

## Arrancamos oficialmente con el Día 1 — Semana 1 del plan para construir tu plataforma Enterprise AI Agent.

Hoy definimos la base del proyecto:

- qué vamos a construir,
- para quién,
- qué problema resuelve,
- por qué es valioso,
- cuál es el alcance técnico,
- y qué entregables concretos vas a producir.

Esto es clave porque es **lo primero que te pediría un jefe o un recruiter técnico**.

---

### **Día 1 — Semana 1: Definición del Caso de Negocio y Alcance**

#### **Objetivo del Día**

Dejar completamente definido **qué sistema de IA vas a construir y qué valor real aporta a una empresa moderna**.

Hoy vas a salir con una idea concreta, clara, vendible y técnicamente sólida.

---

#### **Definimos el Caso de Uso Real (Use Case)**

Vamos a trabajar sobre el caso que ya seleccionamos para maximizar tu empleabilidad:

#### **Enterprise AI Agent Platform**

Un sistema compuesto por:

1. **Un “Agente Analista de Documentos”** (RAG, embeddings, resúmenes inteligentes)
2. **Un “Agente Operativo”** (tareas autónomas, extracción de datos, clasificación, automatizaciones)
3. **Un “Agente Business Insights”** (análisis avanzado, reportes ejecutivos, insight discovery)
4. **Una API que expone a estos agentes**
5. **Una interfaz web simple para demostrarlo**

Este conjunto simula EXACTAMENTE lo que empresas están adoptando hoy: la “AI workforce”, agentes autónomos que trabajan junto a humanos.

---

## 2 ¿Qué problema real resuelve?

Las empresas modernas tienen problemas como:

- Pérdida de tiempo buscando información.
- Procesos lentos por revisar PDFs, contratos, informes, mails.
- Falta de insights de negocio porque la info está dispersa.
- Costos elevados por tareas repetitivas.
- Demoras en la toma de decisiones.

Tu plataforma resuelve eso con:

- ✓ automatización,
- ✓ análisis,
- ✓ extracción,
- ✓ y generación de conocimiento usable.

---

## 3 ¿Qué valor aporta a una empresa?

Este es el punto que más importa en entrevistas:

### 💡 Valor empresarial directo

#### Ahorro de tiempo

Un analista humano tarda 20–60 min leyendo un documento.  
Tu agente lo hace en 2 segundos.

#### Reducción de costos

Automatizar tareas repetitivas baja horas operativas hasta un 70%.

#### Mejor toma de decisiones

El agente integra información dispersa y genera insights listos para usar.

#### Menos errores

No se olvidan datos, no se ocultan riesgos, no se interpreta mal.

#### Escala infinita

Puede analizar cientos o miles de documentos sin aumentar personal.

---

## 4 ¿Qué habilidades técnicas vas a dominar?

Este proyecto te da un stack técnico MUY atractivo para empresas:

### IA / Machine Learning

- RAG (Retrieval-Augmented Generation)
- Embeddings vectoriales
- Pipelines de documentos
- Evaluación de respuestas
- Agentes autónomos (Agentic AI)
- OpenAI, Groq o similar

### Backend

- APIs REST en Node o Java (vos elegís)
- Arquitectura hexagonal / limpia
- Controladores, servicios, repositorios

### Infraestructura

- Docker
- Servicios escalables
- Separación de agentes

### Integraciones

- Base vectorial (Chroma / Pinecone / Qdrant)
- Llamadas a modelos de lenguaje
- Webhooks / colas si querés escalar

### Buenas prácticas

- Diseño de prompts empresariales
- Logs, trazas, errores
- Pruebas unitarias

Te posiciona como **Backend + AI Engineer**, un perfil hoy muy buscado.

---

## 5 Definimos el Alcance del Proyecto (Scope)

Para poder decir “esto es un producto”, hoy vamos a cerrar:

## LO QUE SÍ incluye

- Ingesta de documentos (PDF, TXT)
- Limpieza, chunking, embeddings
- Motor RAG
- 3 agentes autónomos:
  - Analista
  - Operativo
  - Insights
- API REST completa
- UI simple (React o HTML) para demo
- Logs de agentes
- Manual técnico y de negocio

## LO QUE NO incluye

- Modelos entrenados desde cero
- Integraciones corporativas reales
- Seguridad avanzada
- UI compleja

Esto mantiene el proyecto manejable y presentable.

---

### **Entregables concretos del Día 1**

Hoy tenemos que producir:

#### **1. Descripción del Proyecto (versión ejecutiva)**

→ Ya empezamos a escribirla (puedo generarte el PDF si querés).

#### **2. Mapa conceptual del sistema (Arquitectura general)**

→ Lo hacemos hoy.

#### **3. Lista “oficial” de agentes y sus responsabilidades**

→ Lo dejamos listo hoy.

#### **4. Definición de tecnologías**

→ También hoy.

## 5. Definición del flujo de documentos

→ Hoy lo diseñamos.

## Definición completa del alcance + arquitectura general + agentes + tecnologías

Vamos a definir el primer paso a seguir. Todo lo que definamos hoy se usa mañana para crear el repositorio, estructura del proyecto, carpetas y primeros endpoints.

---

### **Arquitectura General del Proyecto (versión final)**

Tu plataforma tendrá **4 capas**:

#### **1. Capa de Ingesta y Procesamiento de Documentos**

- Subida de PDFs / textos.
- Limpieza del contenido.
- Chunking en bloques manejables.
- Generación de embeddings.
- Almacenamiento en una base vectorial (Chroma DB local).

#### **2. Motor RAG**

- Recuperación de chunks relevantes.
- Construcción del contexto.
- Paso al modelo LLM.
- Respuesta enriquecida.

#### **3. Agentes Autónomos**

Tendrás **3 agentes**:

##### **Agente 1: *Document Analyst Agent***

- Resume documentos.
- Extrae puntos clave.
- Identifica riesgos, oportunidades, métricas.

### ★ **Agente 2: *Operational Automation Agent***

- Clasifica documentos por tipo (factura, contrato, análisis, informe).
- Extrae campos estructurados (fecha, cliente, monto).
- Genera acciones automáticas (ej: “crear reporte”).

### ★ **Agente 3: *Business Insights Agent***

- Busca patrones.
- Crea dashboards narrados.
- Detecta tendencias o anomalías.
- Genera resúmenes ejecutivos.

## **4. API REST + UI**

- API en Node para usar los agentes.
  - UI muy simple (HTML + JS o React) para demostrarlo.
  - Endpoint /agents/... para interactuar.
- 

## **2 Tecnologías (versión definitiva)**

### **Backend**

- Node.js 18+
- Express.js
- OpenAI API (para modelos)
- LangChain JS (para agentes + RAG si querés)
- ChromaDB (vector store local y GRATIS)
- pdf-parse o pdfjs-dist (para leer PDF)
- dotenv
- axios/fetch

### **Infraestructura**

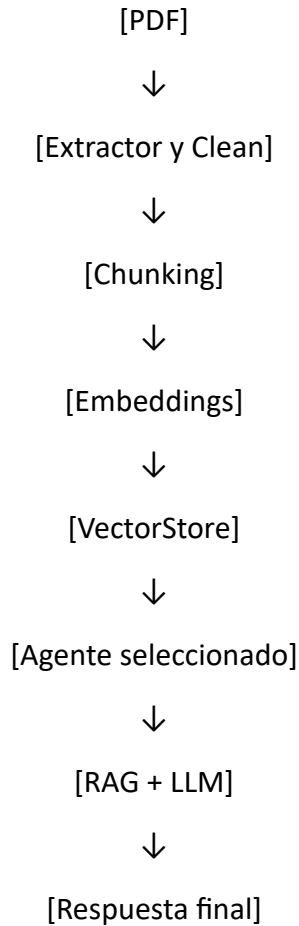
- Docker (lo hacemos en semana 3)
- Logs en JSON

### **Frontend**

- HTML + JS simple (por rapidez)  
o
  - React (si querés dejarlo PRO, semana 4)
- 

### 3 Flujo de Operación del Sistema

Cuando un usuario sube un documento, pasa esto:



## Enterprise AI Agent Platform (Node.js + Express) – Alcance, Objetivo y Arquitectura

---

### 1. Objetivo del Proyecto

Construir una **plataforma modular de IA empresarial basada en agentes y RAG**, capaz de:

- procesar documentos internos,
- recuperar información semántica,
- automatizar análisis operativos,
- generar resúmenes ejecutivos,
- detectar patrones de negocio,
- entregar insights listos para decidir,
- integrarse a través de una API Express moderna.

Este proyecto demuestra **skills 100% alineadas** con roles de LLM Engineer, GenAI Developer y AI Solutions Architect.

---

## 2. Valor para una empresa real

Tu plataforma agrega valor directo en cuatro áreas:

### 1) Reducción de costos operativos

Automatiza tareas como:

- lectura de PDFs,
- clasificación,
- extracción de datos,
- resúmenes.

Esto reduce horas hombre **de analistas** y evita errores humanos.

### 2) Aceleración del análisis interno

Un empleado puede preguntar:

“¿Qué dice el contrato sobre renovación automática?”

Y el sistema responde **en segundos**.

### 3) Capitalización del conocimiento interno

Todas las políticas, manuales, informes y contratos quedan disponibles vía:

- búsqueda semántica,
- agentes inteligentes,
- RAG preciso.

La empresa deja de perder tiempo “buscando información”.

### 4) Insights de valor



El **Business Insights Agent** detecta:

- anomalías,
- tendencias,
- métricas importantes,
- riesgos y oportunidades.

Esto convierte tus datos en **decisiones accionables**.

---

### 3. Arquitectura general confirmada

#### Capa 1 – Ingesta y Preprocesamiento

- Lectura de PDF (pdf-parse)
- Limpieza de texto
- Chunking semántico
- Normalización de contenido

#### Capa 2 – Embeddings + Vector Store

- Generación de embeddings
- Almacenamiento en ChromaDB
- Funciones de recuperación semántica (similarity search)
- Evaluación de recall/precision

#### Capa 3 – Agentes de IA

##### Document Analyst Agent

- Resúmenes
- Extracción de puntos clave
- Comparación entre documentos

##### Operational Automation Agent

- Clasificación automática
- Extracción de entidades (NLP)
- Generación de reportes estructurados

##### Business Insights Agent

- Análisis estadístico simple

- Detección de anomalías
- Insights narrados por IA

## Capa 4 – API en Express

Endpoints:

- /health
- /documents/upload
- /documents/search
- /agents/analyze
- /agents/automate
- /agents/insights

Más el “**Orchestrator**” que decide qué agente usar según la intención.

---

## 4. Tecnologías confirmadas

### Backend

- Node.js 18+
- Express.js
- LangChain JS (agents + RAG)
- OpenAI API
- ChromaDB local
- pdf-parse
- dotenv
- nodemon

### Infra

- Docker (semana 3)
- Logs estructurados JSON
- GitHub Actions (semana 7)

### UI

- HTML simple en las primeras semanas
- React opcional más adelante

---

## 5. Flujo general del sistema

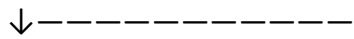
Usuario



[API Express]



Agente Orquestador



DocAgent   AutoAgent   InsightsAgent



Pipeline RAG



Vector Store



LLM



Respuesta