**Plan Completo de Desarrollo - Sistema de Gestión Logística con ML**

**RESUMEN DEL PROYECTO**

**Título:** Diseño e implementación de un sistema de gestión logística con algoritmos de Machine Learning para optimizar el abastecimiento de productos en un mini market del distrito de Mariano Melgar, Arequipa -- 2025

**Tecnologías a utilizar:**

* Backend: Python + Django
* Frontend: React
* Base de datos: PostgreSQL
* ML: scikit-learn, pandas, numpy
* Servidor web: Nginx (opcional para producción)

**ESTRUCTURA COMPLETA DEL PROYECTO**

minimarket\_ml\_system/

├── backend/

│ ├── config/

│ │ ├── \_\_init\_\_.py

│ │ ├── settings.py

│ │ ├── urls.py

│ │ ├── wsgi.py

│ │ └── asgi.py

│ ├── apps/

│ │ ├── \_\_init\_\_.py

│ │ ├── products/

│ │ │ ├── \_\_init\_\_.py

│ │ │ ├── models.py

│ │ │ ├── views.py

│ │ │ ├── serializers.py

│ │ │ ├── urls.py

│ │ │ └── admin.py

│ │ ├── inventory/

│ │ │ ├── \_\_init\_\_.py

│ │ │ ├── models.py

│ │ │ ├── views.py

│ │ │ ├── serializers.py

│ │ │ ├── urls.py

│ │ │ └── admin.py

│ │ ├── sales/

│ │ │ ├── \_\_init\_\_.py

│ │ │ ├── models.py

│ │ │ ├── views.py

│ │ │ ├── serializers.py

│ │ │ ├── urls.py

│ │ │ └── admin.py

│ │ ├── ml\_models/

│ │ │ ├── \_\_init\_\_.py

│ │ │ ├── models.py

│ │ │ ├── views.py

│ │ │ ├── serializers.py

│ │ │ ├── urls.py

│ │ │ ├── predictor.py

│ │ │ ├── trainer.py

│ │ │ └── data\_processor.py

│ │ └── analytics/

│ │ ├── \_\_init\_\_.py

│ │ ├── models.py

│ │ ├── views.py

│ │ ├── serializers.py

│ │ └── urls.py

│ ├── static/

│ ├── media/

│ ├── data/

│ │ ├── raw/

│ │ ├── processed/

│ │ └── models/

│ ├── requirements.txt

│ ├── manage.py

│ └── db.sqlite3

├── frontend/

│ ├── public/

│ │ ├── index.html

│ │ └── favicon.ico

│ ├── src/

│ │ ├── components/

│ │ │ ├── Dashboard/

│ │ │ ├── Products/

│ │ │ ├── Inventory/

│ │ │ ├── Sales/

│ │ │ ├── Predictions/

│ │ │ ├── Analytics/

│ │ │ └── Common/

│ │ ├── pages/

│ │ ├── services/

│ │ ├── utils/

│ │ ├── styles/

│ │ ├── App.js

│ │ ├── index.js

│ │ └── App.css

│ ├── package.json

│ └── package-lock.json

├── docs/

│ ├── README.md

│ ├── INSTALLATION.md

│ └── API\_DOCUMENTATION.md

└── scripts/

├── setup.bat

├── run\_backend.bat

├── run\_frontend.bat

└── data\_migration.py

**ETAPAS COMPLETAS DEL DESARROLLO**

**ETAPA 1: CONFIGURACIÓN INICIAL Y ENTORNO**

**Duración:** 1 semana **Archivos a crear:** Estructura base, configuración de entorno

**Tareas:**

1. Instalación de herramientas (Python, Node.js, PostgreSQL)
2. Creación de la estructura de directorios
3. Configuración del entorno virtual
4. Instalación de dependencias
5. Configuración inicial de Django
6. Configuración inicial de React
7. Configuración de la base de datos PostgreSQL

**Comandos principales:**

bash

*# Crear directorio principal*

mkdir minimarket\_ml\_system

cd minimarket\_ml\_system

*# Crear entorno virtual*

python -m venv venv

venv\Scripts\activate

*# Crear proyecto Django*

django-admin startproject config backend

cd backend

python manage.py startapp products

python manage.py startapp inventory

python manage.py startapp sales

python manage.py startapp ml\_models

python manage.py startapp analytics

*# Crear proyecto React*

cd ..

npx create-react-app frontend

**ETAPA 2: ANÁLISIS Y RECOPILACIÓN DE DATOS**

**Duración:** 2 semanas **Archivos principales:** data\_collection.py, data\_analysis.py

**Tareas:**

1. Creación de modelos de datos para productos, inventario y ventas
2. Desarrollo de scripts para generar datos simulados históricos
3. Análisis exploratorio de datos (EDA)
4. Identificación de patrones de demanda
5. Limpieza y preparación de datos
6. Documentación del análisis actual del mini market

**Entregables:**

* Base de datos con 2 años de datos históricos simulados
* Reportes de análisis exploratorio
* Documentación de patrones identificados

**ETAPA 3: DESARROLLO DE MODELOS DE MACHINE LEARNING**

**Duración:** 3 semanas **Archivos principales:** trainer.py, predictor.py, data\_processor.py

**Tareas:**

1. Implementación de preprocesamiento de datos
2. Desarrollo de modelos de predicción:
   * Regresión Lineal
   * Random Forest
   * Árboles de Decisión
   * Prophet (series temporales)
3. Entrenamiento y validación de modelos
4. Evaluación de métricas (MAE, RMSE)
5. Selección del mejor modelo
6. Creación de API para predicciones

**Entregables:**

* Modelos entrenados y guardados
* API de predicción funcional
* Reportes de evaluación de modelos

**ETAPA 4: DESARROLLO DEL BACKEND (API REST)**

**Duración:** 2 semanas **Archivos principales:** models.py, views.py, serializers.py, urls.py

**Tareas:**

1. Definición completa de modelos de Django
2. Creación de APIs REST para:
   * Gestión de productos
   * Control de inventario
   * Registro de ventas
   * Predicciones ML
   * Analytics y reportes
3. Implementación de autenticación
4. Validaciones y manejo de errores
5. Documentación de APIs

**Entregables:**

* API REST completa y funcional
* Documentación de endpoints
* Sistema de autenticación implementado

**ETAPA 5: DESARROLLO DEL FRONTEND (INTERFAZ WEB)**

**Duración:** 3 semanas **Archivos principales:** Componentes React, páginas, servicios

**Tareas:**

1. Diseño de la interfaz de usuario
2. Desarrollo de componentes React:
   * Dashboard principal
   * Gestión de productos
   * Control de inventario
   * Visualización de predicciones
   * Reportes y analytics
3. Integración con APIs del backend
4. Implementación de gráficos y visualizaciones
5. Responsive design
6. Manejo de estados y navegación

**Entregables:**

* Aplicación web completa y funcional
* Interfaz responsive
* Integración completa con backend

**ETAPA 6: INTEGRACIÓN Y PRUEBAS DEL SISTEMA**

**Duración:** 1 semana **Archivos principales:** test\_\*.py, integration\_tests.py

**Tareas:**

1. Integración completa frontend-backend
2. Pruebas unitarias
3. Pruebas de integración
4. Pruebas de la funcionalidad ML
5. Optimización de rendimiento
6. Corrección de bugs

**Entregables:**

* Sistema completamente integrado
* Suite de pruebas
* Reportes de testing

**ETAPA 7: IMPLEMENTACIÓN PILOTO Y VALIDACIÓN**

**Duración:** 4 semanas **Archivos principales:** deployment scripts, monitoring tools

**Tareas:**

1. Preparación del entorno de producción
2. Deployment del sistema
3. Capacitación del personal del mini market
4. Recolección de datos reales de uso
5. Monitoreo del rendimiento
6. Ajustes basados en feedback

**Entregables:**

* Sistema en producción
* Personal capacitado
* Datos de rendimiento real

**ETAPA 8: EVALUACIÓN Y DOCUMENTACIÓN FINAL**

**Duración:** 2 semanas **Archivos principales:** Documentación, reportes finales

**Tareas:**

1. Análisis comparativo antes/después
2. Cálculo de métricas de mejora
3. Documentación técnica completa
4. Manual de usuario
5. Documentación de mantenimiento
6. Preparación de presentación final

**Entregables:**

* Documentación completa del sistema
* Reportes de resultados
* Manual de usuario
* Código fuente documentado

**CRONOGRAMA DETALLADO**

**MÉTRICAS DE ÉXITO**

**Técnicas:**

* Precisión de predicción > 90%
* Reducción de quiebres de stock > 15%
* Reducción de sobrestock > 18%
* Tiempo de respuesta del sistema < 2 segundos

**Operativas:**

* Disponibilidad de productos mejorada
* Rotación de inventario optimizada
* Satisfacción del personal del mini market
* ROI positivo del sistema

**RECURSOS NECESARIOS**

**Software:**

* Python 3.9+
* Node.js 16+
* PostgreSQL 13+
* Visual Studio Code
* Git para control de versiones

**Hardware:**

* PC con mínimo 8GB RAM
* Espacio en disco: 10GB
* Conexión a internet estable

**Datos:**

* Registros históricos de ventas (2 años)
* Catálogo de productos
* Datos de inventario
* Información de proveedores

**SIGUIENTE PASO**

Para comenzar, ejecutaremos la **ETAPA 1: CONFIGURACIÓN INICIAL Y ENTORNO**.

¿Estás listo para comenzar con la Etapa 1? Te proporcionaré todos los comandos y códigos necesarios para la configuración inicial del proyecto.

**NOTA IMPORTANTE PARA CHATS FUTUROS**

Cuando abras un nuevo chat, proporciona:

1. Este documento completo
2. La etapa en la que te encuentras
3. Cualquier problema o modificación que hayas encontrado

Cada etapa está diseñada para ser autocontenida pero conectada con las anteriores, permitiendo continuidad entre diferentes sesiones de chat.