

## **PROYECTO 3: CRM - SISTEMA DE TOMA DE DECISIONES BASADO EN DOCUMENTOS**

### **Descripción**

CRM inteligente que procesa documentos comerciales (contratos, propuestas, correos) usando NLP y RAG para extraer insights automáticamente y sugerir acciones comerciales basadas en el análisis de contenido.

### **Requerimientos**

- **Desarrollo:** Aplicación web con capacidades de procesamiento de documentos
- **Funcionalidades básicas:**
  - Upload y procesamiento de documentos (PDF, Word, emails)
  - Extracción de entidades comerciales con NLP
  - Sistema de recomendaciones basado en RAG
  - Análisis de sentimiento en comunicaciones
  - Generación automática de resúmenes ejecutivos
  - Alertas inteligentes basadas en contenido
- **Integración:** APIs de procesamiento de documentos, sistemas de email, storage clouds

### **Ayudas para la obtención de datos**

- **API:**
  - OpenAI GPT-4 API para procesamiento de lenguaje
  - Azure Cognitive Services
  - Google Cloud Document AI
- **Datos de prueba:**
  - Colección de contratos comerciales anonimizados
  - Base de conocimiento empresarial
  - Correspondencia comercial histórica

### **Comprensión del problema**

- **Descripción:** Los equipos comerciales pierden información valiosa en documentos no estructurados que podrían impulsar mejores decisiones
- **Aplicaciones existentes:** Salesforce Einstein, Microsoft Viva Sales, Gong.io
- **Diferenciador:** RAG especializado en documentos comerciales con análisis contextual profundo

### Desarrollo de la aplicación

- **Tecnologías:** .NET 8, Next.js 14, Vector Database (Pinecone), LangChain, OpenAI API
- **Backend:**
  - .NET Core con arquitectura hexagonal
  - LangChain.NET para RAG implementation
  - Semantic Kernel para IA workflows
  - Azure Blob Storage para documentos
- **Frontend:**
  - Next.js con drag-and-drop de documentos
  - React PDF Viewer para visualización
  - Monaco Editor para edición de templates
  - Recharts para analytics de documentos
- **Referentes:** Salesforce Einstein Analytics, Microsoft Viva Sales, Chorus.ai

### Entregables

- Sistema RAG para análisis de documentos comerciales
  - Dashboard de insights extraídos automáticamente
  - Motor de recomendaciones basado en contenido
  - API de procesamiento de documentos
  - Manual de configuración de modelos IA
-

## CARACTERÍSTICAS COMUNES DE TODOS LOS PROYECTOS

### Arquitectura Base

- **Backend:** .NET 8 con Clean Architecture/Hexagonal
- **Frontend:** React 18+ con Next.js 14 o Blazor Server
- **Base de datos:** SQLServer como principal, especializadas según necesidad
- **Cache:** Redis para performance
- **Contenedores:** Docker con orquestación

### Seguridad y Autenticación

- **Autenticación:** JWT + OAuth 2.0 / OpenID Connect
- **Validación email:** SendGrid o similar para verificación
- **Autorización:** Role-based + Claims-based
- **Cifrado:** TLS 1.3, cifrado de datos en reposo

### Internacionalización

- **Idiomas:** Español (primario), Inglés, Portugués
- **Framework:** i18next para React, .NET Localization para backend
- **Formato:** Fechas, números y monedas localizados
- **Timezone:** Soporte múltiples zonas horarias

### Inteligencia Artificial

- **ML Framework:** ML.NET para modelos nativos
- **NLP/RAG:** Integration con OpenAI/Azure OpenAI
- **Vector Database:** Pinecone, Weaviate o similar
- **Analytics:** Modelos predictivos específicos por dominio

### DevOps y Deployment

- **CI/CD:** Azure DevOps o GitHub Actions
- **Monitoreo:** Application Insights + Prometheus
- **Logging:** Serilog con Elasticsearch

- **Cloud:** Azure como plataforma principal

### **Consideraciones Especiales**

- **Performance:** Implementación de CQRS donde sea necesario
- **Escalabilidad:** Arquitectura preparada para microservicios
- **Resilience:** Circuit breakers, retry policies, timeouts
- **Testing:** Cobertura mínima 80% con testing automatizado