

PLAN COMPLETO POR FASES - PROJECT IA-CODEV

VISIN GENERAL

Desarrollar una plataforma IA que funcione como un "equipo IT autnomo", asistiendo a un humano visionario en el desarrollo de software, generacin de cdigo, pruebas, documentacin, mejora continua y reaccin emocional mediante avatar. Debe analizarse a s misma desde el inicio, aprender del estilo y decisiones humanas, y funcionar localmente con posibilidad de escalar a la nube.

ESTRUCTURA MODULAR GENERAL

1. Frontend Web (Next.js)

- Interfaz de entrada de ideas y feedback.
- Consola de interaccin con IA.
- Componente de avatar IA.

2. Backend (FastAPI)

- Agentes IA (generator, tester, doc_writer, analyzer).
- Cliente OpenAI (GPT-4, TTS).
- Mdulo de memoria semntica (ChromaDB).

3. Avatar IA

- Modelo animado (VSeeFace, OBS).
- Voz IA (OpenAI TTS o ElevenLabs).

4. Memoria IA

- Base vectorial con feedback, decisiones y estilo de usuario.
- Autodiagnostico (analyzer.py).

5. Sistema de feedback

- Registra cada interacción y respuesta humana.
- Almacena preferencias, errores comunes, ajustes.

6. Observador (futuro)

- Analiza código, errores en tiempo real.
- Sugiere correcciones o soluciones.

PLAN POR FASES

Fase 1: MVP FUNCIONAL + AUTOANÁLISIS INICIAL

Objetivos:

- Frontend funcional (Next.js).
- Backend FastAPI conectado a GPT-4.
- IA genera código a partir de descripciones.
- Se implementa el agente analyzer.py.
- Feedback humano guardado.
- Memoria inicial con ChromaDB.
- Avatar IA con voz básica.

Componentes:

- main.py, client.py, generator.py, analyzer.py.

- index.tsx, Avatar.tsx, FeedbackButtons.tsx.

- .env, requirements.txt, package.json.

Herramientas: OpenAI, ChromaDB, VSeeFace/OBS, Git.

Entregables: Sistema que genera, recibe feedback y se autoanaliza.

Fase 2: AGENTES INTELIGENTES + MEMORIA ACTIVA

Objetivos:

- Agentes tester.py y doc_writer.py activos.
- Respuestas de IA se comparan con decisiones pasadas.
- IA adapta código a estilo preferido.

Componentes:

- Extensión de analyzer.py para categorizar errores.
- Almacenamiento vectorial por proyecto.

Herramientas: ChromaDB, LangChain opcional.

Entregables: IA ajusta respuestas basadas en experiencia pasada.

Fase 3: AVATAR COMPLETO + EXPRESIÓN EMOCIONAL

Objetivos:

- Avatar reacciona a eventos.

- Integra texto-a-voz IA.
- Responde de forma animada y contextual.

Componentes:

- Avatar.tsx, modelo en avatar/model.
- Integración OBS o WebGL + TTS.

Herramientas: VSeeFace, ElevenLabs o OpenAI TTS, OBS.

Entregables: Experiencia inmersiva con IA como compañera visible.

Fase 4: OBSERVADOR + AUTÓNOMA PROGRESIVA

Objetivos:

- Sistema detecta errores y eventos en tiempo real.
- IA propone correcciones sin ser solicitada.
- Modo "supervisión" configurable por el humano.

Componentes:

- Módulo de observación (VS Code plugin o consola).
- Feedback automático al sistema.

Entregables: IA actúa como copiloto y mentor adaptativo.

Fase 5: ESCALADO + ENTORNO AUTÓNOMO INTELIGENTE

Objetivos:

- Escalado a cloud con CI/CD.
- IA multiusuario y multiagente.
- Compartición de memoria entre proyectos.
- Autoevaluación cruzada entre tareas.

Componentes:

- Backend distribuido.
- Seguridad reforzada.
- Memoria compartida con identificador por usuario.

Herramientas: GitHub Actions, AWS/GCP, Pinecone opcional.

Entregables: Sistema IA completo, autoajutable, colaborativo.

REQUISITOS Y CONSIDERACIONES

- OpenAI API Key
- Node.js + npm
- Python 3.10+
- Git
- VSeeFace + OBS
- Seguridad local: entorno virtual, sin Docker (opcional para escalado)
- Modo offline inicial para desarrollo

GESTIÓN DE PROYECTO

- Repositorio Git estructurado desde Fase 1.
- Ramas por funcionalidad: main, dev, features/*
- Documentación continua (README, comentarios IA/humano).
- Feedback como primer dato para aprendizaje continuo.