

ChefBot

AI Assistant for Inclusive Dining - Documentación Técnica

RESUMEN EJECUTIVO

ChefBot es un asistente virtual basado en IA que proporciona recomendaciones gastronómicas personalizadas y seguras para personas con alergias, intolerancias y restricciones dietéticas. Diseñado con arquitectura RAG (Retrieval-Augmented Generation) para garantizar respuestas verificadas y evitar alucinaciones de IA.

Estado	MVP Funcional en Vercel
Fase actual	Optimización de prompts + seguridad
Próxima fase	Piloto en Zaragoza, España
Stack principal	OpenAI API + JavaScript + Vercel
Enfoque	Desarrollo ágil + IA Responsable

ARQUITECTURA DEL SISTEMA

La arquitectura de ChefBot está dividida en 4 capas principales que garantizan escalabilidad, seguridad y mantenibilidad:

Capa	Componentes	Tecnologías
Frontend	Landing Page + Chat Interface	HTML5, CSS3, JavaScript Vanilla
IA Engine	Procesamiento NLP + Recomendaciones	OpenAI API (GPT-4 / GPT-3.5)
Hosting	Deploy + CDN + SSL	Vercel + GitHub Pages
Datos	Menús + Matriz de Alérgenos	JSON estático (futuro: Firebase)

STACK TECNOLÓGICO

Categoría	Tecnología	Propósito
	HTML5 + CSS3	Estructura y estilos de la página
Frontend	JavaScript	Lógica del chat y llamadas API

	OpenAI API	Motor de procesamiento NLP
Backend/IA	GPT-4 / GPT-3.5-turbo	Generación de recomendaciones
	Vercel	Deploy automático + CDN global
Hosting	GitHub	Control de versiones + CI/CD
	JSON Local	Menús y matriz de alérgenos
Datos	Local Storage (opcional)	Caché de preferencias de usuario

DECISIONES ARQUITECTÓNICAS CLAVE

- ✓ **OpenAI API:** Permite prototipado rápido con IA de última generación sin necesidad de entrenar modelos propios.
- ✓ **JavaScript Vanilla:** Sin frameworks pesados - carga ultra-rápida y código simple para un MVP funcional.
- ✓ **Vercel:** Deploy en segundos con CI/CD automático desde GitHub. CDN global y SSL incluidos sin configuración.
- ✓ **JSON Local:** Suficiente para piloto con 1 restaurante. Permite control total y facilita debugging.
- ✓ **Prompt Engineering:** System prompts con contexto del menú y matriz de alérgenos para respuestas seguras.

FLUJO DE USUARIO

- 1 **Punto de entrada:** Usuario accede via URL de Vercel o escanea QR code en mesa.
- 2 **Input de alergias:** Completa formulario con restricciones dietéticas en interfaz JS.
- 3 **Construcción de prompt:** JS construye system prompt con contexto JSON del menú y alérgenos.
- 4 **Llamada OpenAI API:** fetch() envía prompt a GPT-4 con la información del usuario.
- 5 **Recomendaciones:** GPT procesa y devuelve platos seguros filtrados por alérgenos.
- 6 **Renderizado:** JavaScript muestra resultados en UI y permite seguir conversando.

IA RESPONSABLE & SEGURIDAD

ChefBot implementa múltiples capas de seguridad para proteger la salud de los comensales:

Mecanismo	Descripción
System Prompts	Contexto del menú y alérgenos inyectado en cada conversación
Matriz de Alérgenos	JSON con datos verificados por el restaurante como fuente de verdad
Prompt Engineering	Instrucciones explícitas: "NUNCA recomiendas si no estás 100% seguro"
Fallback seguro	Respuesta obligatoria: "Por favor, confirma con el personal" en dudas

Disclaimer visible	UI muestra mensaje: "Siempre verifica con el staff del restaurante"
---------------------------	---

MÉTRICAS DE ÉXITO

- **Precisión:** >95% en identificación correcta de alérgenos
- **Tiempo de respuesta:** <3 segundos por recomendación
- **Satisfacción:** NPS >70 entre usuarios con alergias
- **Escalado:** <10% de consultas requieren intervención humana

ROADMAP

Q4 2024: MVP funcional con OpenAI API + JavaScript + Vercel **Q1 2025:** Migrar API key a Vercel Serverless Functions (seguridad) **Q2 2025:** Piloto en restaurante real en Zaragoza **Q3 2025:** Implementar Firebase para menús dinámicos + analytics **Q4 2025:** Caché de respuestas comunes (reducir costos OpenAI)

Desarrollado por:	Sara Triana Merchan
LinkedIn:	linkedin.com/in/sara-triana-merchan
Email:	saratrianamerchan@gmail.com
Ubicación:	Zaragoza, España