**✅ Artefactos Entregables para VecinIA (Vibe Coding Week)**

**1. 🎯 Goal / Concept Title**

**VecinIA: Red de Apoyo Vecinal con IA Conversacional para Mayores**

**2. 📚 Use Cases for Coding**

* Solicitud de ayuda vecinal (compañía, compras, gestiones).
* Recordatorios con opción de asistencia.
* IA conversacional empática para combatir la soledad.
* Valoración mutua entre mayores y voluntarios.
* Matching inteligente entre vecinos y mayores.

**3. 💡 Solution & Benefits**

* **Usabilidad**: Interfaz accesible con botones grandes y navegación sencilla.
* **Accesibilidad**: Diseño inclusivo para personas mayores con dificultades visuales o cognitivas.
* **Interacción**: Fluida y emocional gracias a IA conversacional.
* **Beneficios tangibles**: Reducción de la soledad, mejora en la gestión de tareas cotidianas.
* **Beneficios intangibles**: Fortalecimiento del tejido social, empoderamiento comunitario.

**4. 🌟 User Experience**

* Navegación intuitiva.
* Diseño cálido y amigable.
* IA que adapta el tono y contenido según el estado emocional del usuario.
* Valoraciones que fomentan confianza y mejora continua.

**5. 🚀 Innovativeness**

* IA conversacional adaptada a mayores.
* Matching vecinal con algoritmos de afinidad.
* Integración de recordatorios con solicitud de ayuda.
* Valoración bidireccional para mejorar la experiencia.

**6. 📈 Business Opportunity / Market Potential**

* Aplicable en ciudades, zonas rurales y comunidades cerradas.
* Potencial de colaboración con ONGs, ayuntamientos y servicios sociales.
* Escalable a otros colectivos vulnerables (personas con discapacidad, etc.).

**7. 🛠️ Ease of Implementation**

* MVP funcional con prototipo en Figma.
* Infraestructura basada en tecnologías accesibles (Flutter, Firebase, OpenAI).
* Bajo coste inicial y alta viabilidad técnica.

**8. 💰 Financial Feasibility**

* Costes iniciales: desarrollo, diseño, IA.
* Costes operativos: mantenimiento, soporte, servidores.
* Modelo de financiación:x colaboración institucional, patrocinios, licencias sociales.
* ROI: mejora en calidad de vida, reducción de costes sociales.

**9. 🔄 Scalable / Reusable**

* Reutilizable en otras apps de asistencia comunitaria.
* Modular para integrar nuevas funcionalidades.
* Adaptable a otros idiomas y culturas.

**10. 📦 Otros Entregables**

* ✅ Prototipo navegable en Figma (en proceso).
* ✅ Pitch deck en PowerPoint.
* ✅ Mockups visuales de pantallas clave.
* ✅ Documento PDF con roadmap y fases del proyecto.

**Fases del MVP en 6 Días**

**🔹 Día 1: Definición y Diseño**

* Validación rápida del concepto con stakeholders (ONGs, usuarios mayores).
* Definición de funcionalidades mínimas:
  + Chat IA
  + Solicitud de ayuda
  + Recordatorios con opción de asistencia
* Diseño de wireframes y mockups (pantallas ya generadas).
* Preparación del diagrama de arquitectura (ya disponible).

**🔹 Día 2: Configuración Técnica**

* Configuración del entorno de desarrollo (Flutter + Firebase).
* Integración básica de OpenAI para IA conversacional.
* Estructura inicial de base de datos (usuarios, solicitudes, valoraciones).
* Preparación del backend con endpoints esenciales.

**🔹 Día 3: Desarrollo del Frontend**

* Implementación de la app móvil para mayores:
  + Pantalla principal
  + Crear recordatorio
  + Solicitar ayuda
* Implementación de la app para voluntarios:
  + Perfil y preferencias
  + Aceptar/rechazar solicitudes

**🔹 Día 4: Integración de IA y Matching**

* Activación del motor de IA conversacional con respuestas empáticas.
* Algoritmo básico de matching por zona y disponibilidad.
* Integración de notificaciones push para solicitudes.

**🔹 Día 5: Validación y Ajustes**

* Pruebas funcionales con usuarios simulados.
* Ajustes de accesibilidad (tamaño de letra, contraste).
* Revisión de flujo de navegación y experiencia de usuario.
* Preparación del prototipo navegable en Figma (en curso).

**🔹 Día 6: Entregables Finales**

* 🎥 Grabación del demo de 2 minutos.
* 📦 Entrega del código base en repositorio compartido.
* 📄 Relleno del MVP Submission Form con todos los artefactos:
  + Diagrama de arquitectura
  + Pitch deck
  + Prototipo navegable
  + Documento de solución y beneficios
  + Evaluación de impacto y escalabilidad

**Prompts para Windsurf – MVP VecinIA**

**🟦 1. Pantalla Principal para Mayores**

*“Crea una interfaz móvil accesible para personas mayores con tres botones grandes: ‘Conversar’, ‘Recordatorios’ y ‘Pedir ayuda’. Usa colores cálidos, tipografía clara y navegación sencilla.”*

**🟦 2. Pantalla de Solicitud de Ayuda**

*“Diseña una pantalla para que el usuario mayor pueda solicitar ayuda. Debe incluir botones grandes para tipo de ayuda (compañía, compra, farmacia), selector de hora, campo de notas, dirección y duración estimada.”*

**🟦 3. Pantalla de Crear Recordatorio**

*“Genera una pantalla para que el usuario cree un recordatorio (médico, banco, otro), con fecha, hora, notas y una casilla para solicitar ayuda asociada al recordatorio.”*

**🟦 4. Pantalla de Valoración del Mayor**

*“Diseña una pantalla para que el mayor valore la ayuda recibida con estrellas grandes y un campo de texto libre. Estilo accesible y amigable.”*

**🟦 5. Pantalla de Perfil del Voluntario**

*“Crea una pantalla donde el voluntario pueda configurar sus preferencias: tipo de ayuda, horarios, zona. Debe mostrar también cuántas veces ha ayudado y su valoración media.”*

**🟦 6. Pantalla de Selección de Voluntarios**

*“Diseña una pantalla para que el mayor pueda ver voluntarios disponibles en su zona, con nombre, foto, valoración y botón para seleccionar.”*

**🟦 7. Pantalla de Aceptar/Rechazar Solicitud (Voluntario)**

*“Genera una pantalla donde el voluntario vea los detalles de una solicitud de ayuda, con botón para aceptar o rechazar, y valoración del mayor que solicita.”*

**🟦 8. Pantalla de Valoración del Voluntario**

*“Diseña una pantalla para que el voluntario valore la experiencia de ayuda con estrellas y comentarios. Estilo accesible y cálido.”*

**🟦 9. Backend y Matching**

*“Genera el backend básico en Node.js con endpoints para usuarios, solicitudes, valoraciones y recordatorios. Incluye lógica de matching por zona y disponibilidad.”*

**🟦 10. IA Conversacional**

*“Integra una IA conversacional empática para personas mayores usando OpenAI. Debe responder con tono cálido, recordar citas y ofrecer compañía.”*

¡Claro que sí! De hecho, ya tenemos todo el código preparado para que el chat se conecte a OpenAI.

El único paso que falta es que tú añadas tu clave de API secreta para que el servidor pueda autenticarse. Como viste antes, el servidor se detuvo porque no encontró la clave.

Para solucionarlo, solo tienes que seguir estos pasos:

1. En la carpeta de tu proyecto (

VecinIA

), busca el archivo llamado

.env.example

.

1. **Crea una copia** de ese archivo en la misma carpeta y renombra la copia a

.env

.

1. Abre el nuevo archivo

.env

 con un editor de texto.

1. **Reemplaza el texto**

your\_api\_key\_here

 con tu clave de API real de OpenAI. El resultado debería ser algo como:

OPENAI\_API\_KEY=sk-xxxxxxxxxxxxxxxxxxxxxx

Una vez que hayas guardado tu clave en el archivo

.env

, avísame. En ese momento, reiniciaré el servidor y el chat estará completamente funcional y conectado a OpenAI.