**Paso 1: Configuración del Entorno**

1. **Instalación de Anaconda y Spyder**:
   * Descarga e instala Anaconda para gestionar tus entornos de Python.
   * Abre Anaconda Navigator y lanza Spyder.
2. **Instalación de MongoDB**:
   * Descarga e instala MongoDB.
   * Inicia el servidor de MongoDB.
3. **Instalación de Bibliotecas Necesarias**:
   * Abre una terminal en Anaconda y ejecuta:
   * conda install pandas numpy scikit-learn pymongo

**Paso 2: Cargar Datos en MongoDB**

1. **Preparar los Datos**:
   * Asegúrate de tener los archivos CSV con los datos de la USA Swimming Association y del Tecnológico Nacional de México.
2. **Cargar los Datos en MongoDB**:
   * Utiliza el siguiente script en Spyder para cargar los datos en MongoDB:

**Python**

import pymongo

import pandas as pd

# Conectar a MongoDB

client = pymongo.MongoClient("mongodb://localhost:27017/")

db = client["natacion"]

collection = db["datos\_nadadores"]

# Cargar datos desde un archivo CSV

data = pd.read\_csv('ruta\_al\_archivo/datos\_nadadores.csv')

data\_dict = data.to\_dict("records")

# Insertar datos en MongoDB

collection.insert\_many(data\_dict)

Código generado por IA. Revisar y usar cuidadosamente. [Más información sobre preguntas frecuentes](https://www.bing.com/new#faq).

**Paso 3: Almacenamiento y Gestión de Datos**

1. **Configuración de Hadoop y HDFS**:
   * Descarga e instala Hadoop.
   * Configura HDFS para almacenar los datos de manera distribuida.
2. **ETL con Apache NiFi**:
   * Descarga e instala Apache NiFi.
   * Configura flujos de datos para extraer datos de MongoDB, transformarlos y cargarlos en HDFS.

**Paso 4: Procesamiento de Datos**

1. **Configuración de Apache Spark**:
   * Descarga e instala Apache Spark.
   * Configura Spark para trabajar con HDFS.
2. **Procesamiento con MapReduce**:
   * Implementa trabajos de MapReduce para procesar los datos en HDFS. Aquí tienes un ejemplo básico en Python:

**Python**

from pyspark import SparkContext, SparkConf

conf = SparkConf().setAppName("NatacionMapReduce")

sc = SparkContext(conf=conf)

# Leer datos desde HDFS

datos = sc.textFile("hdfs://ruta\_al\_archivo/datos\_nadadores.csv")

# Procesamiento MapReduce

def map\_func(line):

fields = line.split(',')

return (fields[0], float(fields[1]))

def reduce\_func(a, b):

return a + b

resultados = datos.map(map\_func).reduceByKey(reduce\_func)

resultados.saveAsTextFile("hdfs://ruta\_al\_archivo/resultados")

Código generado por IA. Revisar y usar cuidadosamente. [Más información sobre preguntas frecuentes](https://www.bing.com/new#faq).

**Paso 5: Análisis de Datos**

1. **Análisis Exploratorio de Datos (EDA) en Spyder**:
   * Utiliza Spyder para realizar EDA. Aquí tienes un ejemplo:

**Python**

import pandas as pd

import matplotlib.pyplot as plt

# Cargar datos desde MongoDB

client = pymongo.MongoClient("mongodb://localhost:27017/")

db = client["natacion"]

collection = db["datos\_nadadores"]

data = pd.DataFrame(list(collection.find()))

# Análisis exploratorio

print(data.describe())

data.hist(bins=50, figsize=(20,15))

plt.show()

Código generado por IA. Revisar y usar cuidadosamente. [Más información sobre preguntas frecuentes](https://www.bing.com/new#faq).

1. **Desarrollo de Modelos Predictivos**:
   * Utiliza Scikit-learn para desarrollar modelos predictivos. Aquí tienes un ejemplo:

**Python**

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

# Separar variables independientes y dependientes

X = data[['estatura', 'peso', 'IMC', 'flexibilidad']]

y = data['tiempo\_mariposa']

# Dividir los datos en conjuntos de entrenamiento y prueba

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

# Entrenar el modelo

modelo = LinearRegression()

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

# Hacer predicciones

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

# Evaluar el modelo

mse = mean\_squared\_error(y\_test, y\_pred)

print(f'Error cuadrático medio: {mse}')

Código generado por IA. Revisar y usar cuidadosamente. [Más información sobre preguntas frecuentes](https://www.bing.com/new#faq).

**Paso 6: Desarrollo del Modelo Predictivo**

1. **Entrenamiento del Modelo**:
   * Entrena tu modelo con los datos recolectados utilizando técnicas de validación cruzada para evaluar su rendimiento.
2. **Optimización del Modelo**:
   * Ajusta los hiperparámetros del modelo para mejorar su precisión y capacidad predictiva.

**Paso 7: Visualización de Datos**

1. **Creación de Dashboards**:
   * Utiliza Tableau o Power BI para crear dashboards interactivos que permitan visualizar los datos y las predicciones.

**Paso 8: Implementación y Monitoreo**

1. **Despliegue del Modelo**:
   * Implementa el modelo predictivo en un entorno de producción utilizando servicios en la nube como AWS o Google Cloud Platform.
2. **Monitoreo y Mantenimiento**:
   * Monitorea el rendimiento del modelo y actualiza los datos y el modelo regularmente para mantener su precisión.

**Ejemplo de Interfaz con Dash**

Para crear una interfaz similar a este chat, puedes utilizar Dash de Plotly:

**Python**

import dash

import dash\_core\_components as dcc

import dash\_html\_components as html

from dash.dependencies import Input, Output

import pandas as pd

import plotly.express as px

# Cargar datos desde MongoDB

client = pymongo.MongoClient("mongodb://localhost:27017/")

db = client["natacion"]

collection = db["datos\_nadadores"]

data = pd.DataFrame(list(collection.find()))

# Crear la aplicación Dash

app = dash.Dash(\_\_name\_\_)

app.layout = html.Div([

dcc.Graph(id='grafico'),

dcc.Dropdown(

id='dropdown',

options=[{'label': col, 'value': col} for col in data.columns],

value='estatura'

)

])

@app.callback(

Output('grafico', 'figure'),

[Input('dropdown', 'value')]

)

def update\_graph(selected\_column):

fig = px.histogram(data, x=selected\_column)

return fig

if \_\_name\_\_ == '\_\_main\_\_':

app.run\_server(debug=True)