**Documentación Profesional — AmigoIA: Asistente Virtual Personalizado**

Autores:  
Carol Nicol Clavijo Bonilla, Adrián David Bravo Montoya

Programa: Ingeniería de Software – Universitaria de Colombia

Bogotá D.C., 2025

# 1. Planteamiento del proyecto

El proyecto AmigoIA consiste en una aplicación móvil desarrollada en Flutter que permite a los usuarios interactuar con una inteligencia artificial (Gemini 2.5 Flash) como si fuera un “mejor amigo virtual”.  
  
Su objetivo principal es brindar compañía y apoyo emocional básico, además de permitir que los usuarios guarden y revisen sus conversaciones en cualquier momento.

# 2. Objetivo general

Diseñar e implementar una aplicación móvil en Flutter que permita a los usuarios comunicarse con una inteligencia artificial como un “mejor amigo virtual”, integrando almacenamiento local de conversaciones y una experiencia personalizada.

# 3. Objetivos específicos

1. Definir y estructurar la base de datos local para gestionar usuarios, conversaciones y mensajes.  
2. Implementar pantallas funcionales: registro, lista de conversaciones y chat.  
3. Integrar la API de Gemini 2.5 Flash para generar respuestas en tiempo real.  
4. Garantizar una interfaz sencilla y una experiencia amigable para el usuario.  
5. Realizar pruebas y correcciones para asegurar la calidad del sistema.

# 4. Alcance

El alcance del proyecto contempla el diseño, desarrollo e implementación de una aplicación móvil en Flutter que:  
- Permita a los usuarios registrarse e iniciar sesión.  
- Gestione conversaciones y mensajes mediante una base de datos local.  
- Integre un motor de IA (Gemini 2.5 Flash) para generar respuestas personalizadas.  
- Ofrezca una interfaz de usuario intuitiva con pantallas de registro, lista de conversaciones y chat.  
- Sea ejecutable en dispositivos móviles Android.

# 5. Justificación

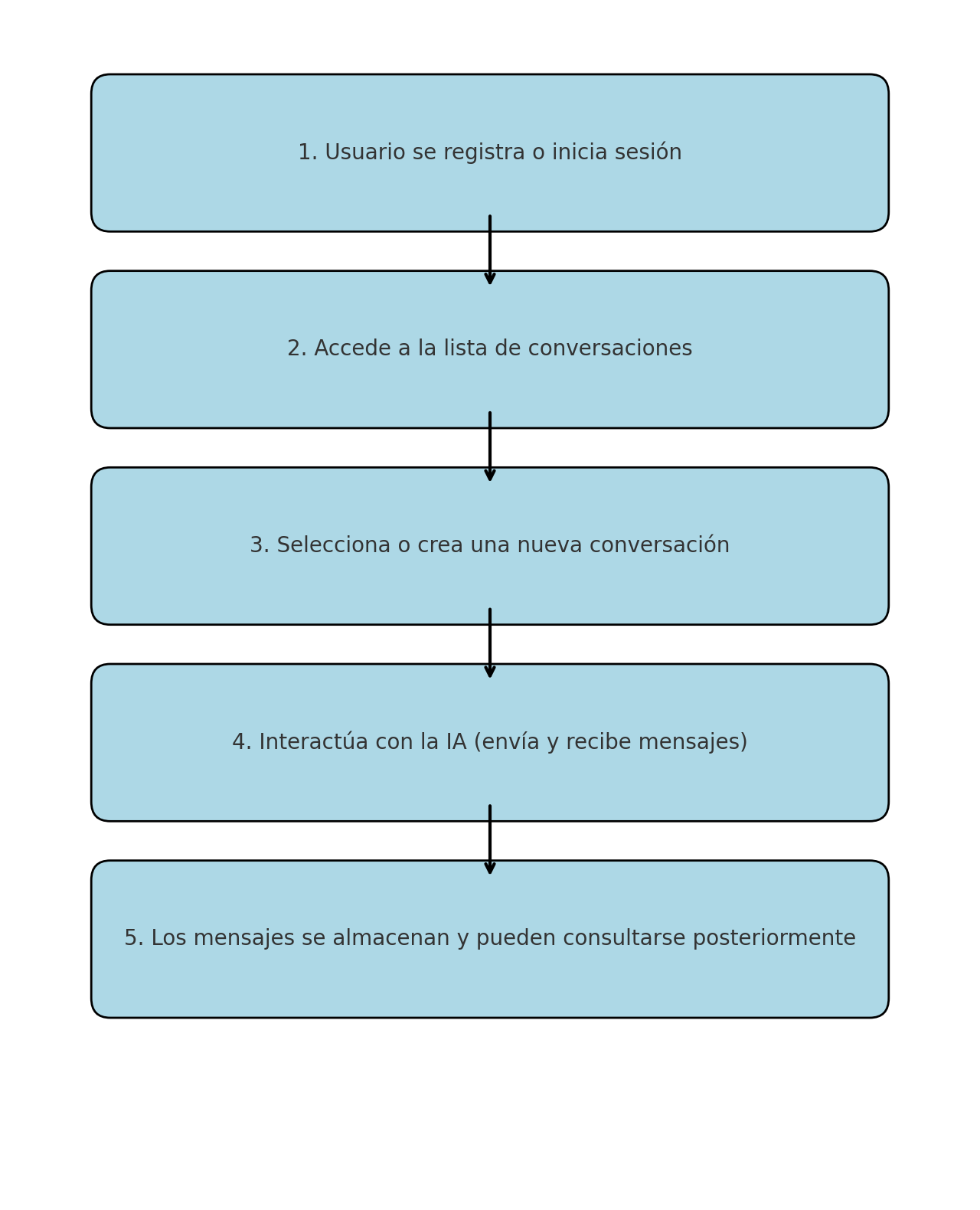
Las aplicaciones de acompañamiento emocional y asistencia virtual están en auge debido a la creciente necesidad de bienestar digital. Sin embargo, muchas carecen de personalización y almacenamiento local de datos.  
  
AmigoIA se diferencia al:  
- Personalizar la interacción refiriéndose al usuario por su nombre.  
- Permitir guardar conversaciones sin depender exclusivamente de servicios en la nube.  
- Facilitar un acceso rápido y amigable mediante una interfaz sencilla.

# 6. Productos entregables

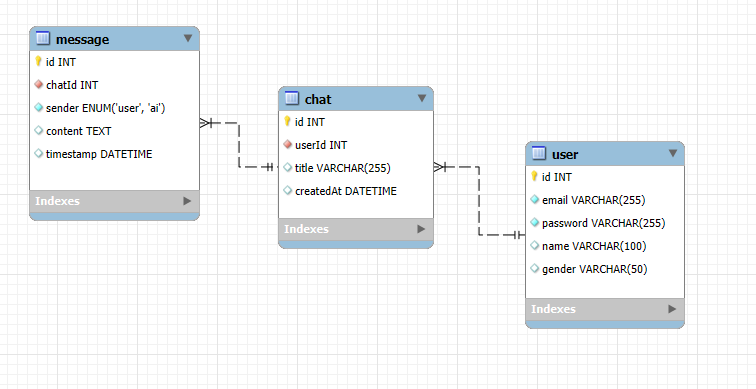
1. Documento técnico y de diseño (esta documentación)   
2. Aplicación móvil funcional desarrollada en Flutter.   
3. Base de datos local (Hive o SQLite) con usuarios, conversaciones y mensajes.   
4. Cronograma de desarrollo y pruebas.

# 7. Mapa de proceso (Descripción)

Proceso principal (resumen):  
1. Usuario se registra o inicia sesión. → 2. Accede a la lista de conversaciones. → 3. Selecciona o crea una nueva conversación. → 4. Interactúa con la IA (envía y recibe mensajes). → 5. Los mensajes se almacenan y pueden consultarse posteriormente.



# 8. MER (Modelo Entidad-Relación)



# 9. Requerimientos funcionales

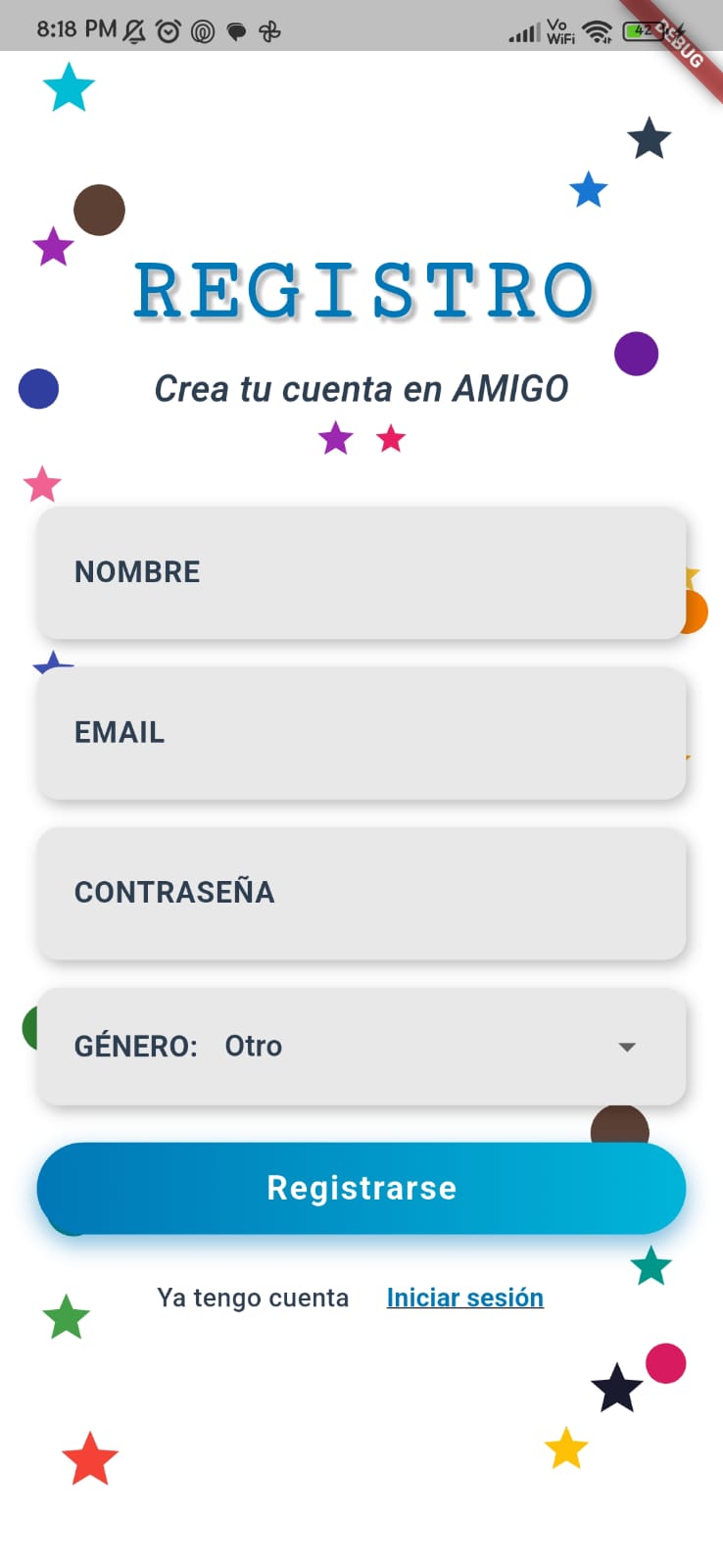
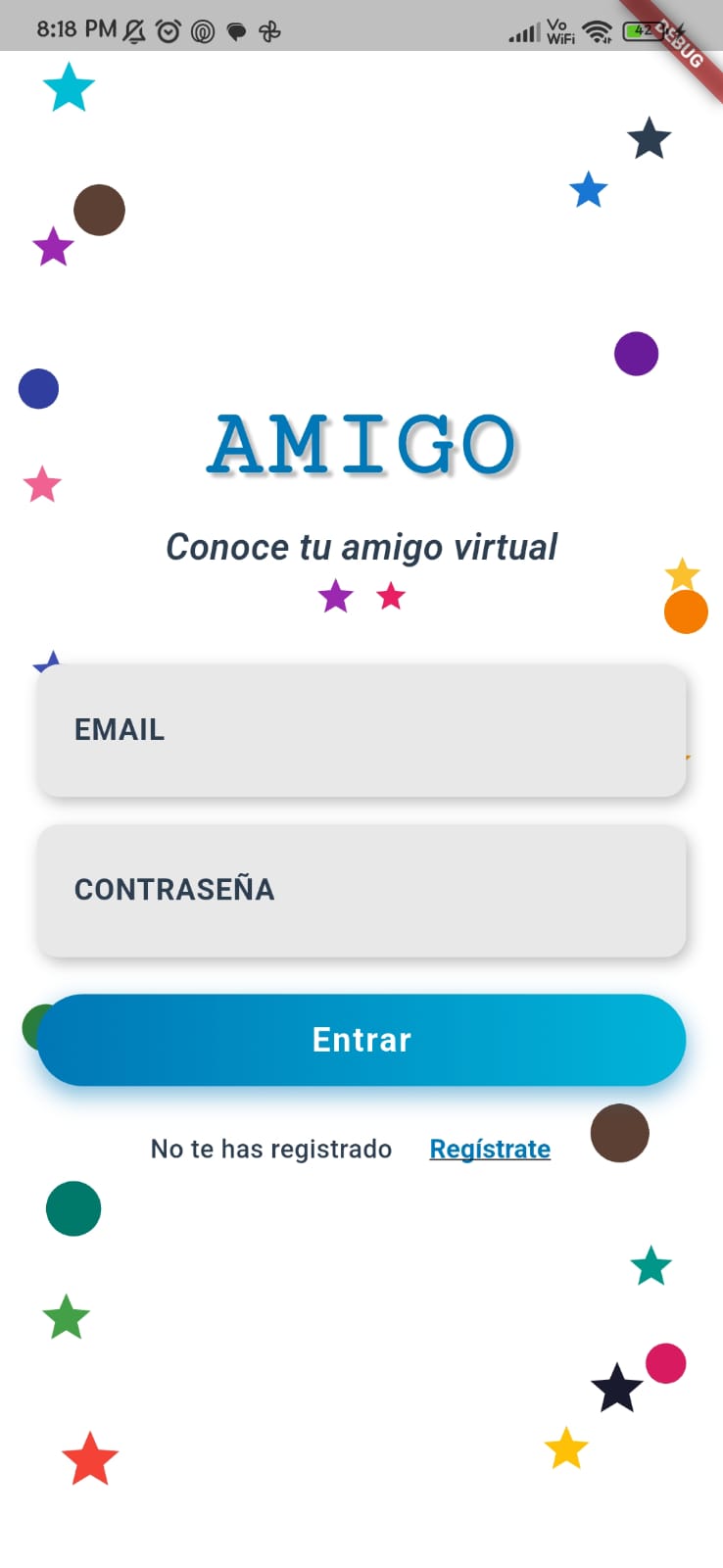
- RF1: Permitir el registro e inicio de sesión de usuarios.  
- RF2: Guardar conversaciones y mensajes en la base de datos local.  
- RF3: Recuperar y mostrar las conversaciones previas.  
- RF4: Integrar la API de Gemini 2.5 Flash para responder en tiempo real.  
- RF5: Personalizar la interacción utilizando el nombre del usuario.  
- RF6: Proporcionar pantallas funcionales: registro, lista de conversaciones y chat.

# 10. Requerimientos no funcionales

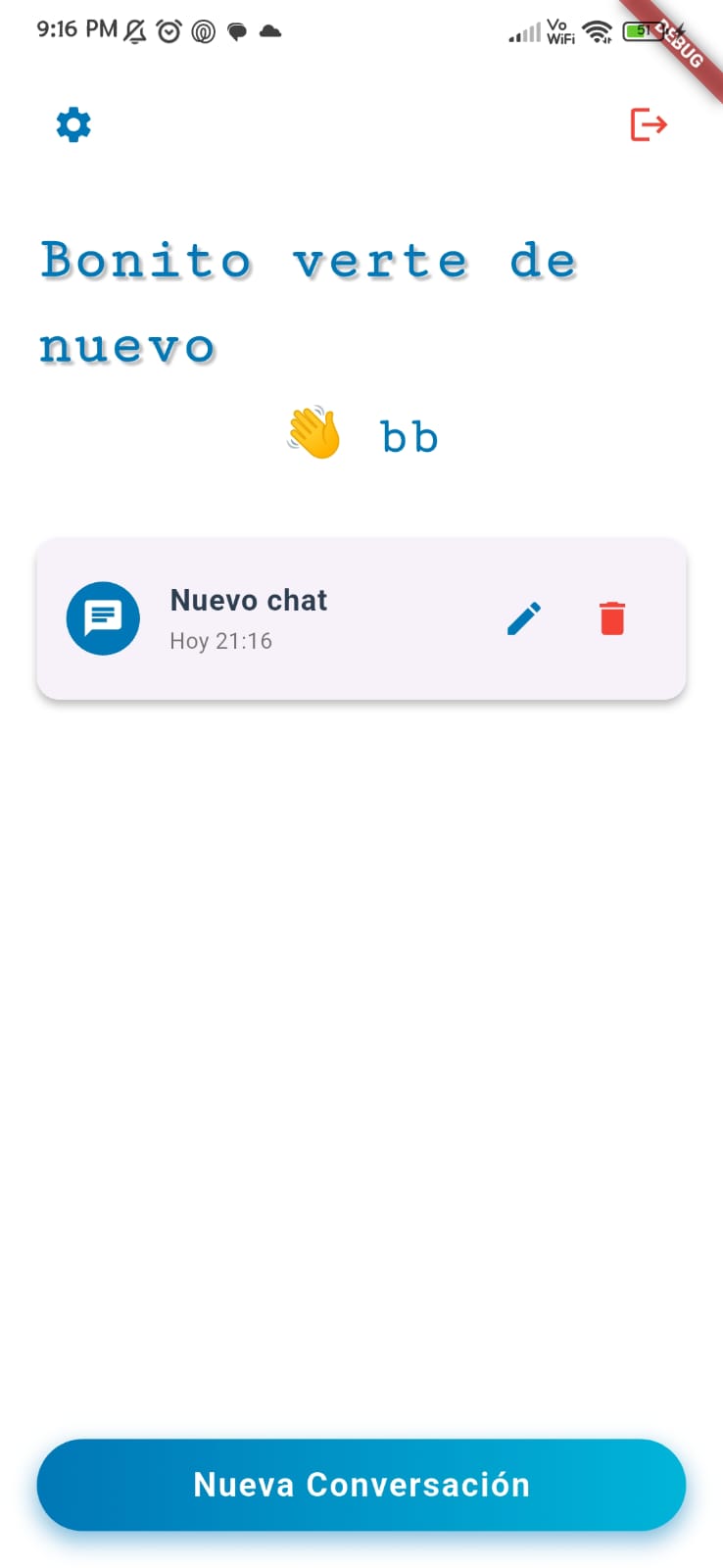
- RNF1: La aplicación debe ser ligera y ejecutarse con fluidez en dispositivos Android de gama media.  
- RNF2: El almacenamiento local debe garantizar consistencia e integridad de datos.  
- RNF3: La interfaz debe ser intuitiva y accesible, siguiendo lineamientos de diseño responsivo.  
- RNF4: El tiempo de respuesta de la IA no debe superar los 3 segundos en condiciones normales de red.  
- RNF5: El sistema debe ser seguro en la gestión de datos personales del usuario.

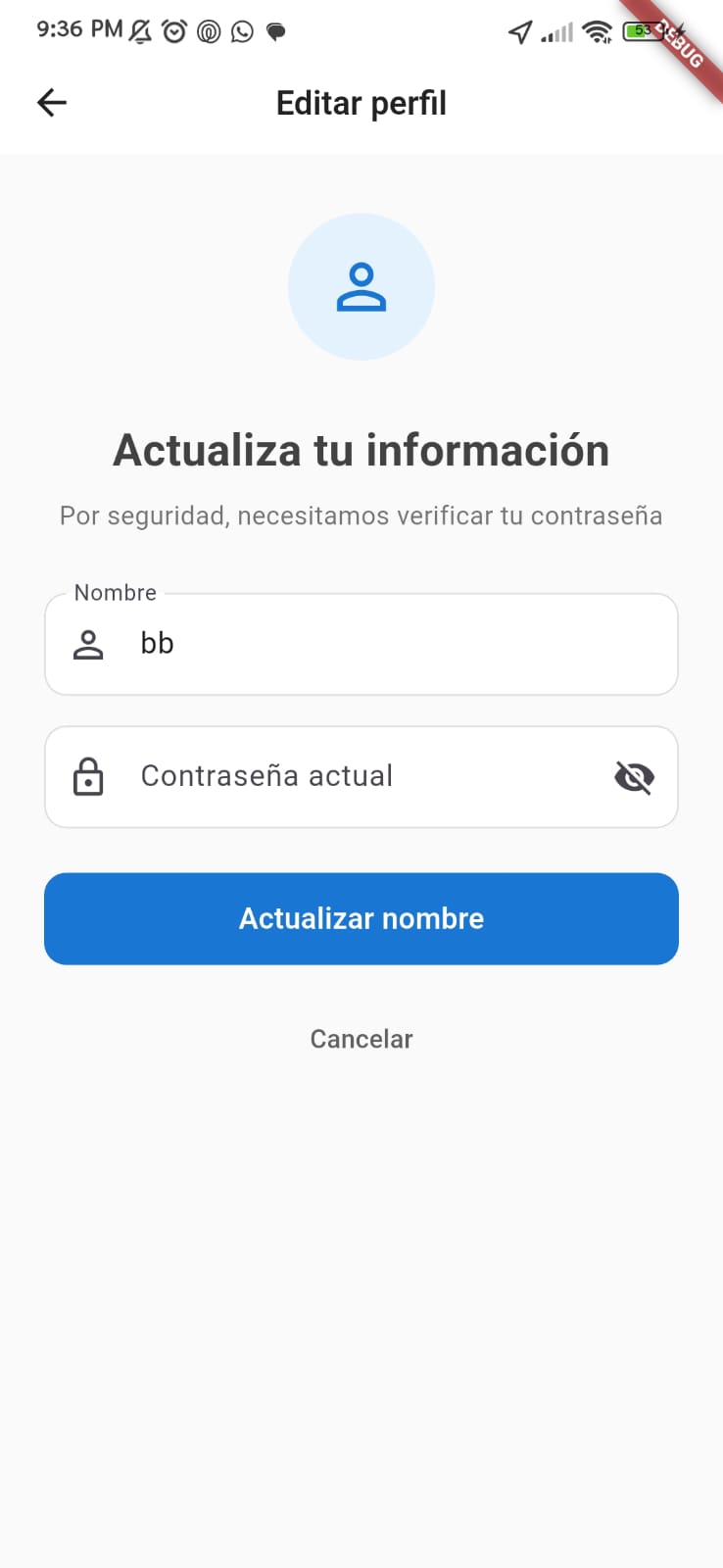
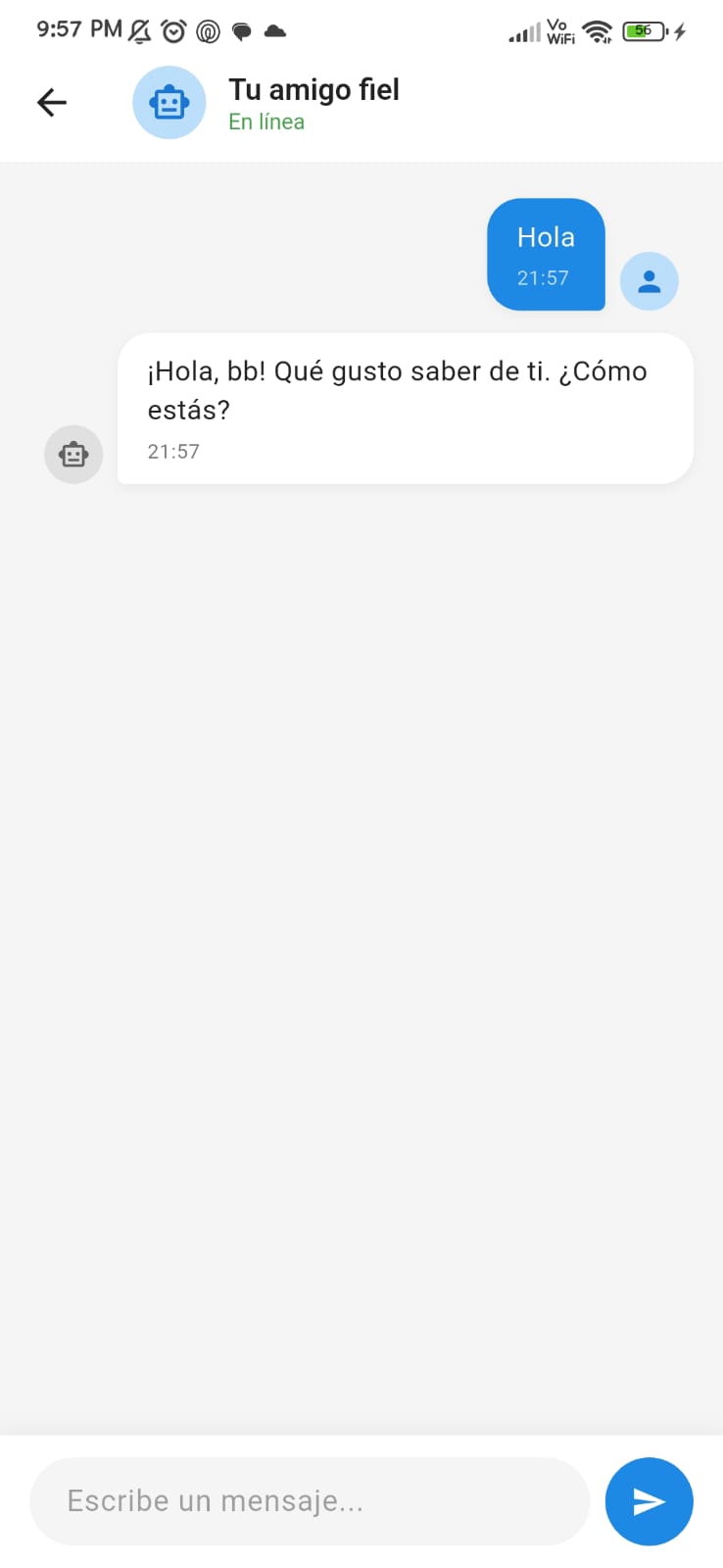
# 11. Interfaz — Diseño y ejemplos

- Pantalla de Registro: campos de usuario, correo, contraseña. Validaciones básicas.



- Pantalla de Lista de Conversaciones: listado de chats previos. Botón para crear nuevo chat.



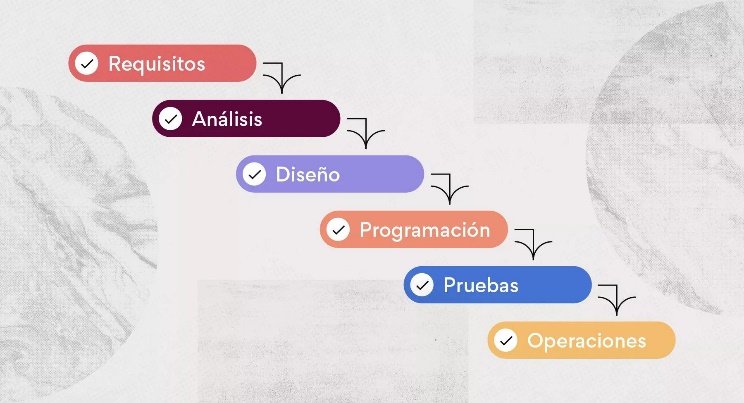
  
- Pantalla de Chat: visualización tipo burbuja, envío y recepción de mensajes en tiempo real.  


# 12. Cronograma y entregables

Periodo: 23 de septiembre – 1 de octubre de 2025  
  
| Fecha | Actividad | Responsable |  
|---------|---------------------------------------------------|-------------|  
| 23 sep | Diseño de base de datos y pantallas principales | Ambos |  
| 24 sep | Configuración de proyecto Flutter y dependencias | Adrián |  
| 25 sep | Implementación de base de datos local | Adrián |  
| 26 sep | Desarrollo de pantalla de registro de usuario | Carol |  
| 27 sep | Desarrollo de pantalla de lista de conversaciones | Carol |  
| 28 sep | Desarrollo de pantalla de chat (UI básica) | Carol |  
| 29 sep | Integración con la API de Gemini 2.5 Flash | Adrián |  
| 30 sep | Guardado y recuperación de mensajes + pruebas | Adrián |  
| 1 oct | Mejora de interfaz y preparación para sustentación| Ambos |

# 13. Metodología

Se empleará la metodología en cascada, ya que permite una planificación lineal y estructurada en fases: análisis, diseño, implementación, pruebas y despliegue.



# 14. Método

Se aplicarán también principios de Scrum, como organización de tareas por sprints cortos, retroalimentación constante y entregas incrementales del sistema.

# 15. Preparación de sustentación

Material a presentar (1-2 oct):  
- Documento final.  
- Modelo Entidad-Relación (insertado en sección 8).  
- Demo en Android Studio mostrando: registro, creación de conversación y chat con la IA.  
- Cronograma de actividades.  
- Aplicación en ejecución.