**Plan paso a paso:**

**SEMANA 1: CIMIENTOS DEL PROYECTO**

**Lunes - Día 1: Configuración del Entorno**

**Objetivo:** Preparar todas las herramientas y entornos necesarios para el desarrollo.

1. **9:00 - 10:30** - Instalación de herramientas básicas:
   * Instalar Visual Studio Code
   * Instalar Git y configurar usuario
   * Instalar Node.js (v16+) y npm
2. **10:30 - 12:30** - Instalación de entorno Flutter:
   * Descargar e instalar Flutter SDK
   * Instalar Android Studio
   * Configurar emulador Android
   * Ejecutar Flutter doctor y resolver problemas
3. **13:30 - 15:30** - Configuración de repositorio:
   * Crear cuenta GitHub (si no existe)
   * Crear repositorio "avatar-chat-app"
   * Clonar repositorio localmente
   * Crear estructura básica de carpetas:
   * 1/backend - API y servicios
   * 2/mobile - Aplicación Flutter

3/docs - Documentación

1. **15:30 - 17:00** - Verificación de instalación:
   * Crear aplicación "Hello World" en Flutter
   * Probar emulador Android
   * Hacer commit inicial al repositorio

**Entregable:** Repositorio Git configurado con estructura básica y entorno de desarrollo funcional.

**Martes - Día 2: Configuración de Servicios en la Nube**

**Objetivo:** Establecer cuentas y acceso a todos los servicios externos necesarios.

1. **9:00 - 10:30** - Configuración de OpenAI:
   * Crear cuenta en OpenAI Platform
   * Generar y guardar API key
   * Probar API con petición simple usando Postman
   * Documentar límites y costos
2. **10:30 - 12:30** - Configuración de ElevenLabs:
   * Crear cuenta en ElevenLabs
   * Explorar Voice Lab y generar API key
   * Probar API con petición simple usando Postman
   * Documentar límites y costos
3. **13:30 - 15:30** - Configuración de D-ID/HeyGen:
   * Crear cuenta en D-ID o HeyGen
   * Explorar funcionalidades de avatar
   * Generar y guardar API key
   * Probar API con petición simple usando Postman
4. **15:30 - 17:00** - Configuración de Firebase:
   * Crear proyecto en Firebase Console
   * Configurar Firebase Authentication (email/password)
   * Configurar Firestore Database
   * Configurar Storage
   * Descargar configuración (serviceAccountKey.json)

**Entregable:** Documento con todas las API keys y configuraciones guardadas en un lugar seguro.

**Miércoles - Día 3: Estructura Backend Básica**

**Objetivo:** Crear la estructura inicial del servidor backend.

1. **9:00 - 10:30** - Inicialización del proyecto Node.js:

BASH

1cd backend

2npm init -y

3npm install express cors dotenv axios multer firebase-admin

4npm install @elevenlabs/node-api openai

5npm install --save-dev nodemon

1. **10:30 - 12:30** - Configuración de estructura de backend:
   * Crear archivo **.env** con API keys
   * Crear estructura de carpetas:
   * 1/config - Configuraciones
   * 2/controllers - Lógica de negocio
   * 3/routes - Endpoints API
   * 4/services - Conexión con servicios externos
   * 5/middlewares - Middlewares

6/utils - Utilidades

* + Agregar **.gitignore** para node\_modules, .env, etc.

1. **13:30 - 15:30** - Implementación del servidor básico:
   * Crear **index.js** con servidor Express
   * Configurar middlewares básicos (cors, json)
   * Implementar ruta de prueba (/health)
   * Configurar dotenv para variables de entorno
2. **15:30 - 17:00** - Implementación de conexión Firebase:
   * Crear **/config/firebase.js**
   * Inicializar Firebase Admin SDK
   * Configurar acceso a Firestore y Storage
   * Probar conexión

**Entregable:** Servidor Express básico funcionando con conexión a Firebase confirmada.

**Jueves - Día 4: Conexión de Servicios Backend**

**Objetivo:** Integrar los servicios de IA en el backend.

1. **9:00 - 10:30** - Implementación del servicio OpenAI:
   * Crear **/services/openai.js**
   * Implementar función básica de generación de respuestas
   * Probar con prompt simple
2. **10:30 - 12:30** - Implementación de controladores de usuario:
   * Crear **/controllers/userController.js**
   * Implementar registro y login
   * Conectar con Firebase Authentication
   * Probar creación de usuario
3. **13:30 - 15:30** - Configuración de rutas API:
   * Crear **/routes/auth.js** para autenticación
   * Crear **/routes/chat.js** para mensajería
   * Crear **/routes/avatar.js** para gestión de avatar
   * Conectar rutas en **index.js**
4. **15:30 - 17:00** - Pruebas de endpoints básicos:
   * Probar ruta de registro con Postman
   * Probar ruta de login con Postman
   * Documentar respuestas y comportamiento

**Entregable:** Backend con rutas de autenticación funcionando y conexión a OpenAI.

**Viernes - Día 5: Planificación de App Móvil**

**Objetivo:** Diseñar y estructurar la aplicación móvil.

1. **9:00 - 10:30** - Diseño de wireframes:
   * Crear wireframes de pantallas principales:
     + Onboarding/Login
     + Creación de avatar (pasos)
     + Chat principal
     + Configuración
   * Usar Figma o herramienta similar
2. **10:30 - 12:30** - Inicialización del proyecto Flutter:

BASH

1cd mobile

2flutter create .

* + Configurar dependencias en **pubspec.yaml**:

YAML

1dependencies:

2 flutter:

3 sdk: flutter

4 http: ^0.13.5

5 provider: ^6.0.5

6 shared\_preferences: ^2.1.0

7 firebase\_core: ^2.7.0

8 firebase\_auth: ^4.2.9

9 cloud\_firestore: ^4.4.3

10 firebase\_storage: ^11.0.14

11 image\_picker: ^0.8.7

12 audio\_recorder: ^1.0.3

13 path\_provider: ^2.0.13

14 flutter\_dotenv: ^5.0.2

15 flutter\_markdown: ^0.6.14

1. **13:30 - 15:30** - Configuración de estructura de la app:
   * Crear estructura de carpetas:
   * 1/lib
   * 2 /models - Modelos de datos
   * 3 /screens - Pantallas
   * 4 /widgets - Componentes reutilizables
   * 5 /services - Servicios (API, etc.)
   * 6 /providers - Estado global
   * 7 /utils - Utilidades

8 /constants - Constantes

* + Configurar tema básico y colores de la app

1. **15:30 - 17:00** - Implementación de navegación básica:
   * Crear esqueleto de navegación entre pantallas
   * Implementar tema claro/oscuro
   * Probar en emulador

**Entregable:** Proyecto Flutter inicializado con estructura y navegación básica.

**SEMANA 2: PROCESO DE PERSONALIZACIÓN**

**Lunes - Día 6: Pantalla de Onboarding**

**Objetivo:** Implementar el flujo de onboarding y registro.

1. **9:00 - 10:30** - Diseño de pantalla de bienvenida:
   * Crear **SplashScreen** con logo y animación
   * Implementar transición automática
   * Crear pantallas de introducción deslizables
2. **10:30 - 12:30** - Implementación de autenticación:
   * Crear **AuthService** para gestionar Firebase Auth
   * Implementar pantalla de registro
   * Implementar pantalla de login
   * Validar formularios
3. **13:30 - 15:30** - Integración con Firebase Auth:
   * Conectar formularios con Firebase
   * Gestionar estados de carga y error
   * Implementar persistencia de sesión
4. **15:30 - 17:00** - Implementación de navegación post-login:
   * Crear flujo condicional (nuevo usuario vs. usuario existente)
   * Implementar verificación de avatar existente
   * Pruebas completas del flujo

**Entregable:** Flujo de onboarding y autenticación completamente funcional.

**Martes - Día 7: Captura de Fotos**

**Objetivo:** Implementar sistema de captura y procesamiento de fotos.

1. **9:00 - 10:30** - Diseño de interfaz de captura:
   * Crear pantalla de instrucciones para fotos
   * Diseñar UI para captura de múltiples fotos
   * Implementar contador de fotos requeridas
2. **10:30 - 12:30** - Implementación de captura con cámara:
   * Integrar **image\_picker** para acceso a cámara
   * Implementar captura de foto
   * Mostrar vista previa de foto tomada
   * Permitir repetir captura
3. **13:30 - 15:30** - Implementación de selección de galería:
   * Permitir selección desde galería
   * Implementar validación básica de imágenes
   * Crear vista de cuadrícula de imágenes seleccionadas
4. **15:30 - 17:00** - Subida de imágenes a backend:
   * Crear servicio de subida a Firebase Storage
   * Implementar barra de progreso de subida
   * Guardar referencias en Firestore
   * Manejar errores de subida

**Entregable:** Sistema completo de captura y almacenamiento de fotos.

**Miércoles - Día 8: Sistema de Grabación de Voz**

**Objetivo:** Implementar captura y procesamiento de muestras de voz.

1. **9:00 - 10:30** - Diseño de interfaz de grabación:
   * Crear pantalla de instrucciones para grabación
   * Diseñar UI con visualizador de ondas de audio
   * Implementar botones de control (grabar, pausar, reproducir)
2. **10:30 - 12:30** - Implementación de grabación:
   * Integrar plugin de grabación de audio
   * Implementar lógica de grabación
   * Mostrar tiempo transcurrido
   * Validar duración mínima (30 segundos)
3. **13:30 - 15:30** - Implementación de reproducción:
   * Permitir reproducir grabación para verificación
   * Implementar controles de reproducción
   * Opción para volver a grabar
4. **15:30 - 17:00** - Subida de audio a backend:
   * Crear servicio de subida a Firebase Storage
   * Implementar compresión básica de audio
   * Guardar referencia en Firestore
   * Manejar errores de subida

**Entregable:** Sistema completo de grabación y almacenamiento de voz.

**Jueves - Día 9: Cuestionario de Personalidad**

**Objetivo:** Implementar sistema de captura de rasgos de personalidad.

1. **9:00 - 10:30** - Diseño de cuestionario:
   * Crear modelo de datos para personalidad
   * Diseñar preguntas clave (15-20)
   * Implementar estructura de pasos/wizard
2. **10:30 - 12:30** - Implementación de formulario básico:
   * Crear componentes de pregunta/respuesta
   * Implementar diferentes tipos de input:
     + Escala deslizable
     + Selección múltiple
     + Entrada de texto
   * Validación de respuestas
3. **13:30 - 15:30** - Implementación de navegación entre preguntas:
   * Crear sistema de pasos con indicador de progreso
   * Permitir navegación adelante/atrás
   * Guardar respuestas temporalmente
4. **15:30 - 17:00** - Procesamiento y almacenamiento:
   * Crear modelo final de personalidad
   * Enviar datos al backend
   * Guardar en Firestore
   * Implementar continuación al siguiente paso

**Entregable:** Cuestionario de personalidad funcional con almacenamiento de datos.

**Viernes - Día 10: Backend para Personalización**

**Objetivo:** Implementar endpoints para procesar datos de personalización.

1. **9:00 - 10:30** - Implementación de controlador de avatar:
   * Crear **/controllers/avatarController.js**
   * Implementar creación de perfil de avatar
   * Conectar con Firestore
2. **10:30 - 12:30** - Implementación de rutas para personalización:
   * Crear endpoints para:
     + **/api/avatars/create**
     + **/api/avatars/:id/upload-photos**
     + **/api/avatars/:id/upload-voice**
     + **/api/avatars/:id/personality**
   * Documentar parámetros y respuestas
3. **13:30 - 15:30** - Implementación de procesamiento de imágenes:
   * Usar Multer para gestionar subidas
   * Procesar imágenes (compresión, validación)
   * Almacenar en Firebase Storage
   * Devolver URLs públicas
4. **15:30 - 17:00** - Implementación de procesamiento de audio:
   * Gestionar subida de archivos de audio
   * Validar formato y duración
   * Almacenar en Firebase Storage
   * Actualizar documento de avatar en Firestore

**Entregable:** API completa para gestión de datos de personalización.

**SEMANA 3: INTEGRACIÓN DE IA Y GENERACIÓN**

**Lunes - Día 11: Integración con OpenAI**

**Objetivo:** Implementar sistema completo de generación de respuestas personalizadas.

1. **9:00 - 10:30** - Mejora del servicio OpenAI:
   * Expandir **/services/openai.js**
   * Implementar manejo de contexto conversacional
   * Crear estructura de prompts dinámicos
2. **10:30 - 12:30** - Implementación de sistema de personalidad:
   * Crear función para generar prompt de sistema basado en personalidad
   * Implementar formato para incluir rasgos, intereses y estilo
   * Probar con diferentes perfiles
3. **13:30 - 15:30** - Implementación de endpoint de chat:
   * Crear **/controllers/chatController.js**
   * Implementar lógica para procesar mensajes
   * Conectar con servicio OpenAI
   * Guardar mensajes en Firestore
4. **15:30 - 17:00** - Pruebas y ajustes:
   * Crear script de prueba para diferentes escenarios
   * Ajustar temperatura y parámetros según resultados
   * Documentar mejores configuraciones
   * Optimizar manejo de tokens

**Entregable:** Sistema de generación de respuestas personalizadas completamente funcional.

**Martes - Día 12: Clonación de Voz con ElevenLabs**

**Objetivo:** Implementar sistema de clonación y generación de voz personalizada.

1. **9:00 - 10:30** - Implementación del servicio ElevenLabs:
   * Crear **/services/elevenlabs.js**
   * Implementar función para clonar voz desde archivo
   * Gestionar respuesta y almacenar ID de voz
2. **10:30 - 12:30** - Implementación de síntesis de voz:
   * Crear función para generar audio desde texto
   * Configurar parámetros (estabilidad, similitud)
   * Implementar almacenamiento de archivos de audio generados
   * Crear sistema de caché para respuestas frecuentes
3. **13:30 - 15:30** - Integración con el controlador de chat:
   * Modificar **/controllers/chatController.js** para usar ElevenLabs
   * Implementar flujo: texto → voz → respuesta
   * Manejar errores de generación
   * Implementar respuestas fallback
4. **15:30 - 17:00** - Pruebas y optimización:
   * Probar con diferentes muestras de voz
   * Ajustar parámetros para mejorar naturalidad
   * Implementar validación de calidad de audio
   * Documentar mejores prácticas para grabación

**Entregable:** Sistema de clonación y generación de voz personalizada completamente funcional.

**Miércoles - Día 13: Generación de Avatar Visual**

**Objetivo:** Implementar sistema de creación y animación de avatar visual.

1. **9:00 - 10:30** - Implementación del servicio D-ID/HeyGen:
   * Crear **/services/avatar.js**
   * Implementar función para crear avatar desde imágenes
   * Gestionar respuesta y almacenar ID de avatar
2. **10:30 - 12:30** - Implementación de generación de video:
   * Crear función para generar video con audio
   * Configurar parámetros de animación
   * Implementar almacenamiento de videos generados
   * Manejar procesamiento asincrónico (polling)
3. **13:30 - 15:30** - Integración con el controlador de chat:
   * Modificar **/controllers/chatController.js** para incluir avatar
   * Implementar flujo completo: texto → voz → avatar → respuesta
   * Gestionar tiempos de espera extendidos
   * Implementar respuestas progresivas
4. **15:30 - 17:00** - Pruebas y optimización:
   * Probar con diferentes imágenes
   * Ajustar parámetros para mejorar expresividad
   * Implementar sistema de caché para reducir generaciones
   * Crear estrategia para respuestas rápidas vs. completas

**Entregable:** Sistema de generación de avatar visual completamente funcional.

**Jueves - Día 14: Sistema de Almacenamiento de Perfiles**

**Objetivo:** Implementar estructura de datos completa para perfiles de avatar.

1. **9:00 - 10:30** - Diseño de esquema de datos:
   * Finalizar modelo de datos para avatares
   * Definir relaciones entre colecciones
   * Documentar estructura completa
2. **10:30 - 12:30** - Implementación de controladores CRUD:
   * Expandir **/controllers/avatarController.js**
   * Implementar operaciones:
     + Obtener perfil
     + Actualizar perfil
     + Eliminar perfil
   * Agregar validaciones
3. **13:30 - 15:30** - Implementación de lógica de versiones:
   * Crear sistema para mantener versiones de avatar
   * Implementar actualizaciones parciales
   * Gestionar referencias a recursos externos (D-ID, ElevenLabs)
4. **15:30 - 17:00** - Optimización de consultas:
   * Implementar índices en Firestore
   * Optimizar recuperación de perfiles
   * Implementar caché de perfiles frecuentes
   * Probar con múltiples perfiles simultáneos

**Entregable:** Sistema completo de gestión de perfiles de avatar con operaciones CRUD.

**Viernes - Día 15: Pruebas de Integración Backend**

**Objetivo:** Verificar el funcionamiento completo del flujo backend.

1. **9:00 - 10:30** - Preparación de entorno de pruebas:
   * Crear conjunto de datos de prueba
   * Configurar entorno aislado
   * Preparar scripts de prueba automatizados
2. **10:30 - 12:30** - Pruebas de flujo completo:
   * Probar ciclo: creación de avatar → personalización → chat
   * Documentar tiempos de respuesta
   * Identificar cuellos de botella
3. **13:30 - 15:30** - Optimización de rendimiento:
   * Implementar procesamientos en paralelo donde sea posible
   * Optimizar manejo de promesas
   * Implementar timeout adecuados
   * Reducir tamaño de payload
4. **15:30 - 17:00** - Documentación de API:
   * Crear documentación completa de endpoints
   * Incluir ejemplos de solicitud/respuesta
   * Documentar códigos de error
   * Crear colección de Postman para pruebas

**Entregable:** Backend con flujo completo probado y documentado, listo para integración con frontend.

**SEMANA 4: INTERFAZ DE CHAT PRINCIPAL**

**Lunes - Día 16: Pantalla de Creación de Avatar**

**Objetivo:** Implementar la pantalla final del proceso de creación de avatar.

1. **9:00 - 10:30** - Diseño de pantalla de resumen:
   * Crear UI para mostrar datos recopilados
   * Implementar visualización de fotos seleccionadas
   * Mostrar resumen de personalidad
   * Diseñar botón de confirmación
2. **10:30 - 12:30** - Implementación de pantalla de procesamiento:
   * Crear UI con animación de "creando avatar"
   * Implementar indicadores de progreso por etapa
   * Mostrar mensajes informativos del proceso
   * Gestionar estados de carga
3. **13:30 - 15:30** - Integración con backend:
   * Conectar con endpoint **/api/avatars/:id/generate**
   * Implementar manejo de respuesta asincrónica
   * Gestionar errores y reintentos
   * Mostrar vista previa cuando esté disponible
4. **15:30 - 17:00** - Implementación de transición a chat:
   * Crear animación de "avatar listo"
   * Mostrar mensaje de bienvenida del avatar
   * Implementar transición fluida a pantalla de chat
   * Guardar estado de finalización

**Entregable:** Flujo completo de creación de avatar con feedback visual y transición a chat.

**Martes - Día 17: Diseño de Interfaz de Chat**

**Objetivo:** Implementar la pantalla principal de conversación.

1. **9:00 - 10:30** - Diseño de pantalla de chat:
   * Crear layout principal:
     + Área de visualización de avatar (superior)
     + Lista de mensajes (central)
     + Área de entrada (inferior)
   * Implementar tema visual coherente
2. **10:30 - 12:30** - Implementación de lista de mensajes:
   * Crear componente **MessageList**
   * Implementar burbujas de chat diferentes para usuario y avatar
   * Añadir timestamps y estados de mensaje
   * Implementar scroll automático
3. **13:30 - 15:30** - Implementación de entrada de mensaje:
   * Crear componente **MessageInput**
   * Implementar campo de texto expansible
   * Añadir botón de envío
   * Implementar validación básica
   * Añadir botón de grabación de voz (opcional)
4. **15:30 - 17:00** - Implementación de header y opciones:
   * Crear barra superior con información del avatar
   * Implementar menú de opciones
   * Añadir botones de acceso rápido a configuración
   * Implementar navegación a otras secciones

**Entregable:** Interfaz de chat completa con todos los componentes visuales.

**Miércoles - Día 18: Sistema de Envío/Recepción**

**Objetivo:** Implementar la lógica de comunicación bidireccional.

1. **9:00 - 10:30** - Implementación de servicio de chat:
   * Crear **ChatService** en Flutter
   * Implementar métodos para enviar/recibir mensajes
   * Conectar con API backend
   * Gestionar estados de mensajes
2. **10:30 - 12:30** - Implementación de estado de conversación:
   * Crear **ChatProvider** usando Provider o Bloc
   * Implementar gestión de estado de mensajes
   * Manejar estados de carga
   * Implementar actualización en tiempo real
3. **13:30 - 15:30** - Implementación de envío de mensajes:
   * Conectar **MessageInput** con **ChatProvider**
   * Implementar lógica de envío
   * Mostrar estado "enviando" temporalmente
   * Actualizar lista al recibir confirmación
   * Manejar errores de envío
4. **15:30 - 17:00** - Implementación de recepción de mensajes:
   * Mostrar indicador "escribiendo..."
   * Implementar recepción y renderizado de respuestas
   * Gestionar mensajes largos
   * Implementar notificaciones de nuevos mensajes

**Entregable:** Sistema funcional de envío y recepción de mensajes con gestión de estado.

**Jueves - Día 19: Visualización del Avatar**

**Objetivo:** Implementar el componente de visualización del avatar animado.

1. **9:00 - 10:30** - Diseño del componente de avatar:
   * Crear **AvatarView** para mostrar el avatar
   * Implementar diferentes estados visuales:
     + Neutral/en espera
     + Escuchando
     + Pensando/procesando
     + Hablando/respondiendo
2. **10:30 - 12:30** - Implementación de reproductor de video:
   * Integrar **video\_player** para respuestas
   * Implementar carga y buffering
   * Manejar ciclo de vida del video
   * Implementar controles básicos (pausar/reanudar)
3. **13:30 - 15:30** - Integración con flujo de chat:
   * Conectar **AvatarView** con **ChatProvider**
   * Implementar cambios de estado según flujo de conversación
   * Cargar videos de respuesta
   * Sincronizar estado visual con mensajes
4. **15:30 - 17:00** - Optimización de experiencia visual:
   * Implementar transiciones suaves entre estados
   * Añadir animaciones de idle para estado neutral
   * Optimizar carga de recursos
   * Probar en diferentes tamaños de pantalla

**Entregable:** Componente de visualización de avatar completamente funcional e integrado con el chat.

**Viernes - Día 20: Reproducción de Audio**

**Objetivo:** Implementar sistema de reproducción de respuestas de voz.

1. **9:00 - 10:30** - Implementación de reproductor de audio:
   * Integrar **audioplayers** o similar
   * Implementar controles de reproducción
   * Manejar ciclo de vida del audio
   * Implementar precarga de archivos
2. **10:30 - 12:30** - Sincronización con avatar:
   * Sincronizar reproducción de audio con video
   * Implementar controles unificados
   * Manejar diferentes duraciones
   * Implementar fallbacks si falla alguno
3. **13:30 - 15:30** - Integración con mensajes:
   * Conectar reproducción con visualización de mensajes
   * Implementar destacado de mensaje actual
   * Añadir controles en burbujas de chat
   * Permitir reproducción de mensajes antiguos
4. **15:30 - 17:00** - Mejoras de accesibilidad y UX:
   * Implementar controles de volumen
   * Añadir opciones de velocidad de reproducción
   * Implementar modo "solo texto"
   * Probar experiencia completa

**Entregable:** Sistema de audio completamente funcional y sincronizado con avatar y mensajes.

**SEMANA 5: REFINAMIENTO DE EXPERIENCIA**

**Lunes - Día 21: Persistencia de Conversaciones**

**Objetivo:** Implementar almacenamiento y recuperación de historial de chat.

1. **9:00 - 10:30** - Implementación de modelo de conversación:
   * Finalizar modelo de datos **Conversation**
   * Implementar métodos de serialización/deserialización
   * Crear utilidades para gestión de conversaciones
2. **10:30 - 12:30** - Implementación de almacenamiento local:
   * Usar **shared\_preferences** para guardar mensajes recientes
   * Implementar lógica de guardado automático
   * Crear sistema de caché para respuestas
   * Gestionar limitaciones de espacio
3. **13:30 - 15:30** - Implementación de sincronización con Firestore:
   * Crear **ConversationService** para sincronización
   * Implementar carga de historial al iniciar
   * Configurar sincronización periódica
   * Manejar conflictos de sincronización
4. **15:30 - 17:00** - Implementación de búsqueda y filtrado:
   * Crear interfaz para ver historial
   * Implementar búsqueda de mensajes
   * Añadir filtros por fecha
   * Permitir eliminar conversaciones

**Entregable:** Sistema completo de persistencia y recuperación de conversaciones.

**Martes - Día 22: Animaciones y Transiciones**

**Objetivo:** Mejorar la experiencia visual con animaciones fluidas.

1. **9:00 - 10:30** - Implementación de animaciones de UI:
   * Añadir animaciones a elementos de interfaz:
     + Aparición/desaparición de mensajes
     + Transición entre pantallas
     + Botones y controles interactivos
   * Usar **AnimationController** para control preciso
2. **10:30 - 12:30** - Mejora de transiciones entre estados:
   * Implementar transiciones suaves para estados del avatar
   * Añadir indicadores de actividad animados
   * Mejorar feedback visual de acciones
   * Implementar "respiración" y movimientos sutiles en idle
3. **13:30 - 15:30** - Implementación de gestos y micro-interacciones:
   * Añadir respuestas táctiles a interacciones
   * Implementar gestos (swipe, long press, etc.)
   * Crear animaciones de reacción a eventos
   * Mejorar comportamiento de scroll
4. **15:30 - 17:00** - Optimización de rendimiento:
   * Evaluar rendimiento de animaciones
   * Identificar y corregir jank (saltos)
   * Optimizar para dispositivos de gama baja
   * Documentar configuraciones óptimas
   * Implementar animaciones condicionales según capacidad del dispositivo
   * Probar en diferentes dispositivos

**Entregable:** Interfaz con animaciones fluidas y transiciones pulidas en toda la aplicación.

**Miércoles - Día 23: Mejora de Naturalidad**

**Objetivo:** Refinar la personalidad y naturalidad de las respuestas del chatbot.

1. **9:00 - 10:30** - Refinamiento de prompts de OpenAI:
   * Revisar y mejorar estructura de prompts
   * Implementar variantes para diferentes contextos
   * Crear "memoria" de conversaciones anteriores
   * Agregar ejemplos específicos de estilo de comunicación
2. **10:30 - 12:30** - Implementación de variaciones de respuesta:
   * Modificar backend para incluir factor de aleatoriedad controlada
   * Implementar diferentes "humores" o estados de ánimo
   * Crear sistema de frases características
   * Evitar repeticiones excesivas
3. **13:30 - 15:30** - Mejora de manejo contextual:
   * Implementar referencias a mensajes anteriores
   * Mejorar seguimiento de temas de conversación
   * Crear capacidad de retomar conversaciones interrumpidas
   * Implementar detección de cambios abruptos de tema
4. **15:30 - 17:00** - Pruebas y ajustes:
   * Realizar conversaciones de prueba extensas
   * Documentar comportamientos inconsistentes
   * Ajustar parámetros de temperatura y respuesta
   * Implementar logging para análisis de calidad

**Entregable:** Sistema conversacional con respuestas más naturales, contextuales y variadas.

**Jueves - Día 24: Sistema de Memoria Conversacional**

**Objetivo:** Implementar un sistema que permita al chatbot recordar información importante.

1. **9:00 - 10:30** - Diseño del sistema de memoria:
   * Definir tipos de memoria:
     + Memoria a corto plazo (conversación actual)
     + Memoria a medio plazo (sesiones recientes)
     + Memoria a largo plazo (información permanente)
   * Diseñar estructura de datos para almacenamiento
2. **10:30 - 12:30** - Implementación de extracción de información:
   * Crear sistema para identificar datos importantes:
     + Preferencias mencionadas
     + Información personal compartida
     + Eventos o fechas discutidas
   * Implementar almacenamiento estructurado
3. **13:30 - 15:30** - Integración con generación de respuestas:
   * Modificar sistema de prompts para incluir memoria relevante
   * Implementar priorización de información
   * Crear sistema de "olvido" controlado
   * Implementar referencias naturales a información pasada
4. **15:30 - 17:00** - Implementación de persistencia:
   * Guardar memoria en Firestore
   * Implementar sincronización entre dispositivos
   * Crear interfaz para visualizar/editar memoria (opcional)
   * Pruebas de recuperación de información

**Entregable:** Sistema de memoria funcional que mejora la continuidad y personalización de conversaciones.

**Viernes - Día 25: Feedback Visual**

**Objetivo:** Mejorar la comunicación de estados y acciones al usuario.

1. **9:00 - 10:30** - Implementación de indicadores de estado:
   * Crear indicadores visuales para:
     + "Escuchando" - cuando el usuario está escribiendo
     + "Pensando" - durante generación de respuesta
     + "Hablando" - durante reproducción de respuesta
     + "Esperando" - en estado idle
   * Diseñar transiciones entre estados
2. **10:30 - 12:30** - Implementación de feedback táctil y sonoro:
   * Añadir vibración sutil para eventos clave
   * Implementar sonidos ambientales según contexto
   * Crear efectos de sonido para acciones
   * Hacer todo configurable (activar/desactivar)
3. **13:30 - 15:30** - Mejora de notificaciones:
   * Implementar notificaciones in-app
   * Crear toast messages para confirmaciones
   * Diseñar alertas para errores importantes
   * Implementar banners informativos
4. **15:30 - 17:00** - Implementación de guías contextuales:
   * Crear tooltips para funcionalidades clave
   * Implementar sugerencias de uso
   * Añadir ayuda contextual
   * Diseñar primera experiencia guiada

**Entregable:** Sistema completo de feedback visual, táctil y sonoro que comunica claramente el estado de la aplicación.

**SEMANA 6: PERSONALIZACIÓN AVANZADA**

**Lunes - Día 26: Optimización de Carga**

**Objetivo:** Mejorar tiempos de respuesta y experiencia durante cargas.

1. **9:00 - 10:30** - Implementación de precarga de recursos:
   * Crear sistema de precarga de assets frecuentes
   * Implementar carga diferida de recursos pesados
   * Optimizar inicialización de componentes
   * Medir y documentar tiempos de carga
2. **10:30 - 12:30** - Implementación de estados intermedios:
   * Diseñar respuestas rápidas mientras se genera contenido
   * Implementar placeholders animados
   * Crear transiciones progresivas de carga
   * Mostrar contenido parcial cuando esté disponible
3. **13:30 - 15:30** - Optimización de backend:
   * Implementar procesamiento en paralelo donde sea posible
   * Crear sistema de caché para respuestas frecuentes
   * Optimizar consultas a bases de datos
   * Implementar timeout adaptativo
4. **15:30 - 17:00** - Implementación de modo offline:
   * Crear funcionalidad básica offline
   * Implementar sincronización cuando vuelve la conexión
   * Mostrar estado de conectividad
   * Priorizar funciones críticas en modo offline

**Entregable:** Aplicación con tiempos de carga optimizados y experiencia fluida incluso durante operaciones pesadas.

**Martes - Día 27: Gestos y Accesibilidad**

**Objetivo:** Mejorar la usabilidad y accesibilidad de la aplicación.

1. **9:00 - 10:30** - Implementación de gestos avanzados:
   * Añadir soporte para:
     + Swipe para acciones rápidas
     + Pinch para zoom (donde aplique)
     + Long press para opciones adicionales
     + Shake para reportar problemas
   * Documentar gestos disponibles
2. **10:30 - 12:30** - Mejora de navegación:
   * Implementar atajos de teclado (dispositivos con teclado)
   * Crear navegación por voz (opcional)
   * Optimizar flujos para reducir pasos frecuentes
   * Implementar "back" inteligente
3. **13:30 - 15:30** - Implementación de accesibilidad:
   * Configurar correctamente etiquetas para TalkBack/VoiceOver
   * Mejorar contraste y legibilidad
   * Implementar ajustes de tamaño de texto
   * Añadir descripciones para elementos visuales
4. **15:30 - 17:00** - Pruebas de accesibilidad:
   * Probar con herramientas de accesibilidad
   * Verificar navegación con TalkBack/VoiceOver
   * Implementar mejoras según hallazgos
   * Documentar consideraciones de accesibilidad

**Entregable:** Aplicación con controles gestuales intuitivos y soporte de accesibilidad mejorado.

**Miércoles - Día 28: Ajustes de Personalidad**

**Objetivo:** Implementar opciones para ajustar la personalidad del avatar después de la creación.

1. **9:00 - 10:30** - Diseño de interfaz de ajustes de personalidad:
   * Crear pantalla de "Personalidad"
   * Diseñar controles para ajustar rasgos
   * Implementar vista previa de cambios
   * Crear sistema de perfiles/presets
2. **10:30 - 12:30** - Implementación de ajustes de rasgos:
   * Crear controles deslizables para rasgos principales:
     + Extroversión/introversión
     + Formalidad/informalidad
     + Emotividad/racionalidad
     + Concisión/detalle
   * Implementar guardado de cambios
3. **13:30 - 15:30** - Implementación de frases y expresiones:
   * Crear sistema para añadir/editar frases características
   * Implementar campo para expresiones favoritas
   * Añadir ejemplos predefinidos
   * Conectar con backend para actualización
4. **15:30 - 17:00** - Implementación de estilos de comunicación:
   * Crear presets de estilos de comunicación
   * Implementar personalización detallada
   * Añadir vista previa de estilo
   * Probar diferentes configuraciones

**Entregable:** Sistema completo para ajustar y refinar la personalidad del avatar después de la creación inicial.

**Jueves - Día 29: Configuración de Temas**

**Objetivo:** Implementar sistema para personalizar temas de conversación.

1. **9:00 - 10:30** - Diseño de interfaz de temas:
   * Crear pantalla de "Temas de conversación"
   * Diseñar visualización de categorías
   * Implementar sistema de preferencias
   * Crear interfaz para añadir temas personalizados
2. **10:30 - 12:30** - Implementación de preferencias temáticas:
   * Crear sistema de temas favoritos
   * Implementar niveles de interés (me encanta/me gusta/neutral/evitar)
   * Diseñar selección por categorías
   * Conectar con sistema de generación de respuestas
3. **13:30 - 15:30** - Implementación de conocimientos específicos:
   * Crear sistema para añadir "hechos" sobre temas
   * Implementar base de conocimientos personalizada
   * Diseñar interfaz para gestionar información
   * Conectar con backend para almacenamiento
4. **15:30 - 17:00** - Integración con sistema conversacional:
   * Modificar generación de prompts para incluir preferencias temáticas
   * Implementar detección de temas en conversación
   * Crear sistema de sugerencias basadas en temas
   * Probar conversaciones con diferentes configuraciones

**Entregable:** Sistema completo para personalizar temas de conversación y conocimientos específicos del avatar.

**Viernes - Día 30: Opciones de Apariencia**

**Objetivo:** Implementar personalización visual del avatar y la interfaz.

1. **9:00 - 10:30** - Diseño de opciones de apariencia del avatar:
   * Crear pantalla de "Apariencia"
   * Diseñar controles para ajustes visuales
   * Implementar vista previa en tiempo real
   * Crear presets visuales
2. **10:30 - 12:30** - Implementación de ajustes de avatar:
   * Crear opciones para:
     + Fondos de avatar
     + Filtros visuales
     + Iluminación
     + Tamaño en pantalla
   * Conectar con servicios de generación
3. **13:30 - 15:30** - Implementación de temas de interfaz:
   * Crear sistema de temas (claro, oscuro, personalizado)
   * Implementar selección de colores de acento
   * Diseñar opciones de estilos de burbuja de chat
   * Añadir opciones de fuente y tamaño
4. **15:30 - 17:00** - Implementación de guardado de configuración:
   * Crear sistema para guardar configuraciones
   * Implementar presets compartibles
   * Añadir restauración de valores predeterminados
   * Probar sincronización entre dispositivos

**Entregable:** Sistema completo de personalización visual tanto del avatar como de la interfaz de usuario.

**SEMANA 7: OPTIMIZACIÓN Y PRUEBAS**

**Lunes - Día 31: Guardado de Configuraciones**

**Objetivo:** Implementar sistema robusto para persistencia de configuraciones.

1. **9:00 - 10:30** - Diseño del sistema de configuraciones:
   * Definir estructura de datos para configuraciones
   * Crear jerarquía de preferencias
   * Diseñar sistema de valores predeterminados
   * Planificar estrategia de migración para actualizaciones
2. **10:30 - 12:30** - Implementación de almacenamiento local:
   * Usar **shared\_preferences** para configuraciones básicas
   * Implementar **hive** para configuraciones complejas
   * Crear sistema de backup local
   * Implementar validación de configuraciones
3. **13:30 - 15:30** - Implementación de sincronización en la nube:
   * Crear sistema de sincronización con Firestore
   * Implementar resolución de conflictos
   * Diseñar estrategia para múltiples dispositivos
   * Añadir encriptación para datos sensibles
4. **15:30 - 17:00** - Implementación de gestión de configuraciones:
   * Crear pantalla de "Configuración"
   * Implementar importación/exportación
   * Añadir reset a valores predeterminados
   * Crear log de cambios de configuración

**Entregable:** Sistema completo de gestión de configuraciones con persistencia local y en la nube.

**Martes - Día 32: Relaciones y Contextos**

**Objetivo:** Implementar sistema que permita al avatar evolucionar según la interacción.

1. **9:00 - 10:30** - Diseño del sistema de relación:
   * Definir métricas de relación (familiaridad, tono, etc.)
   * Crear modelo de datos para seguimiento
   * Diseñar algoritmo de evolución de relación
   * Planificar impacto en interacciones
2. **10:30 - 12:30** - Implementación de seguimiento de interacciones:
   * Crear sistema para registrar patrones de uso
   * Implementar análisis básico de sentimiento
   * Diseñar detección de temas recurrentes
   * Conectar con sistema de memoria
3. **13:30 - 15:30** - Implementación de adaptación contextual:
   * Crear sistema para adaptar respuestas según relación
   * Implementar variaciones de tono según contexto
   * Diseñar respuestas prioritarias para temas importantes
   * Añadir referencias a interacciones pasadas
4. **15:30 - 17:00** - Implementación de visualización de relación:
   * Crear indicadores sutiles de estado de relación
   * Implementar feedback sobre evolución
   * Diseñar interfaz opcional para ver "estadísticas"
   * Probar evolución con diferentes escenarios

**Entregable:** Sistema que permite al avatar adaptar su comportamiento y respuestas según la evolución de la relación con el usuario.

**Miércoles - Día 33: Pruebas de Usuario**

**Objetivo:** Realizar pruebas con usuarios reales y recopilar feedback.

1. **9:00 - 10:30** - Preparación de protocolo de pruebas:
   * Definir escenarios de prueba
   * Crear guía de tareas para participantes
   * Diseñar formularios de feedback
   * Preparar entorno de pruebas
2. **10:30 - 12:30** - Reclutamiento y configuración:
   * Identificar 3-5 usuarios para pruebas
   * Preparar dispositivos de prueba
   * Configurar herramientas de grabación (opcional)
   * Crear versión de prueba de la aplicación
3. **13:30 - 15:30** - Realización de pruebas:
   * Conducir sesiones de prueba individuales
   * Tomar notas detalladas
   * Observar puntos de fricción
4. **15:30 - 17:00** - Análisis y documentación:
   * Compilar resultados de todas las pruebas
   * Identificar patrones comunes de problemas
   * Categorizar issues por severidad:
     + Críticos (bloquean uso)
     + Importantes (afectan experiencia)
     + Menores (molestias menores)
   * Crear documento de hallazgos
   * Priorizar mejoras para implementación

**Entregable:** Informe completo de usabilidad con problemas identificados y recomendaciones priorizadas.

**Jueves - Día 34: Correcciones de Usabilidad**

**Objetivo:** Implementar mejoras basadas en el feedback de usuarios.

1. **9:00 - 10:30** - Planificación de correcciones:
   * Revisar informe de pruebas de usuario
   * Crear lista de tareas priorizadas
   * Estimar esfuerzo para cada corrección
   * Seleccionar issues críticos e importantes para abordar primero
2. **10:30 - 12:30** - Corrección de problemas críticos:
   * Implementar soluciones para bloqueos identificados
   * Corregir flujos confusos
   * Mejorar etiquetas y textos de UI poco claros
   * Probar correcciones internamente
3. **13:30 - 15:30** - Mejoras de experiencia de usuario:
   * Implementar mejoras en navegación
   * Optimizar flujos frecuentes
   * Añadir ayudas contextuales
   * Mejorar feedback visual
4. **15:30 - 17:00** - Verificación de correcciones:
   * Realizar pruebas de regresión
   * Verificar que las correcciones no crearon nuevos problemas
   * Documentar cambios implementados
   * Preparar notas para posible segunda ronda de pruebas

**Entregable:** Primera ronda de mejoras implementadas basadas en feedback de usuarios con documentación de cambios.

**Viernes - Día 35: Optimización de Rendimiento**

**Objetivo:** Mejorar la eficiencia y velocidad de la aplicación.

1. **9:00 - 10:30** - Análisis de rendimiento:
   * Realizar profiling de la aplicación
   * Identificar cuellos de botella
   * Monitorear uso de memoria
   * Documentar métricas de línea base
2. **10:30 - 12:30** - Optimización de UI:
   * Implementar renderizado eficiente de listas
   * Optimizar jerarquía de widgets
   * Reducir reconstrucciones innecesarias
   * Implementar carga diferida de recursos pesados
3. **13:30 - 15:30** - Optimización de red y almacenamiento:
   * Implementar caching eficiente
   * Optimizar tamaño de payloads
   * Mejorar estrategia de descarga de assets
   * Reducir operaciones de lectura/escritura
4. **15:30 - 17:00** - Medición y documentación:
   * Realizar mediciones post-optimización
   * Documentar mejoras logradas
   * Identificar áreas para futuras optimizaciones
   * Actualizar guías de mejores prácticas

**Entregable:** Aplicación con rendimiento mejorado y documentación de optimizaciones realizadas.

**SEMANA 8: PULIDO FINAL Y LANZAMIENTO**

**Lunes - Día 36: Reducción de Latencia**

**Objetivo:** Minimizar tiempos de espera y mejorar la respuesta de la aplicación.

1. **9:00 - 10:30** - Análisis de latencia:
   * Medir tiempos de respuesta en puntos críticos:
     + Inicio de la aplicación
     + Generación de respuestas
     + Carga de recursos
   * Identificar operaciones más lentas
   * Establecer objetivos de mejora
2. **10:30 - 12:30** - Optimización de llamadas a API:
   * Implementar solicitudes en paralelo donde sea posible
   * Añadir timeout inteligente
   * Crear sistema de reintentos graduales
   * Implementar compresión de datos
3. **13:30 - 15:30** - Implementación de estrategias de precarga:
   * Crear sistema predictivo para contenido probable
   * Implementar carga anticipada de recursos
   * Diseñar pipeline de procesamiento eficiente
   * Priorizar recursos críticos
4. **15:30 - 17:00** - Implementación de feedback durante esperas:
   * Mejorar indicadores de progreso
   * Crear animaciones de carga atractivas
   * Implementar "esqueletos" de contenido
   * Mostrar mensajes informativos durante esperas largas

**Entregable:** Aplicación con tiempos de respuesta significativamente mejorados y mejor feedback durante operaciones lentas.

**Martes - Día 37: Pruebas en Diferentes Dispositivos**

**Objetivo:** Garantizar compatibilidad con diversa gama de dispositivos.

1. **9:00 - 10:30** - Preparación para pruebas multiplataforma:
   * Preparar suite de pruebas para diferentes dispositivos
   * Configurar herramientas de testing (Firebase Test Lab, etc.)
   * Crear escenarios de prueba específicos
   * Preparar matriz de dispositivos a probar
2. **10:30 - 12:30** - Pruebas en dispositivos Android:
   * Probar en al menos 3 tamaños de pantalla diferentes
   * Verificar en dispositivos de gama baja, media y alta
   * Probar diferentes versiones de Android (10, 11, 12, 13)
   * Documentar problemas específicos
3. **13:30 - 15:30** - Ajustes de responsividad:
   * Corregir problemas de layout
   * Optimizar para diferentes densidades de pantalla
   * Ajustar tamaños de fuente y elementos
   * Implementar layouts adaptativos
4. **15:30 - 17:00** - Optimización para dispositivos específicos:
   * Implementar ajustes para dispositivos problemáticos
   * Crear fallbacks para características no soportadas
   * Documentar requerimientos mínimos
   * Actualizar manifiesto de compatibilidad

**Entregable:** Aplicación probada y optimizada para funcionar correctamente en diversos dispositivos.

**Miércoles - Día 38: Corrección de Bugs**

**Objetivo:** Resolver problemas pendientes y mejorar estabilidad.

1. **9:00 - 10:30** - Revisión de bugs pendientes:
   * Compilar lista completa de bugs conocidos
   * Categorizar por severidad y área afectada
   * Priorizar para corrección
   * Asignar tiempo estimado
2. **10:30 - 12:30** - Corrección de bugs críticos:
   * Resolver problemas que causan crasheos
   * Corregir errores de conexión/sincronización
   * Arreglar problemas de persistencia de datos
   * Resolver fallos en flujos principales
3. **13:30 - 15:30** - Corrección de bugs secundarios:
   * Resolver problemas visuales
   * Corregir comportamientos inconsistentes
   * Arreglar problemas de navegación menores
   * Solucionar fallos en características secundarias
4. **15:30 - 17:00** - Pruebas de regresión:
   * Verificar que las correcciones no causaron nuevos problemas
   * Probar flujos principales completos
   * Validar estabilidad general
   * Actualizar lista de bugs

**Entregable:** Aplicación estable con bugs principales resueltos y documentación actualizada.

**Jueves - Día 39: Implementación de Feedback Final**

**Objetivo:** Realizar últimos ajustes basados en todo el feedback recibido.

1. **9:00 - 10:30** - Revisión final de feedback:
   * Recopilar todo el feedback pendiente
   * Identificar mejoras de alta prioridad y bajo esfuerzo
   * Crear lista final de cambios
   * Evaluar impacto vs. esfuerzo
2. **10:30 - 12:30** - Implementación de ajustes finales de UX:
   * Refinar textos y mensajes
   * Mejorar consistencia visual
   * Ajustar espaciado y alineación
   * Optimizar flujos frecuentes
3. **13:30 - 15:30** - Pulido visual final:
   * Revisar assets gráficos
   * Mejorar iconos y botones
   * Ajustar paleta de colores
   * Refinar animaciones
4. **15:30 - 17:00** - Mejoras finales de contenido:
   * Actualizar textos de ayuda
   * Mejorar mensajes de error
   * Refinar prompts predeterminados
   * Completar traducciones (si aplica)

**Entregable:** Versión pulida de la aplicación incorporando mejoras finales basadas en todo el feedback recibido.

**Viernes - Día 40: Lanzamiento**

**Objetivo:** Preparar y publicar la versión final de la aplicación.

1. **9:00 - 10:30** - Preparación para producción:
   * Configurar variables de entorno para producción
   * Asegurar claves de API finales
   * Revisar configuraciones de Firebase
   * Verificar límites y cuotas de servicios
2. **10:30 - 12:30** - Generación de build final:
   * Incrementar número de versión
   * Generar APK firmado / Bundle de Android
   * Ejecutar pruebas finales en build de producción
   * Preparar assets para Google Play
3. **13:30 - 15:30** - Publicación en Google Play:
   * Completar ficha de Play Store:
     + Capturas de pantalla
     + Descripción
     + Política de privacidad
     + Clasificación de contenido
   * Subir APK/Bundle a Google Play Console
   * Configurar lanzamiento (testing interno → producción)
4. **15:30 - 17:00** - Configuración de monitoreo post-lanzamiento:
   * Configurar Firebase Crashlytics
   * Implementar Google Analytics
   * Preparar dashboard de monitoreo
   * Crear plan de soporte inicial

**Entregable:** Aplicación publicada y disponible para descarga en Google Play Store con sistemas de monitoreo activos.

**RESUMEN DE ENTREGABLES PRINCIPALES**

**Semana 1: Cimientos del Proyecto**

* Repositorio Git configurado y entorno de desarrollo funcional
* Acceso a todas las APIs de IA (OpenAI, ElevenLabs, D-ID/HeyGen)
* Proyecto Firebase configurado (Auth, Firestore, Storage)
* Servidor backend básico con conexión a Firebase
* Proyecto Flutter inicializado con estructura y navegación básica

**Semana 2: Proceso de Personalización**

* Flujo completo de onboarding y autenticación
* Sistema de captura y almacenamiento de fotos
* Sistema de grabación y almacenamiento de voz
* Cuestionario de personalidad funcional
* API completa para gestión de datos de personalización

**Semana 3: Integración de IA y Generación**

* Sistema de generación de respuestas personalizadas
* Sistema de clonación y generación de voz
* Sistema de generación de avatar visual
* Estructura de datos completa para perfiles
* Backend con flujo completo probado y documentado

**Semana 4: Interfaz de Chat Principal**

* Flujo completo de creación de avatar con feedback visual
* Interfaz de chat completa con componentes visuales
* Sistema funcional de envío/recepción de mensajes
* Componente de visualización de avatar
* Sistema de audio sincronizado con avatar y mensajes

**Semana 5: Refinamiento de Experiencia**

* Sistema completo de persistencia de conversaciones
* Interfaz con animaciones y transiciones fluidas
* Sistema conversacional con respuestas naturales
* Sistema de memoria conversacional
* Sistema de feedback visual, táctil y sonoro

**Semana 6: Personalización Avanzada**

* Aplicación con tiempos de carga optimizados
* Controles gestuales y soporte de accesibilidad
* Sistema para ajustar personalidad post-creación
* Sistema para personalizar temas de conversación
* Sistema de personalización visual de avatar e interfaz

**Semana 7: Optimización y Pruebas**

* Sistema de gestión de configuraciones
* Sistema de adaptación según la relación usuario-avatar
* Informe de usabilidad con problemas identificados
* Mejoras implementadas basadas en feedback
* Aplicación con rendimiento optimizado

**Semana 8: Pulido Final y Lanzamiento**

* Aplicación con tiempos de respuesta mejorados
* Compatibilidad verificada en diversos dispositivos
* Aplicación estable con bugs principales resueltos
* Versión pulida con mejoras finales
* **Aplicación publicada en Google Play Store**