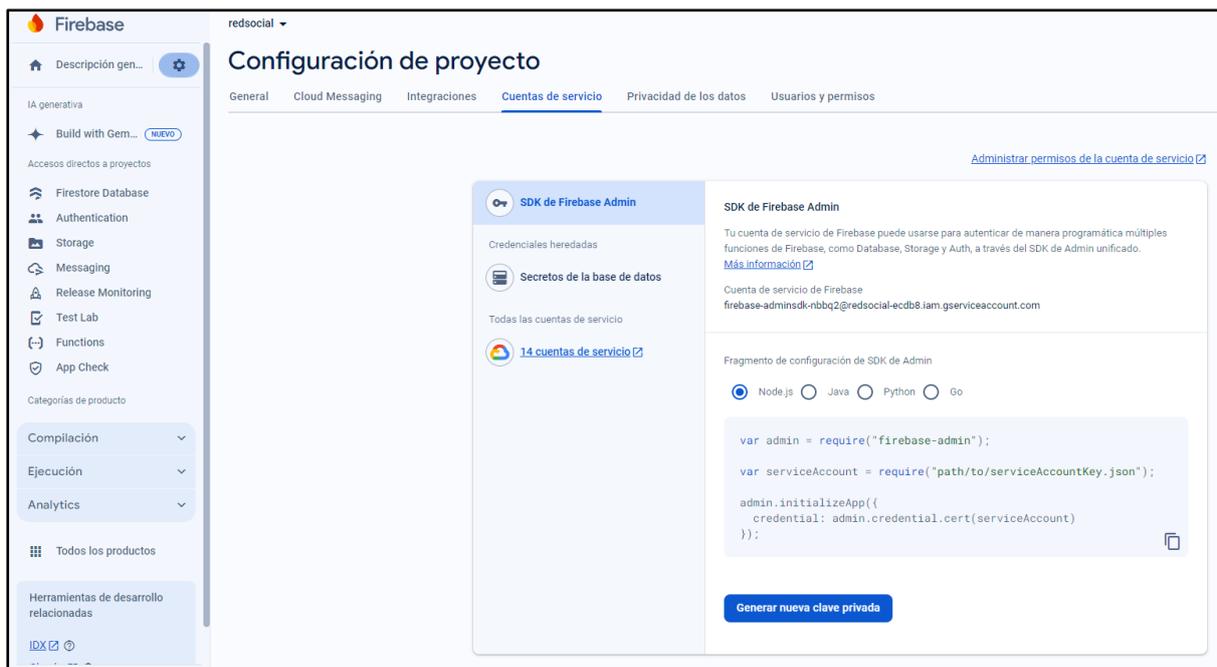
Como se muestra en la siguiente figura:



**Figura 87. Elección de canal de mensajería.**  
Fuente: (OneSignal, s.f.)

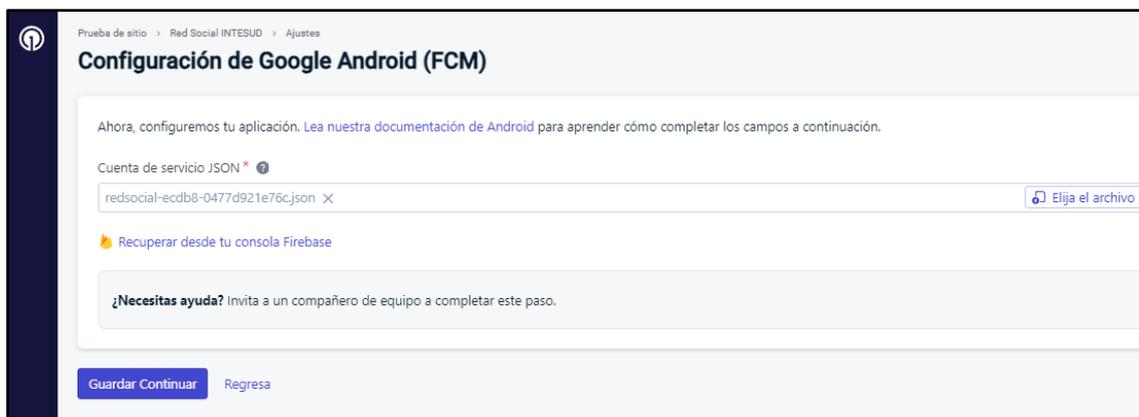
Para el siguiente punto es necesario descargar una llave de acceso a Firebase en formato .JSON, para esto se ingresa al proyecto creado en la base de datos, en descripción general del proyecto se ingresa en configuración del proyecto y posteriormente en cuentas de servicio para generar una nueva clave privada seleccionando el formato Node.js

Este apartado se muestra en esta figura:



**Figura 88. Descarga de nueva llave privada.**  
Fuente: El Autor.

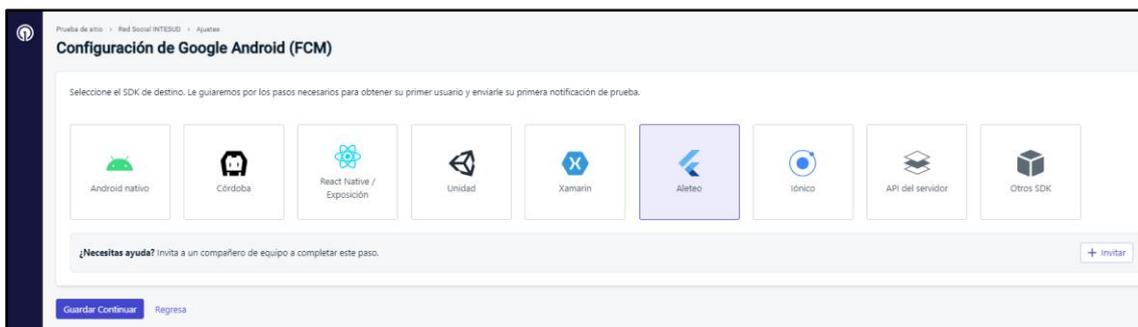
Dentro de OneSigna, se agrega la clave recién descargada como se muestra en la siguiente figura:



**Figura 89. Importar clave privada.**  
Fuente: El Autor.

Ya que se importó la clave privada, se selecciona el SDK de destino, en este caso Flutter, pues FlutterFlow es una extensión de Flutter.

Como se observa a continuación en la figura:



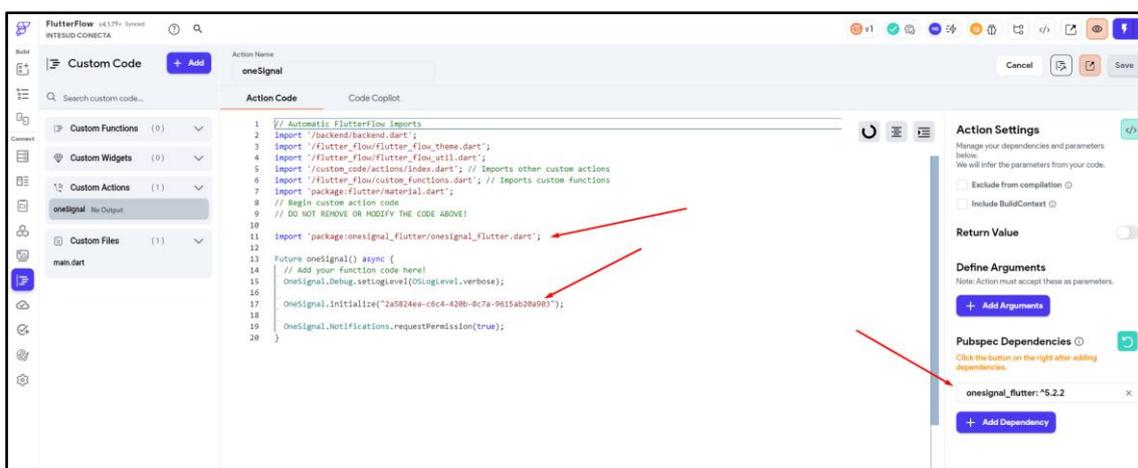
**Figura 90. Selección SDK de destino.**  
Fuente: El Autor.

Aquí entrega el id de la aplicación, lo cual ayudara a conectar la API al proyecto. Este identificador hay que copiarlo y guardarlo para usarlo más adelante.

### 6.9.2 Conectar API dentro de FlutterFlow

Dentro de FlutterFlow, se ingresa en la sección de “custom code”, y se debe crear una nueva acción, la cual se nombró como “onesignal” para una rápida detección, y se importó su última dependencia “onesignal\_flutter: ^5.2.2” hasta este momento, seguido a esto, inicializamos la API siguiendo la documentación de OneSignal la cual proporciona las líneas de código necesarias para este proceso, teniendo en cuenta que aquí se cambia el “ID” de ejemplo por el que nos proporcionó OneSignal anteriormente.

Este proceso se observa en esta figura:



**Figura 91. Agregar API al proyecto.**  
Fuente: El Autor.

### 6.9.3 Funcionalidad de las notificaciones

Al momento que un usuario descargue la aplicación en su dispositivo, la API registrara un “TOKEN” el cual es la identificación del dispositivo y será esto lo que use para enviar la notificación solo a los dispositivos que poseen la app en su celular.

En la siguiente figura se observa una notificación recibida en un dispositivo Android:



**Figura 92. Notificación exitosa.**  
Fuente: El Autor.

## **7 Publicación de la aplicación dentro de Google Play**

El proceso de publicación de la aplicación para su revisión por Google fue algo extenso y demandando una serie de pasos cruciales y confusos para asegurar el cumplimiento de las políticas y requisitos que esta plataforma solicita.

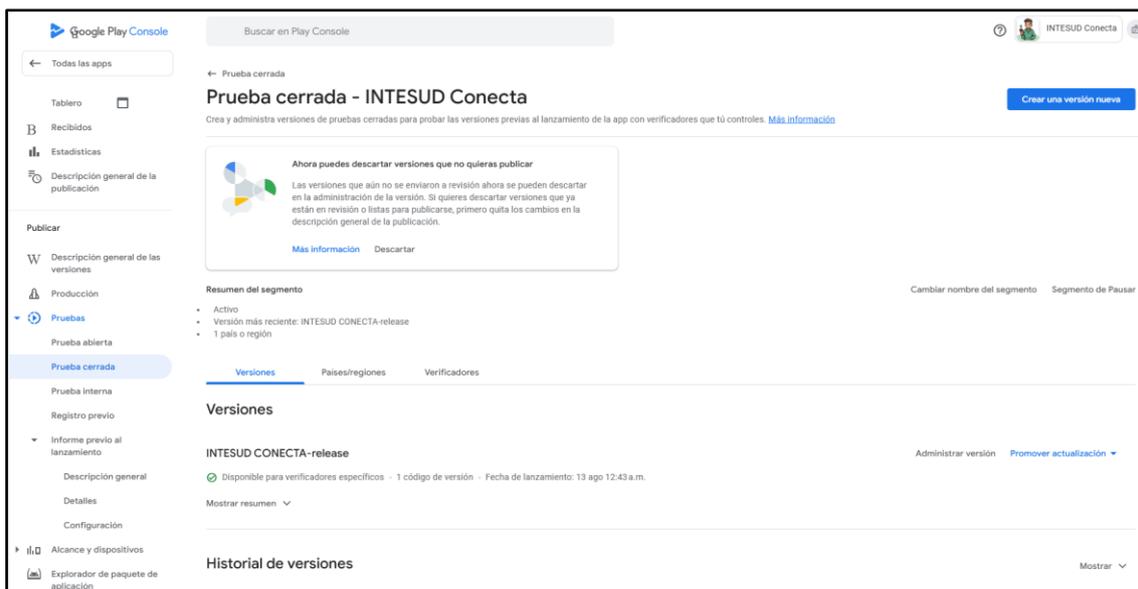
Se requiere tener una cuenta de desarrollador de Google la cual fue proporcionada por la misma institución, por lo cual se redujo el tiempo de creación de la misma y se pasó directamente a completar el procedimiento de publicación que consta de rellenar distintos formularios en los cuales se requiere especificar información como los países en los cuales la aplicación deberá ser visible, su idioma, edades de los usuarios que puedan hacer uso de la red social entre otros datos. Este proceso fue tardado debido a la estructura de la web de “Google Play Console”.

Una vez completados los formularios se solicita subir el aplicativo con distintas capturas de pantalla de las funciones que posee el archivo. Al ser una aplicación, que requiere de registro del usuario, es necesario compartir una cuenta creada dentro del aplicativo para que los profesionales de Google puedan ingresar y revisar toda la aplicación.

Finalmente se requirió una lista de correos electrónicos reales los cuales serían personas de la misma institución los cuales tendrían el rol de “testers” de la aplicación.

Aunque el proceso fue confuso y prolongado se logró cumplir con los requisitos que fueron atendidos con éxito para que la aplicación pase a su fase de revisión.

La aplicación subida a la cuenta de Google Play Console se observa en la siguiente figura:



**Figura 93. App subida dentro de Google Play Console para su revisión.  
Fuente: El Autor.**

Las pruebas de la aplicación están en función de la publicación en la tienda de Android (Google Play) y los “testers” de la institución.

Es responsabilidad del departamento de marketing planificar y hacer el lanzamiento oficial de la aplicación (red social académica INTESUD CONECTA) para conocimiento de la comunidad institucional.

## 8 Conclusiones y Recomendaciones

### 8.1 Conclusiones

- Se desarrollo y se documentó los conceptos, definiciones, tecnologías y herramientas necesarias para la implementación del proyecto de red social académica, Este proceso ha sido fundamental para construir una base sólida que forma el camino del desarrollo de esta forma argumentando las decisiones que se han tomado, facilitando la creación de la aplicación.
- La selección de las herramientas y tecnologías para la implementación de este proyecto fueron evaluadas por su capacidad para cumplir las funcionalidades requeridas y para escalar de forma eficiente según las necesidades del Instituto Superior Tecnológico Sudamericano y otras instituciones, garantizando que la infraestructura tecnológica sea solida segura y adaptable para el futuro.
- El analizar las funcionalidades de plataformas como Facebook, Instagram o Tinder han sido fundamentales para entender las dinámicas que hacen exitosas y utilizables a estas redes sociales. Este análisis ayudo a identificar que aspectos eran más valorados por los usuarios, para ser adaptados a la plataforma académica.
- El diseño de la red social ha sido diseñado cuidadosamente parte por parte para asegurar el cumplir con las necesidades específicas de la institución, considerando minuciosamente aspectos como la estructura de la interfaz de usuario y los datos que sería capaz de mostrar.

- El desarrollo de la aplicación ha sido un proceso prolongado, centrado en crear funcionalidades esenciales como las publicaciones o el servicio de mensajería, cada sección se diseñó con el objetivo de ser eficiente y conectar a toda la comunidad institucional, de esta manera cumple con el objetivo de reducir las barreras existentes entre la institución.
- La aplicación fue enviada a Google Play para ser testeada por los profesionales de Google, posteriormente se subirá el aplicativo a la plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android, una vez se complete su verificación.
- Se consiguió desarrollar e implementar una red social académica que promueva las relaciones entre los miembros que conforman la institución y permita el socializar la información relevante para la comunidad del Instituto Superior Tecnológico Sudamericano

## 8.2 Recomendaciones

- Es aconsejable realizar revisiones y mantenimientos periódicos de la aplicación. Este proceso incluye actualizar y optimizar sus funcionalidades para garantizar que continúe satisfaciendo las necesidades de la institución, manteniendo su eficiencia y seguridad frente a posibles vulnerabilidades.
- Se sugiere implementar actualizaciones y mejoras en las funcionalidades de la aplicación de manera constante. Estas actualizaciones deben estar orientadas a cumplir con los requisitos cambiantes de la institución y a integrar nuevas tecnologías
- Es aconsejable crear canales de comunicación claros y accesibles para que los usuarios puedan informar sobre errores y ofrecer sugerencias. La interacción con el personal administrativo debe ser constante para garantizar que cualquier problema identificado se solucione de manera oportuna y que las mejoras pertinentes se implementen con eficacia.
- Se sugiere llevar a cabo evaluaciones periódicas de la satisfacción de los usuarios, utilizando encuestas y retroalimentación directa. La información recopilada debe ser analizada para realizar ajustes y mejoras en la aplicación.
- La plataforma low-code de FlutterFlow es una herramienta que sugiero a la institución mantener y promover, Durante el desarrollo de este proyecto la utilización de FlutterFlow resulto una experiencia satisfactoria a la hora de crear una aplicación desde cero, siendo un instrumento cuyo funcionamiento es comprensible y accesible en pocas semanas. Estas características facilitan el aprendizaje abriendo numerosas puertas futuras al fomentar las habilidades del desarrollo de aplicaciones. Aunque encontré algunos desafíos en la

plataforma como el idioma, la obtención de permisos para determinadas funciones entre otros, todos fueron superables mediante una lectura cuidadosa y búsqueda constante de soluciones. Enseñar el uso de FlutterFlow impulsaría significativamente el desarrollo dentro y fuera del Instituto Superior Tecnológico Sudamericano, por lo que incluir este material en la malla curricular sería algo innovador que prepararía a los usuarios para enfrentar retos del mercado actual, otorgando competencias altamente demandadas en la industria tecnológica.

- Durante el desarrollo del proyecto surgió un desafío significativo el cual como protagonista tenía a las notificaciones push, este desafío inicia tras la reciente actualización de Fregase, la cual ocasiono que las notificaciones no llegaran a los dispositivos que tenían la aplicación instalada, ante esta situación se optó por el uso de una API de OneSignal la cual resulto exitosa y con mayores beneficios, pues se implementó notificaciones push y dentro de la aplicación, por lo cual demostró ser una solución viable, eficiente y segura, asegurando de esta forma que los usuarios se mantengan al tanto de forma oportuna y confiable. Esta experiencia resalta la importancia de mantenerse informado al respecto de las actualizaciones, últimas versiones y/o cambios dentro de estas plataformas ya que en el ámbito de desarrollo móvil se vuelve algo dinámico y evolutivo al pasar el tiempo por lo que algunas utilidades pueden cambiar y agilizarse y otras dejar de funcionar o ser reemplazadas con nuevas funcionalidades.
- Se sugiere que la institución adquiera una cuenta de Apple Developer, lo que permitirá a la aplicación ser compatible con sistema operativo iOS. El proceso de obtención de la cuenta

puede ser tardado entre 30 a 60 días, esta cuenta ofrece una gran cantidad de beneficios donde se resalta la posibilidad de enviar la aplicación a revisión y posterior publicación en la tienda oficial de Apple. Es importante considerar que tiene un costo anual de 100 dólares anuales, lo que la convierte en una opción realmente costosa a largo plazo.

## 9 Referencias

Coppola, M. (16 de 01 de 2023). *Qué es Java, para qué sirve, características e historia.*

Obtenido de HubSpot: <https://blog.hubspot.es/website/que-es-java>

DataScientest. (07 de 04 de 2022). *MongoDB : todo sobre la base de datos NoSQL orientada*

*a documentos.* Obtenido de DataScientest: [https://datascientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a-](https://datascientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a-documentos#:~:text=ingenier%C3%ADa%20de%20datos.-,MongoDB%20es%20una%20base%20de%20datos%20NoSQL%20orientada%20a%20documentos,almacenan%20como%20colecciones%20y%20documento)

[documentos#:~:text=ingenier%C3%ADa%20de%20datos.-](https://datascientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a-documentos#:~:text=ingenier%C3%ADa%20de%20datos.-,MongoDB%20es%20una%20base%20de%20datos%20NoSQL%20orientada%20a%20documentos,almacenan%20como%20colecciones%20y%20documento)

[,MongoDB%20es%20una%20base%20de%20datos%20NoSQL%20orientada%20a%20documentos,almacenan%20como%20colecciones%20y%20documento](https://datascientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a-documentos#:~:text=ingenier%C3%ADa%20de%20datos.-,MongoDB%20es%20una%20base%20de%20datos%20NoSQL%20orientada%20a%20documentos,almacenan%20como%20colecciones%20y%20documento)

Equipo editorial de IONOS. (15 de 10 de 2020). *Dart de Google: Una introducción al lenguaje*

*Dart.* Obtenido de IONOS: <https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/lenguaje-de-programacion-dart-de-google/>

Equipo editorial, Etecé. (19 de 11 de 2023). *Base de datos.* Obtenido de Concepto:

<https://concepto.de/base-de-datos/>

Firebase. (s.f.). *Crea tu app lo mejor posible con Firebase y la IA generativa.* Recuperado el

07 de 07 de 2024, de Crea tu app lo mejor posible con Firebase y la IA generativa:

[https://firebase.google.com/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=latam-LATAM-all-es-dr-SKWS-all-all-trial-b-dr-1707800-](https://firebase.google.com/?utm_source=google&utm_medium=cpc&utm_campaign=latam-LATAM-all-es-dr-SKWS-all-all-trial-b-dr-1707800-LUAC0020239&utm_content=text-ad-none-any-DEV_c-CRE_654758081663-ADGP_Hybrid%20%7C%20SKWS%20-%20BRO%20%7C%20Txt_Compute-Firebas)

[LUAC0020239&utm\\_content=text-ad-none-any-DEV\\_c-CRE\\_654758081663-](https://firebase.google.com/?utm_source=google&utm_medium=cpc&utm_campaign=latam-LATAM-all-es-dr-SKWS-all-all-trial-b-dr-1707800-LUAC0020239&utm_content=text-ad-none-any-DEV_c-CRE_654758081663-ADGP_Hybrid%20%7C%20SKWS%20-%20BRO%20%7C%20Txt_Compute-Firebas)

[ADGP\\_Hybrid%20%7C%20SKWS%20-%20BRO%20%7C%20Txt\\_Compute-](https://firebase.google.com/?utm_source=google&utm_medium=cpc&utm_campaign=latam-LATAM-all-es-dr-SKWS-all-all-trial-b-dr-1707800-LUAC0020239&utm_content=text-ad-none-any-DEV_c-CRE_654758081663-ADGP_Hybrid%20%7C%20SKWS%20-%20BRO%20%7C%20Txt_Compute-Firebas)

[Firebas](https://firebase.google.com/?utm_source=google&utm_medium=cpc&utm_campaign=latam-LATAM-all-es-dr-SKWS-all-all-trial-b-dr-1707800-LUAC0020239&utm_content=text-ad-none-any-DEV_c-CRE_654758081663-ADGP_Hybrid%20%7C%20SKWS%20-%20BRO%20%7C%20Txt_Compute-Firebas)

Firebase. (s.f.). *Firebase.* Recuperado el 08 de 07 de 2024, de Firebase:

[https://firebase.google.com/?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=latam-LATAM-all-es-dr-SKWS-all-all-trial-b-dr-1707800-](https://firebase.google.com/?utm_source=google&utm_medium=cpc&utm_campaign=latam-LATAM-all-es-dr-SKWS-all-all-trial-b-dr-1707800-LUAC0020239&utm_content=text-ad-none-any-DEV_c-CRE_654758081663-ADGP_Hybrid%20%7C%20SKWS%20-%20BRO%20%7C%20Txt_Compute-Firebas)

[LUAC0020239&utm\\_content=text-ad-none-any-DEV\\_c-CRE\\_654758081663-](https://firebase.google.com/?utm_source=google&utm_medium=cpc&utm_campaign=latam-LATAM-all-es-dr-SKWS-all-all-trial-b-dr-1707800-LUAC0020239&utm_content=text-ad-none-any-DEV_c-CRE_654758081663-ADGP_Hybrid%20%7C%20SKWS%20-%20BRO%20%7C%20Txt_Compute-Firebas)

ADGP\_Hybrid%20%7C%20SKWS%20-%20BRO%20%7C%20Txt\_Compute-  
Firebas

FlutterFlow. (s.f.). *Cree aplicaciones más rápido que nunca*. Recuperado el 07 de 07 de 2024, de Cree aplicaciones más rápido que nunca: <https://flutterflow.io/>

FlutterFlow. (s.f.). *FlutterFlow*. Recuperado el 08 de 07 de 2024, de FlutterFlow: <https://flutterflow.io/>

Greaves, A., & Mengistu, A. (s.f.). *Flutterflow*. Recuperado el 07 de 07 de 2024, de NoCodeHackers: <https://www.nocodehackers.es/herramientas-nocode/flutterflow#:~:text=Qu%C3%A9%20es%20Flutterflow%3F-,Flutterflow%20es%20una%20herramienta%20nocode%20de%20desarrollo%20de%20aplicaciones%20m%C3%B3viles,sin%20necesidad%20de%20escribir%20c%C3%B3digo.>

IBM (Ed.). (s.f.). *¿Qué es una base de datos relacional?* Recuperado el 07 de 07 de 2024, de IBM: <https://www.ibm.com/mx-es/topics/relational-databases>

Kinsta. (19 de 04 de 2019). *¿Qué es PostgreSQL?* Obtenido de Kinsta: <https://kinsta.com/es/base-de-conocimiento/que-es-postgresql/>

Kolsquare. (27 de 06 de 2024). *Estadísticas clave de Facebook para 2024*. Obtenido de Kolsquare: <https://www.kolsquare.com/es/blog/las-estadisticas-de-facebook-para-2023>

Lima, V. (20 de 08 de 2022). *¿Qué es bubble.io y cómo funciona?* Obtenido de Tutorialsbubble: <https://tutorialsbubble.com/que-es-bubble-io-y-como-funciona>

Londoño, P. (03 de 04 de 2023). *Qué es Python, para qué sirve y cómo se usa (+ recursos para aprender)*. Obtenido de HubSpot: <https://blog.hubspot.es/website/que-es-python>

López Mora, S. (17 de 05 de 2020). *Firebase: qué es, para qué sirve, funcionalidades y ventajas*. Obtenido de DIGITAL55: <https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/>

- Mosca, G. (11 de 07 de 2023). *NoSQL: Qué son, para qué sirven, tipos y ventajas*. Obtenido de InnovaciónDigital360: <https://www.innovaciondigital360.com/big-data/nosql-que-son-para-que-sirven-tipos-y-ventajas/>
- Pérez, A. (12 de 03 de 2024). *Más de 200 millones de negocios usan Facebook para conectar con sus audiencias*. Obtenido de Reactiva Online: <https://www.reactivaonline.com/estadisticas-facebook/#estadisticas-generales>
- Roast. (2024). *Tinder estadísticas 2024: ¡Todo lo que necesitas saber sobre la aplicación para ligar!* Obtenido de Roast: <https://roast.dating/es/blog/estadisticas-de-tinder>
- Robledano, A. (24 de 09 de 2019). *Qué es MySQL: Características y ventajas*. Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-es-mysql/>
- Statista. (01 de 2024). *Redes sociales con mayor número de usuarios activos mensuales a nivel mundial en enero de 2024*. Obtenido de Statista: <https://es.statista.com/estadisticas/600712/ranking-mundial-de-redes-sociales-por-numero-de-usuarios/>
- Wikipedia. (s.f.). *Historia de Facebook*. Recuperado el 08 de 07 de 2024, de Wikipedia: [https://en.wikipedia.org/wiki/History\\_of\\_Facebook](https://en.wikipedia.org/wiki/History_of_Facebook)

## **ANEXOS**

**Anexo 1: Manual de Red Social Académica.**