

ChatoChat

Autor: Pablo Ortiz Muñoz

Tutor: Javier Martin Rivero

Fecha de entrega: 21/5/2025

Convocatoria: 2024 2025

Índice:

Introducción.....	1
Motivación.....	1
Abstract.....	2
Objetivos Propuestos.....	2
Metodología Utilizada.....	3
Tecnologías y herramientas utilizadas en el proyecto:.....	4
Estimación de recursos y planificación:.....	6
Análisis.....	6
Diseño.....	14
Despliegue y pruebas.....	15
Conclusiones.....	18
Vías Futuras.....	18
Glosario.....	19
Bibliografía.....	25

Introducción

En la actualidad, la comunicación instantánea se ha convertido en un aspecto fundamental de la vida diaria, especialmente entre los jóvenes, quienes utilizan diversas plataformas para interactuar de manera rápida y directa. El auge de aplicaciones de mensajería como WhatsApp, Telegram y otras, ha transformado por completo la forma en que nos comunicamos, haciendo que la interacción en tiempo real sea accesible y conveniente para todos. Este fenómeno ha logrado que surjan nuevas necesidades y expectativas en los usuarios, quienes buscan aplicaciones no solo para conectarse con sus contactos, sino también para conocer nuevas personas y disfrutar de interacciones más dinámicas y personalizadas.

Con este contexto en mente, el presente trabajo se centra en el desarrollo de una aplicación de mensajería instantánea, que no solo se limita a ofrecer la funcionalidad básica de intercambio de mensajes entre usuarios, sino que va más allá al incorporar dos características innovadoras. La primera de estas es la posibilidad de interactuar con personas desconocidas, lo que permite a los usuarios ampliar su red de contactos de forma espontánea y enriquecedora. Esta función responde a la creciente demanda de plataformas que favorezcan la expansión de redes sociales, facilitando nuevas conexiones entre individuos con intereses similares.

La segunda característica diferenciadora es la integración de un módulo de inteligencia artificial (IA) amigable. Este componente tiene como objetivo proporcionar a los usuarios una experiencia interactiva diferente, en la que pueden conversar con una IA diseñada para ofrecer compañía, proporcionar información relevante o simplemente mantener una conversación informal. La inclusión de este módulo se debe a la creciente relevancia de la inteligencia artificial en la interacción digital, buscando una experiencia más personalizada y atractiva, que podría transformar la forma en que las personas utilizan las aplicaciones de mensajería.

La justificación de la elección de este tema radica en el interés por explorar nuevas formas de interacción social en plataformas de mensajería, incorporando tecnologías de vanguardia como Kotlin para el desarrollo de la aplicación y Firebase como plataforma para la gestión de datos y la autenticación de usuarios. Este enfoque tiene como objetivo no solo ofrecer una herramienta funcional, sino también innovadora, que combine la simplicidad de las aplicaciones tradicionales con características más avanzadas y atractivas para el usuario.

En cuanto al contenido del trabajo, se desarrollará un análisis detallado sobre las decisiones técnicas y de diseño tomadas durante el proceso de desarrollo de la aplicación. Se explicará el uso de Kotlin como lenguaje principal de desarrollo y Firebase como solución para la gestión de datos en tiempo real y la autenticación de usuarios. Además, se abordarán las fases del desarrollo del proyecto, los retos enfrentados, las soluciones adoptadas y el impacto potencial de la integración de inteligencia artificial en la experiencia del usuario. Finalmente, se presentarán las conclusiones del proyecto, destacando los logros alcanzados y las áreas de mejora para futuras actualizaciones.

En resumen, este trabajo no solo ofrece una aplicación funcional para mensajería instantánea, sino que también introduce elementos innovadores que podrían definir el futuro de las plataformas de comunicación digital, convirtiéndola en una herramienta útil y dinámica en el día a día de los usuarios.

Motivación

Este proyecto surgió de varias experiencias previas que tuve, especialmente en el proyecto de “Dating Love” que hice con mis compañeros. En ese proyecto, trabajamos con Kotlin, y la verdad es que me gustó mucho. Es un lenguaje muy fácil de usar, más limpio que Java, y tiene un montón de herramientas que facilitan el trabajo en Android. Me sentí muy cómodo usándolo y, por eso, cuando empecé a pensar en hacer una nueva app, supe que Kotlin sería el ideal para este proyecto de mensajería.

La idea de crear una app de mensajería como WhatsApp ya estaba rondando en mi cabeza desde hace un tiempo, pero cuando empecé a desarrollarlo, WhatsApp todavía no había implementado la inteligencia artificial (IA) en su plataforma, y eso me dio una idea. Pensé que podía ser algo interesante y único incorporar una IA amigable dentro de la app, algo que no solo sirviera para enviar mensajes, sino también para ofrecer compañía o tener una conversación divertida de vez en cuando. Al no ver esto en aplicaciones tan populares como WhatsApp en ese momento, sentí que podía ser una buena oportunidad para innovar un poco.

Además, la IA siempre me ha interesado, y este proyecto me dio la excusa perfecta para adentrarme más en ese campo. Quería hacer algo diferente, no solo una app de mensajería común, sino algo que tuviera ese toque extra de interacción que podría hacer que los usuarios disfruten más al usarla. La combinación de Kotlin, que ya me había funcionado bien en otros proyectos, y la idea de integrar una IA dentro de la app, me motivó a seguir adelante con el proyecto.

En resumen, el proyecto me motivó principalmente por las experiencias previas con Kotlin, que me hicieron confiar en el lenguaje, y por las oportunidades que vi para innovar con IA en una app de mensajería. Creo que esta mezcla de tecnología y creatividad le puede dar un toque único a la app y, de paso, me ha permitido aprender y aplicar nuevas cosas.

Objetivos Propuestos

Objetivo General

En un mundo cada vez más interconectado, las aplicaciones de mensajería instantánea se han convertido en una de las herramientas digitales más utilizadas para la comunicación. Plataformas como WhatsApp han transformado la forma en que las personas interactúan, permitiendo conversaciones fluidas y en tiempo real. Sin embargo, muchas de estas aplicaciones están centradas únicamente en la comunicación entre contactos conocidos, dejando de lado la posibilidad de conectar con nuevas personas de manera sencilla y segura.

Este proyecto surge con la intención de desarrollar una aplicación de mensajería para Android que combine la funcionalidad clásica de los chats privados con una característica innovadora: la posibilidad de entablar conversaciones aleatorias con otros usuarios

Objetivos Específicos

1. Diseñar y desarrollar una interfaz intuitiva y atractiva para la experiencia de usuario (UX/UI), asegurando una navegación fluida en la aplicación.
2. Implementar un sistema de mensajería en tiempo real utilizando Firebase Realtime Database, para garantizar la comunicación instantánea.

3. Desarrollar una funcionalidad de emparejamiento aleatorio que permita a los usuarios iniciar conversaciones con personas desconocidas bajo ciertos criterios de anonimato y seguridad.
4. Garantizar la privacidad y seguridad de los usuarios mediante cifrado de mensajes, autenticación segura y mecanismos de moderación para evitar el mal uso de la plataforma.
5. Aplicar metodologías ágiles para la gestión y desarrollo del proyecto.

Abstract

In an increasingly interconnected world, instant messaging applications have become essential tools for digital communication. While platforms like WhatsApp have revolutionized how people stay in touch, they often focus solely on communication between known contacts. This leaves a gap for users who wish to connect with new people in a simple and secure way.

This project aims to develop an Android messaging app that merges the traditional functionality of private chats with a novel feature: random conversations with other users. This dynamic approach is designed to promote spontaneous interactions, ideal for those looking to expand their social circle or engage in casual conversations.

The main goal is to offer a user-friendly and secure platform that encourages both familiar and unexpected interactions. To achieve this, the application will include a modern, intuitive user interface (UI/UX), real-time messaging powered by Firebase Realtime Database, and a secure, anonymous matching system for random chats. Strong emphasis will be placed on user privacy and platform safety through encrypted communication, secure authentication, and moderation tools to prevent misuse.

The project will be developed using agile methodologies to ensure flexibility, iterative improvement, and a user-centered design process.

Objetivos Propuestos

Objetivo General

- Desarrollar una aplicación de mensajería instantánea en Kotlin que permita a los usuarios comunicarse de forma segura y eficiente mediante chats

privados, aleatorios y con una inteligencia artificial, integrando funcionalidades avanzadas como notificaciones, cifrado y gestión de usuarios, garantizando una experiencia de usuario fluida y moderna.

Objetivos Específicos

- Implementar un sistema de chat privado que permita el envío y recepción de mensajes en tiempo real, con soporte para imágenes, estado en línea y eliminación de mensajes, utilizando Firebase Realtime Database como backend.
- Desarrollar un módulo de chat aleatorio que conecte al usuario con personas desconocidas, ofreciendo la posibilidad de abandonar la conversación y reportar o bloquear en caso de mal uso.
- Integrar un sistema de chat con una inteligencia artificial que permita al usuario mantener conversaciones simuladas con un asistente amigable.
- Establecer un sistema de notificaciones push que informe al usuario cuando recibe nuevos mensajes y permita configurar las preferencias de notificación.
- Garantizar la seguridad y privacidad de las conversaciones mediante cifrado de datos, autenticación segura y un sistema de moderación automatizado para contenido inapropiado.
- Diseñar una interfaz intuitiva y adaptable, que funcione correctamente en distintos tamaños de pantalla.
- Asegurar la mantenibilidad y escalabilidad del código fuente, siguiendo buenas prácticas de desarrollo y facilitando futuras actualizaciones sin pérdida de datos.

Metodología Utilizada

Para el desarrollo del proyecto, se ha optado por la metodología ágil KANBAN, ya que permite una gestión visual del flujo de trabajo, facilitando la organización de tareas y la optimización del proceso de desarrollo. Esta metodología se basa en la mejora continua, permitiendo adaptarse a cambios sin necesidad de ciclos de desarrollo estrictos.

KANBAN ayuda a mantener un flujo de trabajo constante mediante el uso de un tablero visual, donde las tareas se dividen en columnas representando diferentes estados, como "Pendiente", "En proceso" y "Finalizado". Esto permite un control eficiente del progreso y ayuda a identificar posibles bloqueos en el desarrollo de la aplicación.

Backlog de Tareas

1. Diseño de la Aplicación

- Creación de la interfaz en Figma.
- Creación del diagrama de Gantt
- Documentación (Requisitos funcionales y no funcionales, diagrama de clases , etc ...)

2. Desarrollo del Chat

- Implementación del sistema de autenticación de usuarios
- Configuración de bases de datos en tiempo real
- Desarrollo del envío y recepción de mensajes
- Notificaciones Push

3. Funcionalidad de Chat Aleatorio

- Implementación de un sistema para hablar con cualquiera

4. Funcionalidad IA

- Implementación IA

5. Seguridad y Privacidad

- Corrección de errores y optimización de código
- Reportes

Tecnologías y herramientas utilizadas en el proyecto:

Para el desarrollo de este proyecto, se han seleccionado herramientas y tecnologías que permiten la creación de una aplicación Android eficiente, escalable y con una buena experiencia de usuario. A continuación, se detallan las principales tecnologías utilizadas y los motivos de su elección:

1. Entorno de Desarrollo

Android Studio: Es el entorno de desarrollo integrado (IDE) oficial para la creación de aplicaciones Android. Se eligió por su compatibilidad con el SDK de Android, sus herramientas avanzadas de depuración y su integración con emuladores y dispositivos físicos.

2. Lenguaje de Programación

Kotlin: Se ha optado por Kotlin debido a sus ventajas sobre Java, como una sintaxis más limpia y una mejor integración con Firebase.

3. Diseño de Interfaz y Experiencia de Usuario (UX/UI)

Figma: Utilizado para la creación de prototipos y el diseño de la interfaz de usuario. Se eligió por su facilidad de uso y compatibilidad con herramientas de desarrollo.

RemoveBG: Herramienta para la eliminación automática de fondos en imágenes, facilitando la creación de elementos gráficos sin necesidad de software más complejo.

4. Backend y Gestión de Datos

Firebase Realtime Database: Base de datos NoSQL utilizada para el almacenamiento y sincronización de datos en tiempo real. Se eligió por su facilidad de integración con Android y su bajo tiempo de respuesta para aplicaciones de mensajería.

Firebase Cloud Messaging (FCM): Servicio utilizado para el envío de notificaciones push y la gestión de mensajes en tiempo real entre los usuarios. Es una solución eficiente y gratuita dentro del ecosistema Firebase.

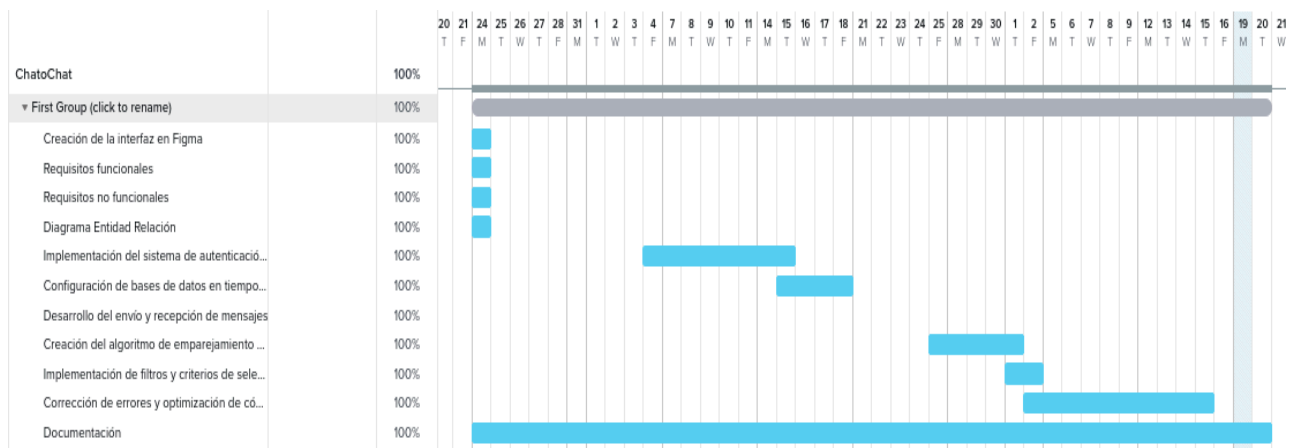
5. Gestión de Proyecto

[Trello](#): Plataforma para la gestión ágil del proyecto basada en metodología SCRUM. Se eligió por su flexibilidad en la organización de tareas y facilidad de uso en equipos de desarrollo.

Justificación de la Elección de las Tecnologías

El ecosistema de Android ofrece múltiples herramientas y tecnologías para el desarrollo de aplicaciones móviles. En este caso, la elección de Firebase como backend se debe a su facilidad de uso y su integración nativa con Android. Kotlin ha sido seleccionado por su eficiencia y popularidad en el desarrollo moderno de Android. Además, herramientas como Figma y Trello facilitan el diseño y la gestión del proyecto. En conjunto, estas tecnologías permiten desarrollar una aplicación de mensajería rápida, segura y con una experiencia de usuario optimizada.

Estimación de recursos y planificación:



Análisis

Requisitos funcionales:

Chat Privado

- Los usuarios podrán enviar y recibir mensajes en tiempo real.
- Se podrá compartir imágenes dentro del chat.
- Se podrá compartir audios dentro del chat.
- Se podrá compartir ubicaciones con GoogleMaps dentro del chat.
- Los mensajes se almacenarán en Firebase Realtime Database.
- Se permitirá ver el estado en línea de otros usuarios.

Chat con quien quieras

- El usuario podrá conectarse con cualquier persona que esté registrada
- El usuario podrá abandonar la conversación en cualquier momento.
- Se podrá reportar usuarios en caso de mal uso.

Chat IA

- El usuario podrá hablar con una IA

Notificaciones

- Se enviarán notificaciones push cuando el usuario reciba un nuevo mensaje.
- El usuario podrá configurar las preferencias de notificación (activar/desactivar).

Seguridad y Privacidad

- Se implementará cifrado de mensajes para proteger la privacidad de las conversaciones.
- El usuario podrá reportar a otros usuarios.
- Se implementará un sistema de moderación para detectar contenido inapropiado.

Requisitos No funcionales:

Rendimiento y escalabilidad

- La aplicación debe ser capaz de manejar múltiples conversaciones simultáneamente sin afectar el rendimiento.
- El tiempo de respuesta del chat en tiempo real no debe superar los 2 segundos.
- La aplicación debe ser compatible con dispositivos Android desde la versión 8.0 (Oreo) en adelante.

Usabilidad y experiencia de usuario (UX/UI)

- La interfaz debe ser intuitiva y fácil de usar.
- El diseño debe adaptarse a diferentes tamaños de pantalla y resoluciones.
- La aplicación debe ofrecer un modo oscuro para mejorar la experiencia de uso.

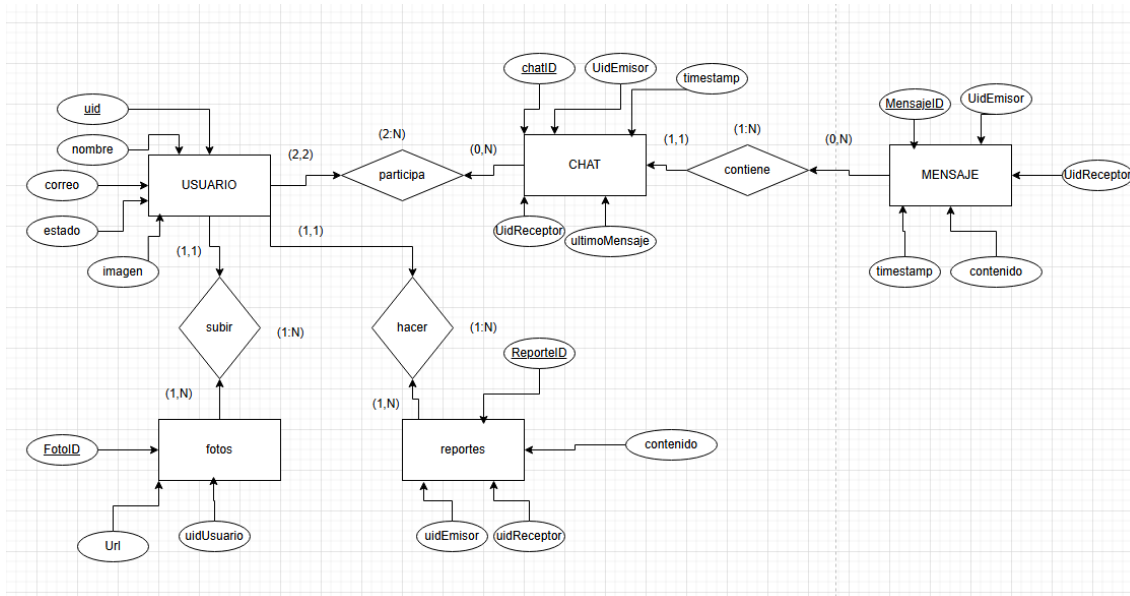
Seguridad y privacidad

- Los datos de usuario deben almacenarse de forma segura en Firebase con reglas de acceso adecuadas.
- La aplicación debe cumplir con normativas de protección de datos como GDPR o similares.
- Se debe evitar el acceso no autorizado a las conversaciones mediante autenticación segura.

Mantenimiento y actualización

- La aplicación debe permitir actualizaciones sin afectar la información almacenada de los usuarios.
- El código debe seguir buenas prácticas de desarrollo para facilitar su mantenimiento y escalabilidad.

ER



Modelo Relacional:

USUARIO (ID_USUARIO PK)

FOTOS (ID PK, ID_USUARIO FK → USUARIO(ID_USUARIO))

REPORTES (ID PK, ID_USUARIO FK → USUARIO(ID_USUARIO), DESCRIPCION)

MENSAJEAR (ID PK, EMISOR_ID FK → USUARIO(ID_USUARIO), RECEPTOR_ID
FK → USUARIO(ID_USUARIO))

INTERACTUAR (ID PK, ID_USUARIO1 FK → USUARIO(ID_USUARIO),
ID_USUARIO2 FK → USUARIO(ID_USUARIO))

MENSAJE_ALEATORIO (ID PK, ID_USUARIO1 FK → USUARIO(ID_USUARIO),
ID_USUARIO2 FK → USUARIO(ID_USUARIO))

Diagrama de clases Inicial

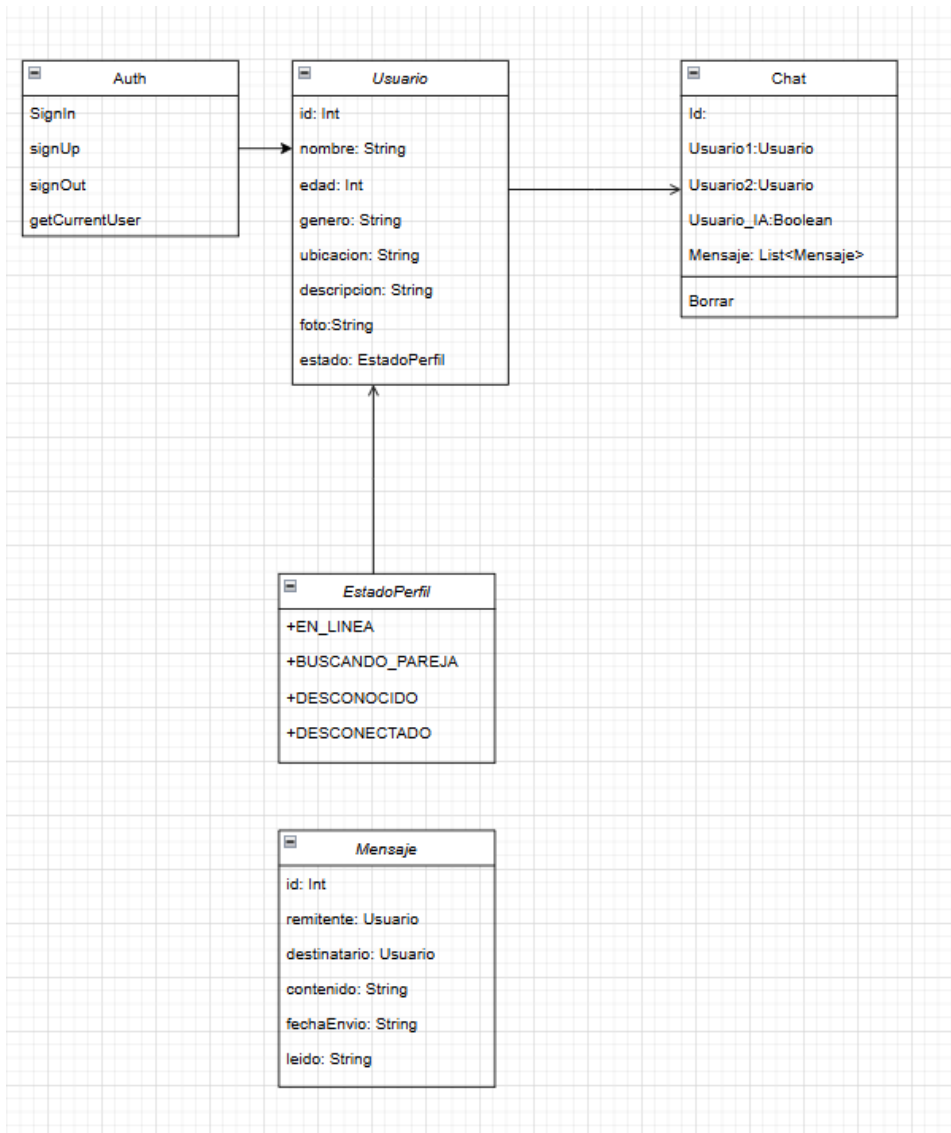
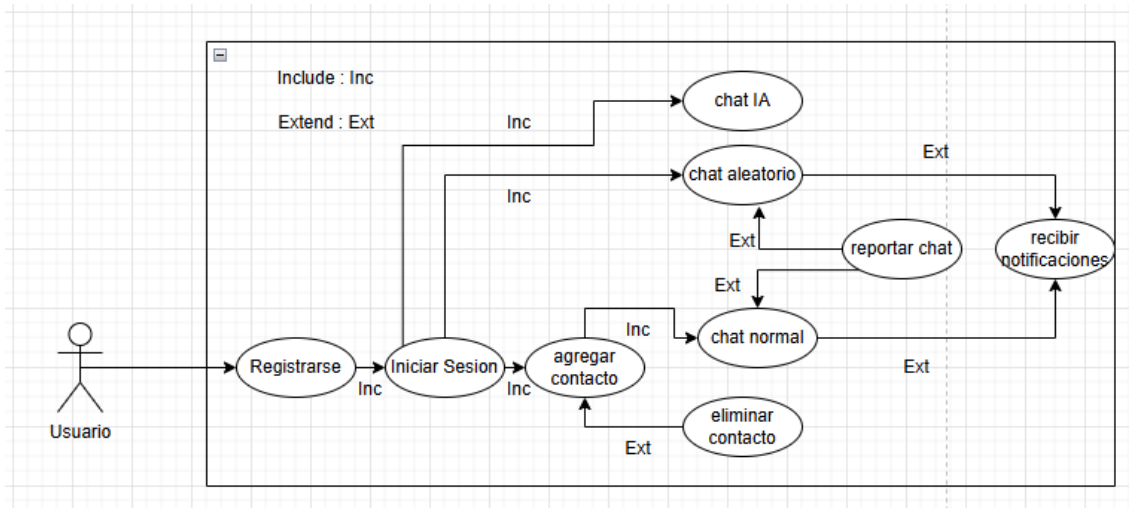


Diagrama de clases Final

Diagrama de casos de uso



Caso de Uso 1	Registrarse
Alias	-
Actores	Usuario
Requisito Funcional	El usuario puede registrarse en la App
Descripción	Permite al usuario registrarse en la App para tener una cuenta y recordar las credenciales de esta
Referencias	-
Comentarios	-

Caso de Uso 2	Iniciar Sesión
----------------------	----------------

Alías	-
Actores	Usuario
Requisito Funcional	El usuario puede iniciar sesión en la App
Descripción	Permite al usuario iniciar sesión en la App con su usuario y contraseña o automáticamente
Referencias	-
Comentarios	-

Caso de Uso 3	Agregar Contacto
Alías	-
Actores	Usuario
Requisito Funcional	El usuario puede encontrar a otros usuarios
Descripción	Permite al usuario acceder a una lista con los usuarios del servidor
Referencias	-
Comentarios	-

Caso de Uso 4	Eliminar contacto
Alías	-
Actores	Usuario
Requisito Funcional	El usuario puede eliminar un usuario de su chat
Descripción	Permite al usuario eliminar el chat que tenga con otro usuario para no verlo más
Referencias	-
Comentarios	-

Caso de Uso 5	Chat Normal
Alías	-
Actores	Usuario
Requisito Funcional	El usuario puede crear un chat con un usuario
Descripción	Permite al usuario crear un chat para hablar, enviar ubicaciones, imágenes y audios con otro usuario
Referencias	-
Comentarios	-

Caso de Uso 6	Reportar Chat
Alías	-

Actores	Usuario
Requisito Funcional	El usuario puede reportar a un usuario por el uso indebido del chat
Descripción	Permite al usuario crear un formulario de reporte para enfrentar un mal uso del chat por parte del otro usuario
Referencias	-
Comentarios	-

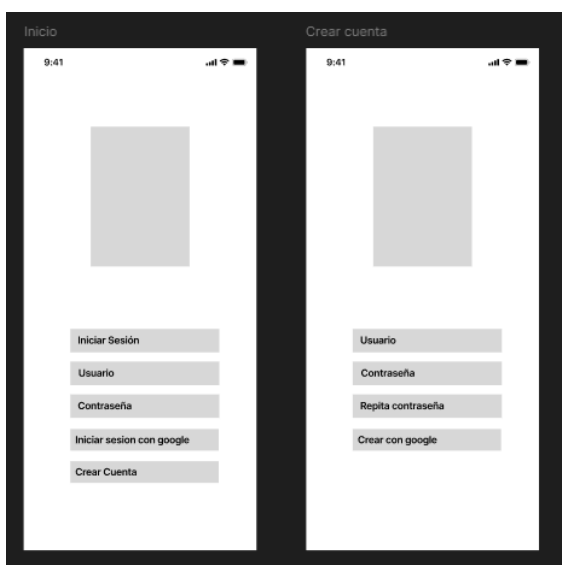
Caso de Uso 7	Chat Aleatorio
Alías	-
Actores	Usuario
Requisito Funcional	El usuario puede hablar con cualquier otro usuario
Descripción	Permite al usuario entablar una conversación con cualquier usuario que esté registrado
Referencias	-
Comentarios	-

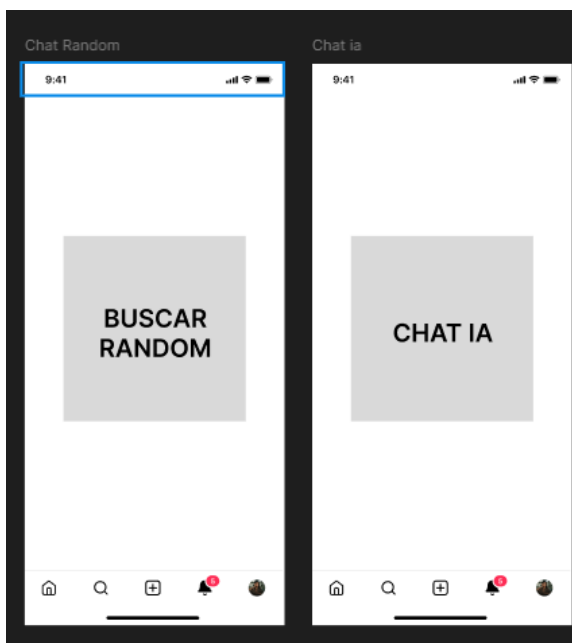
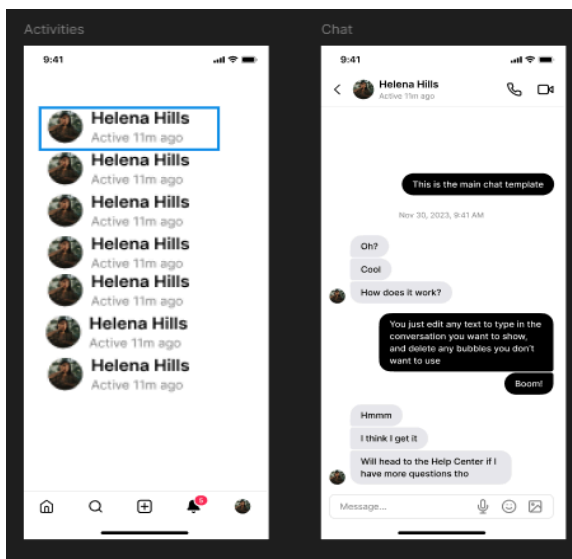
Caso de Uso 8	Notificaciones
Alías	-
Actores	Usuario
Requisito Funcional	El usuario puede recibir notificaciones
Descripción	Permite al usuario recibir notificaciones

	de mensajes a tiempo real siempre que disponga de conexión a la red
Referencias	-
Comentarios	-

Caso de Uso 9	Chat IA
Alías	-
Actores	Usuario
Requisito Funcional	El usuario puede entablar conversación con una IA
Descripción	Permite al usuario hablar con una IA
Referencias	-
Comentarios	-

Diseño





Código relevante

Registro de usuario usando Firebase Authentication y Realtime Database

Identificador	Proveedores	Fecha de creación ↓	Fecha de acceso	UID de usuario
rafa@rafa.com	✉	14 may 2025	14 may 2025	YhiaxZBDDT09u5NfILQcMKN...
pablo@pablo.com	✉	14 may 2025	14 may 2025	H4nIDTWPq9MBUbur1xyoBh...
prueba@prueba.com	✉	14 may 2025	14 may 2025	BBMnoptMXnPaEAwSUNyXxV...

Al configurar el registro de usuarios, nos apoyamos en Firebase Authentication para dar de alta una cuenta nueva con un correo y una clave. A esto le añadimos Firebase Realtime Database, que es donde guardamos los detalles extra del usuario, como su nombre y su dirección de correo.

Todo esto se hace dentro de la función RegistrarUsuario(nombre: String, correo: String, contraseña: String). Primero, llamamos al método createUserWithEmailAndPassword que nos da Firebase Authentication para registrar al usuario. Si todo va bien, conseguimos el UID (un Identificador Único de Usuario), que Firebase da solo y asegura que cada usuario sea único en todo el sistema.

Después, entramos a la base de datos en tiempo real buscando el nodo Usuarios, usando el UID como el nombre del nuevo nodo. Creamos un HashMap con lo que queremos guardar: UID, nombre, correo electrónico, una imagen (que al principio está vacía) y un campo para buscar con el nombre en minúsculas, que sirve para las búsquedas en la app.

Estos datos se meten en la base de datos con el método updateChildren. Si todo sale bien, le avisamos al usuario con un mensaje (Toast) y lo mandamos directo a la pantalla principal de la app (Inicio). Si hay algún problema, atrapamos el error y le mostramos al usuario un mensaje explicando lo que pasó.

Despliegue y pruebas

Pruebas de caja negra

<u>REGISTRO</u>		
Datos Introducidos	Especificación de la prueba	Resultado
correo: prueba@prueba.com	Probamos que pasa si el correo es correcto pero la	La validación es errónea

contra: 123	contraseña es menor a 6 dígitos	
correo: prueba.prueba.com contra: 123456	Probamos que pasa si la contraseña es correcta pero el correo es erróneo	La validación es errónea
correo: prueba@prueba.com contra: 123456	Probamos si los dos campos están bien	La validación es correcta

<u>Login</u>		
Datos Introducidos	Especificación de la prueba	Resultado
correo: prueba@prueba.com contra: 123	Probamos que pasa si el correo es correcto pero la contraseña es menor a 6 dígitos	La validación es errónea
correo: prueba.prueba.com contra: 123456	Probamos que pasa si la contraseña es correcta pero el correo es erróneo	La validación es errónea
correo: prueba@prueba.com contra: 123456	Probamos si los dos campos están bien	La validación es correcta

REPORTAR

Datos Introducidos	Especificación de la prueba	Resultado
Motivo:	Probamos que pasa si el motivo está vacío	No se manda el reporte
Motivo: ...	Probamos que pasa si el motivo está relleno	El reporte se crea

Conclusiones

El desarrollo de esta aplicación de mensajería instantánea ha sido una experiencia enriquecedora que me ha permitido explorar y aplicar una serie de tecnologías modernas en el entorno Android, con un enfoque principal en el lenguaje Kotlin y la plataforma Firebase. A lo largo del proceso de desarrollo, se ha logrado cumplir la mayoría de los objetivos propuestos, aunque algunos no se alcanzaron de la manera inicialmente planteada. En este sentido, la aplicación es funcional, intuitiva y segura, permitiendo a los usuarios no solo comunicarse con sus contactos conocidos, sino también entablar conversaciones con usuarios aleatorios. Este último objetivo, aunque no se implementó de acuerdo con el plan inicial, fue sustituido por una innovación propia que permite una mejor integración social dentro de la plataforma.

Uno de los principales logros fue la integración de una funcionalidad que facilita el chat entre usuarios aleatorios, una característica que pretendía ser uno de los pilares distintivos de la app. Sin embargo, a medida que avanzaba el desarrollo, me di cuenta de que la implementación del sistema de chat aleatorio presentaba varios desafíos técnicos y de seguridad. Después de evaluar diferentes alternativas, decidí optar por un enfoque innovador para la creación de chats, que mejor se adaptara tanto a la experiencia del usuario como a las restricciones de seguridad y privacidad. Este cambio de enfoque derivó en un rediseño que, aunque no alineado con los objetivos originales, resultó en una solución que aporta mayor flexibilidad y escalabilidad al sistema.

La metodología ágil KANBAN jugó un papel crucial durante el proceso de desarrollo. Su implementación permitió organizar y gestionar de manera visual el flujo de trabajo, facilitando la asignación y seguimiento de tareas, lo que resultó en una mejora continua del proyecto. La capacidad de adaptarse rápidamente a cambios y ajustes, sin la necesidad de estrictos ciclos de desarrollo, fue clave para mantener el progreso en todo momento, a pesar de los imprevistos. Esta metodología también fue fundamental para optimizar el tiempo de desarrollo y mejorar la colaboración del equipo en las diferentes fases del proyecto.

La escalabilidad del proyecto fue otro aspecto crucial que se abordó de manera efectiva. Gracias a la flexibilidad que ofrece Firebase y la arquitectura modular diseñada para la aplicación, la plataforma está preparada para crecer y manejar un número mayor de usuarios y funciones en el futuro, sin comprometer su rendimiento. La interfaz de usuario también fue adaptada a distintos tamaños de pantalla, lo que garantiza una experiencia consistente y agradable en dispositivos Android de diferentes características.

Al analizar la combinación de mensajería tradicional con las funciones innovadoras, como el módulo de IA, se concluye que este enfoque puede ofrecer una experiencia de usuario más rica y variada, que podría diferenciar la aplicación de otros competidores en el mercado. Si bien no se incluyó una IA completamente personalizada en esta versión, la implementación inicial de la IA para interacciones básicas representa un primer paso hacia un sistema de conversación más avanzado y natural.

En cuanto a los objetivos no alcanzados, el chat aleatorio fue uno de los más destacados. Aunque inicialmente este componente iba a ser central en la aplicación, decidí no implementarlo de la manera original debido a los problemas relacionados con la seguridad y la protección de los usuarios. A pesar de no haber alcanzado este objetivo de la forma en que se había planeado, el proyecto logró evolucionar hacia una dirección más innovadora, mejor adaptada a las necesidades y demandas del mercado actual.

En conclusión, el desarrollo de esta aplicación de mensajería ha sido un proceso de aprendizaje continuo y adaptación. Las dificultades encontradas han sido superadas con soluciones innovadoras que no solo cumplen los objetivos planteados, sino que también han permitido explorar nuevas funcionalidades que pueden enriquecer la experiencia del usuario. De cara al futuro, se espera que la aplicación evolucione con nuevas características que puedan hacerla aún más competitiva, como el uso de inteligencia artificial más avanzada, la mejora de la seguridad, y la integración de nuevas formas de interacción entre los usuarios.

Vías Futuras

Aunque la aplicación desarrollada cumple con las funcionalidades básicas y diferenciadoras propuestas, existen múltiples áreas de mejora y expansión que pueden implementarse en versiones futuras, con el objetivo de ofrecer una experiencia aún más completa y competitiva. Entre las mejoras previstas se encuentran:

- Implementación de imágenes de perfil personalizadas: permitir a los usuarios subir y modificar su foto de perfil, lo que mejoraría la personalización y la identificación dentro de la plataforma.
- Envío de imágenes dentro del chat: habilitar la posibilidad de enviar y recibir imágenes en tiempo real, enriqueciendo la experiencia de conversación
- Envío de mensajes de audio: incorporar la función de grabar y enviar audios, facilitando una forma de comunicación más rápida y expresiva, especialmente en situaciones en las que escribir no es práctico.
- Compartir ubicación con Google Maps: permitir el envío de la ubicación del usuario directamente desde la aplicación utilizando los servicios de Google Maps, ideal para encuentros o coordinación entre usuarios.
- Desarrollo de una inteligencia artificial propia: sustituir la integración actual por una IA desarrollada internamente, optimizada para el contexto específico de la aplicación y capaz de ofrecer respuestas más coherentes, contextuales y personalizadas.
- Implementación de servidores ya que la forma en la que se abren chats es pública permitir crear servidores como públicos o como privados para que los usuarios decidan quién les puede hablar

Estas funcionalidades no han sido implementadas en la versión actual debido a limitaciones económicas, ya que muchas de estas características requieren servicios

externos de pago, mayor almacenamiento en la nube, uso de APIs premium o infraestructura adicional.

Glosario

A

- Algoritmo

Conjunto de pasos o instrucciones definidos que permiten resolver un problema o realizar una tarea. En ChatoChat, los algoritmos se utilizan, por ejemplo, para gestionar la lógica de emparejamiento aleatorio entre usuarios o procesar respuestas en el chat con la IA.

- Android

Sistema operativo de código abierto desarrollado por Google para dispositivos móviles. ChatoChat está desarrollado específicamente para dispositivos Android, por lo que se han utilizado herramientas compatibles como Android Studio y Kotlin.

- Android Studio

IDE (entorno de desarrollo integrado) oficial para aplicaciones Android. Permite compilar, probar y depurar apps móviles. Se eligió por su integración con el SDK de Android, su potente emulador y su soporte para Kotlin.

- API (Application Programming Interface)

Conjunto de funciones que permiten que distintas aplicaciones se comuniquen entre sí. ChatoChat utiliza APIs como Firebase o Google Maps para añadir funcionalidades como autenticación o compartir ubicación.

- Autenticación

Proceso mediante el cual se verifica la identidad de un usuario. Se implementa en ChatoChat mediante Firebase Authentication, permitiendo el registro y acceso seguro con correo electrónico y contraseña.

B

- Backend

Parte de la aplicación que gestiona la lógica del servidor, bases de datos y funciones invisibles para el usuario. En este proyecto, el backend se construye principalmente con Firebase.

- Base de Datos NoSQL

Modelo de base de datos que no se organiza en tablas como las bases SQL tradicionales. Firebase Realtime Database es una base de datos NoSQL, ideal para la sincronización en tiempo real.

- Bug

Error o fallo en el código que impide que una función se ejecute correctamente. Durante el desarrollo de ChatoChat se han corregido múltiples bugs, especialmente en la funcionalidad de chat en tiempo real.

C

- Chat Aleatorio

Función que permite conversar con personas desconocidas. En ChatoChat, esta característica promueve la socialización y requiere mecanismos de seguridad adicionales.

- Chat IA (Inteligencia Artificial)

Módulo que permite conversar con un asistente virtual. Utiliza modelos de lenguaje para simular una conversación humana. Esta funcionalidad en ChatoChat busca ofrecer compañía e interacción incluso cuando no haya usuarios conectados.

- Cifrado

Técnica de codificación de datos para proteger la información. En ChatoChat se usa para asegurar que las conversaciones privadas no puedan ser interceptadas o leídas por terceros.

- Cloud (Nube)

Infraestructura digital que permite almacenar y procesar datos a través de internet. Firebase ofrece servicios en la nube como bases de datos y almacenamiento de archivos.

- **Compatibilidad**

Capacidad de una aplicación para ejecutarse correctamente en distintos dispositivos y versiones del sistema operativo. ChatoChat es compatible con Android 8.0 (Oreo) en adelante.

D

- **Depuración (Debugging)**

Proceso de identificar y corregir errores en el código. Se ha realizado frecuentemente en ChatoChat, utilizando herramientas del entorno Android Studio.

- **Diseño Adaptativo (Responsive Design)**

Diseño que se ajusta a distintos tamaños de pantalla. ChatoChat implementa esta técnica para ofrecer una experiencia uniforme en smartphones de diversas resoluciones.

- **Diseño UX/UI**

UX se refiere a la experiencia del usuario y UI a la interfaz visual. ChatoChat ha sido diseñado considerando ambos conceptos para facilitar su uso y hacer la navegación agradable.

E

- **Emulador**

Programa que simula el comportamiento de un dispositivo móvil. Permite probar apps sin necesidad de un teléfono físico. Android Studio incluye un emulador potente que se ha utilizado extensivamente en ChatoChat.

- **Envío de Notificaciones Push**

Tecnología que permite enviar mensajes directamente al dispositivo del usuario. En ChatoChat se utiliza para avisar sobre mensajes nuevos incluso cuando la app está cerrada.

- **Escalabilidad**

Capacidad de una app para crecer en funcionalidad y usuarios sin perder rendimiento. ChatoChat está diseñado con buenas prácticas de programación para permitir su futura expansión.

F

- Figma

Herramienta de diseño de interfaces colaborativa en la nube. Se utilizó para prototipar y diseñar la interfaz gráfica de ChatoChat antes del desarrollo en Android Studio.

- Firebase

Plataforma de desarrollo móvil de Google que ofrece servicios como bases de datos, autenticación, almacenamiento y notificaciones. Es el backend principal de ChatoChat.

- Firebase Authentication

Módulo de Firebase que permite gestionar usuarios, registro e inicio de sesión. Se usa para mantener segura la entrada a la app.

- Firebase Cloud Messaging (FCM)

Servicio que permite el envío de notificaciones push a los dispositivos. ChatoChat utiliza FCM para notificar sobre nuevos mensajes o actividades relevantes.

- Firebase Realtime Database

Base de datos NoSQL en tiempo real. En ChatoChat, se usa para guardar mensajes, usuarios, contactos y más, actualizando la información al instante entre todos los dispositivos conectados.

G

- GitHub

Plataforma de desarrollo colaborativo y control de versiones. Aunque no mencionado directamente, es habitual utilizar GitHub para gestionar versiones del código fuente de una app como ChatoChat.

H

- Hosting

Servicio de alojamiento de archivos o bases de datos en la nube. Firebase actúa como proveedor de hosting para la lógica de backend de ChatoChat.

I

- IA (Inteligencia Artificial)

Tecnología que permite simular comportamientos humanos, como mantener una conversación. En ChatoChat se incorpora un módulo IA para ofrecer una experiencia social incluso sin otros usuarios presentes.

- Identificador Único (UID)

Código que identifica de forma única a cada usuario. Firebase genera un UID automáticamente al registrar un usuario, asegurando que cada cuenta sea distinta.

- Interfaz de Usuario (UI)

Conjunto de pantallas, botones y elementos visuales con los que interactúa el usuario. Se diseñó cuidadosamente en ChatoChat para ser clara, intuitiva y visualmente atractiva.

K

- KANBAN

Metodología ágil de gestión de proyectos basada en tableros visuales. Fue la metodología elegida para el desarrollo de ChatoChat por su enfoque visual y adaptativo.

- Kotlin

Lenguaje de programación oficial para el desarrollo de aplicaciones Android. Fue elegido por su sintaxis moderna, interoperabilidad con Java y compatibilidad con Firebase.

L

- Login (Inicio de Sesión)

Proceso por el cual un usuario accede a su cuenta mediante sus credenciales. En ChatoChat se realiza mediante Firebase Authentication.

M

- Mensajería Instantánea

Sistema que permite el intercambio de mensajes en tiempo real. Es la función principal de ChatoChat, diseñada para ser rápida y segura.

- Moderación Automática

Sistema que detecta y actúa sobre comportamientos inapropiados. Se incluye en ChatoChat para prevenir el mal uso, especialmente en chats aleatorios.

- Modelo Relacional

Representación de cómo se relacionan las tablas en una base de datos. Aunque Firebase no usa bases relacionales, se elaboró un modelo para estructurar la información lógicamente durante el diseño.

N

- Notificaciones Push

Mensajes enviados desde el servidor que aparecen en el dispositivo móvil del usuario. Son esenciales en apps de mensajería como ChatoChat para mantener la comunicación activa.

P

- Prototipo

Versión preliminar de una aplicación usada para probar conceptos de diseño antes del desarrollo. En ChatoChat se desarrolló un prototipo en Figma.

- Privacidad Digital

Protección de los datos personales del usuario. Se considera en cada aspecto de ChatoChat, especialmente en el almacenamiento y transmisión de mensajes.

R

- Realtime (Tiempo Real)

Característica que permite ver cambios o mensajes instantáneamente. ChatoChat usa Firebase Realtime Database para sincronizar los mensajes sin demora perceptible.

- Reporte de Usuario

Funcionalidad que permite señalar comportamientos inapropiados. Se implementa en ChatoChat para garantizar un entorno seguro.

S

- Seguridad Informática

Conjunto de medidas y tecnologías destinadas a proteger los sistemas y datos de accesos no autorizados. ChatoChat incluye cifrado, autenticación y validaciones para garantizar la seguridad.

Bibliografía

1. Firebase. (s.f.). Firebase. Recuperado el 10 de Abril de 2025, de <https://firebase.google.com/?hl=es-419>
2. Android Developers. (s.f.). Android Studio: Introducción. Recuperado el 10 de Abril de 2025, de <https://developer.android.com/studio/intro?hl=es-419>
3. OpenAI. (s.f.). ChatGPT. Recuperado el 10 de Abril de 2025, de <https://chatgpt.com/>
4. Anthropic. (s.f.). Claude AI. Recuperado el 10 de Mayo de 2025, de <https://claude.ai/login?returnTo=%2F%3F>
5. Kotlin. (s.f.). Kotlin Programming Language. Recuperado el 10 de abril de 2025, de <https://kotlinlang.org/>
6. OpenAI. (s.f.). OpenAI API. Recuperado el 16 de Mayo de 2025, de <https://openai.com/index/openai-api/>