**Consulta: quiero hacer prototipos, usando IA, para mostrar posibles soluciones de procesos, estos prototipos deben correr en maquinas virtuales y mostrar resultados en HTML**

**1. Procesamiento e IA**

* Lenguaje: Python (ideal para IA)
* Librerías: pandas, scikit-learn, transformers, openai, xgboost, etc.
* Modelos: puede usar modelos preentrenados o entrenar los suyos en procesos livianos.

**2. Entorno de Ejecución**

* **Máquinas virtuales:**
  + AWS EC2 (Linux o Windows)
  + Google Cloud Compute Engine
  + Azure VMs
* Automatice el setup con scripts bash o Docker si lo prefiere portatil y reproducible.

**3. Visualización y Resultados en HTML**

* **Frameworks recomendados:**
  + [Streamlit](https://streamlit.io/) → súper rápido y visual
  + [Gradio](https://gradio.app/) → genial para prototipos con IA
  + Dash (más técnico, con enfoque empresarial)
  + Flask o FastAPI si desea algo más personalizado en HTML puro

**4. Ejemplo de flujo típico**

1. VM recibe o detecta archivo (por cron job o carga manual)
2. Código de Python procesa los datos con IA
3. Resultado se guarda en HTML dinámico (Streamlit o Gradio lo hacen automáticamente)
4. Usuario accede vía navegador al dashboard de resultados

**un prototipo básico**

Generar uno con:

* Carga de archivo .xlsx
* Análisis de datos con IA
* Visualización en HTML con Streamlit
* Listo para ejecutar en una VM (con instrucciones de despliegue)

Antes de prototipos , el demo es fabuloso: streamlit hello

http://localhost:8501/dataframe\_demo

**# prototipo\_ia\_streamlit.py /// http://localhost:8501/**

import streamlit as st

import pandas as pd

import plotly.express as px

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

# Título del dashboard

st.title("Prototipo IA - Análisis de Datos y Visualización")

# Subir archivo Excel

archivo = st.file\_uploader("Suba un archivo .xlsx con datos", type=["xlsx"])

if archivo:

hojas = pd.ExcelFile(archivo).sheet\_names

hoja = st.selectbox("Seleccione la hoja a analizar", hojas)

df = pd.read\_excel(archivo, sheet\_name=hoja)

st.subheader("Vista previa de los datos")

st.dataframe(df.head())

columnas\_numericas = df.select\_dtypes(include='number').columns.tolist()

# Simple IA: regresión lineal entre dos variables

st.subheader("Modelo de Regresión Lineal")

x\_col = st.selectbox("Variable Independiente (X)", columnas\_numericas)

y\_col = st.selectbox("Variable Dependiente (Y)", columnas\_numericas, index=1 if len(columnas\_numericas) > 1 else 0)

if x\_col != y\_col:

X = df[[x\_col]]

y = df[y\_col]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

modelo = LinearRegression()

modelo.fit(X\_train, y\_train)

y\_pred = modelo.predict(X\_test)

resultado\_df = pd.DataFrame({"X\_test": X\_test[x\_col], "y\_real": y\_test, "y\_pred": y\_pred})

st.success(f"Coeficiente: {modelo.coef\_[0]:.4f}, Intercepto: {modelo.intercept\_:.4f}")

fig = px.scatter(resultado\_df, x="X\_test", y="y\_real", title="Predicciones vs Valores Reales")

fig.add\_scatter(x=resultado\_df["X\_test"], y=resultado\_df["y\_pred"], mode='lines', name='Predicciones')

st.plotly\_chart(fig)

else:

st.warning("Las variables X y Y deben ser diferentes.")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Ya tiene un prototipo base listo en Python con Streamlit, que:**

1. Recibe un archivo .xlsx
2. Permite elegir una hoja y visualizar los datos
3. Aplica un modelo de **regresión lineal simple**
4. Muestra resultados en un gráfico interactivo en HTML (usando Plotly) :

**🖥️ Para ejecutarlo localmente:**

1. Instale las dependencias:

pip install streamlit pandas plotly scikit-learn openpyxl

1. Ejecute el prototipo:

streamlit run prototipo\_ia\_streamlit.py

 Se abrirá en su navegador en http://localhost:8501

1. Ejecute el prototipo: streamlit run prototipo\_ia\_streamlit.py
2. Se abrirá en su navegador en <http://localhost:8501>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**🖥️ o un entorno virtual preconfigurado :**

prepare un requirements.txt para este proyecto

**requirements.txt**

Cree este archivo en su carpeta del proyecto (src/prototipos\_IA/requirements.txt) con el siguiente contenido:

streamlit

pandas

plotly

scikit-learn

openpyxl

**🛠️ Crear y usar un entorno virtual en Windows**

1. **Abra la terminal en la raíz de su proyecto**:

cd c:\Users\PC\Documents\mi\_proyecto\_python\src\prototipos\_IA

1. **Cree el entorno virtual**:

python -m venv venv

1. **Actívelo**:

.\venv\Scripts\activate

1. **Instale las dependencias**:

pip install -r requirements.txt

**✅ Verificación final**

Ahora ejecute su prototipo con:

streamlit run prototipo\_ia\_streamlit.py

**Notas acerca de steamlit:** para ejecutar una prueba con salida html

Warning: to view this Streamlit app on a browser, run it with the following command:

streamlit run c:/Users/PC/Documents/mi\_proyecto\_python/src/prototipos\_IA/prb.py [ARGUMENTS]