PRÁCTICA 2 – BD NOSQL

IRENE GONZÁLEZ GONZÁLEZ
irenegonzalez@opendeusto.es

2024–2025

ÍNDICE

1. INTRODUCCIÓN

Esta práctica tiene como objetivo demostrar cómo modelar, almacenar y consultar datos relacionados con jugadoras de fútbol femeninas en una base de datos NoSQL a elección propia explicando porque se ha considerado como mejor opción.

2. DESCRIPCIÓN DEL DATASET

El dataset seleccionado proviene de la plataforma Kaggle y contiene información detallada sobre jugadoras de fútbol, incluyendo atributos físicos, habilidades técnicas y datos relacionados con su equipo y la liga en la que juegan.

Algunas de las columnas más relevantes son:

- **Name**: Nombre de la jugadora.
- **Rank**: Clasificación general.
- **Age**: Edad de la jugadora.
- **Position**: Posición en el campo.
- **Team**: Equipo al que pertenece.
- **League** y **Nation**: Liga y nacionalidad.
- **Atributos técnicos**: Velocidad, disparo, pases, entre otros.
- **Start_year**: Año de inicio en el fútbol profesional.
- **Nation**: Nacionalidad de la jugadora

Este dataset se podrá encontrar en el repositorio de GitHub.

3. ELECCIÓN DE LA BASE DE DATOS

Para esta práctica, se han evaluado varias bases de datos NoSQL con el objetivo de seleccionar la más adecuada para los requisitos planteados. A continuación, se analiza cada una de ellas, destacando sus características principales y justificando la elección final.

- **InfluxDB**

  InfluxDB es una base de datos diseñada específicamente para series temporales y métricas, optimizada para manejar grandes volúmenes de datos con marcas de tiempo. [2]

- Ventajas:
  - Buen rendimiento para consultas acerca de datos temporales
  - Eficiente para análisis de métricas en tiempo real

- Desventajas
  - Enfocado únicamente en series temporales
  - No permite modelar relaciones complejas entre entidades

- **Cassandra**

Cassandra es una base de datos distribuida orientada a columnas, diseñada para gestionar grandes volúmenes de datos con alta disponibilidad y tolerancia a fallos.

- Ventajas:
  - Alta escalabilidad horizontal, ideal para entornos distribuidos.
  - Excelente rendimiento para escrituras masivas y cargas de datos.

- Desventajas
  - No apto para búsquedas complejas
  - Requiere un diseño previo del modelo de datos asntes de su implementación.

- **MongoDB**

MongoDB es una base de datos orientada a documentos que utiliza el formato JSON binario (BSON) para almacenar datos jerárquicos y flexibles. Es conocida por su facilidad de uso y escalabilidad.

- Ventajas:
  - Permite almacenar datos semi-estructurados y estructurados con esquemas flexibles
  - Ofrece consultas dinámicas, soporte para índices y análisis detallados de los datos.
  - Escalabilidad horizontal para manejar grandes volúmenes de datos distribuidos.
  - Compatible con múltiples tipos de consultas, desde búsquedas simples hasta agregaciones complejas.

- Desventajas
  - Menos eficiente para escrituras masivas
  - Gestión de relaciones entre documentos requiere un diseño cuidadoso o desnormalización.

Para esta práctica, MongoDB es la opción más adecuada debido a la naturaleza de los datos y las necesidades del caso de uso. Los datos relacionados con jugadoras de fútbol incluyen atributos jerárquicos y relaciones complejas, que pueden ser modelados de manera natural en MongoDB gracias a su enfoque orientado a documentos.

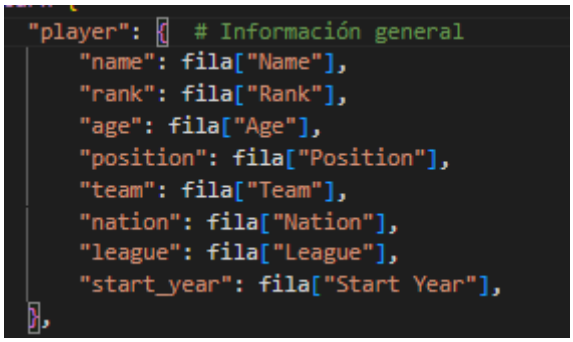
A diferencia de InfluxDB, que está limitada a datos de series temporales, MongoDB ofrece mayor versatilidad para trabajar con datos heterogéneos. Por otro lado, aunque Cassandra es excelente para la escalabilidad y las escrituras masivas, carece de la flexibilidad y las capacidades de consulta avanzadas que ofrece MongoDB. [1]

4. MODELADO DE LOS DATOS

Para realizar el modelado de los datos, se ha optado por utilizar un documento JSON con la siguiente estructura:

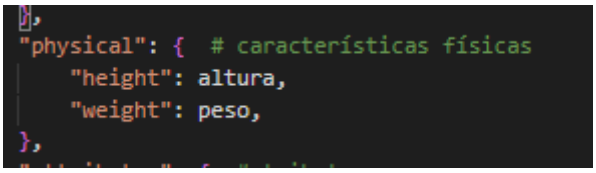
- **Información general de la jugadora**

  Esta sección recoge información básica sobre la jugadora, como su nombre, clasificación, edad, posición, equipo, nacionalidad, liga y el año de inicio en su carrera profesional.



- **Características Físicas**

  Incluye atributos físicos de la jugadora como su altura y peso.



- **Atributos Técnicos**

  Este apartado es el más detallado e incluye características específicas que describen las habilidades técnicas de la jugadora. Estas se agrupan en las siguientes categorías:

- Velocidad: aceleración, velocidad de sprint, agilidad.
- Disparo: finalización, potencia de tiro, tiros lejanos, voleas, penaltis.
- Pases: pases cortos, pases largos, visión, centros, precisión de tiros libres, curva.

- Defensa: intercepciones, precisión de cabeza, conciencia defensiva, entradas de pie y deslizantes.
- Físico: salto, resistencia, fuerza, agresividad.

```
},
"attributes": {  #atributes
    "overall": fila["OVR"],
    "speed": { #velocidad
        "acceleration": fila["Acceleration"],
        "sprint_speed": fila["Sprint Speed"],
        "agility": fila["Agility"]
    },
    "shooting": { #disparo
        "finishing": fila["Finishing"],
        "shot_power": fila["Shot Power"],
        "long_shots": fila["Long Shots"],
        "volleys": fila["Volleys"],
        "penalties": fila["Penalties"]
    },
    "passing": { #pases
        "short_passing": fila["Short Passing"],
        "long_passing": fila["Long Passing"],
        "vision": fila["Vision"],
        "crossing": fila["Crossing"],
        "free_kick_accuracy": fila["Free Kick Accuracy"],
        "curve": fila["Curve"]
    },
    "dribbling": { #regate
        "dribbling": fila["Dribbling"],
        "balance": fila["Balance"],
        "ball_control": fila["Ball Control"],
        "composure": fila["Composure"]
    },
    "defending": { #defensa
        "defense": fila["DEF"],
        "interceptions": fila["Interceptions"],
        "heading_accuracy": fila["Heading Accuracy"],
        "defensive_awareness": fila["Def Awareness"],
        "standing_tackle": fila["Standing Tackle"],
        "sliding_tackle": fila["Sliding Tackle"]
    },
    "physicality": { #fisico
        "jumping": fila["Jumping"],
        "stamina": fila["Stamina"],
        "strength": fila["Strength"],
        "aggression": fila["Aggression"]
    },
```

- **Estilo de Juego**

Por último, este apartado representa los estilos de juego específicos de la jugadora. Si esta información está disponible, se almacena como una lista de estilos.

```
},
"play_style": fila["play style"].split(", ") if pd.notnull(fila["play style"]) else [],  # Estilo de juego
```

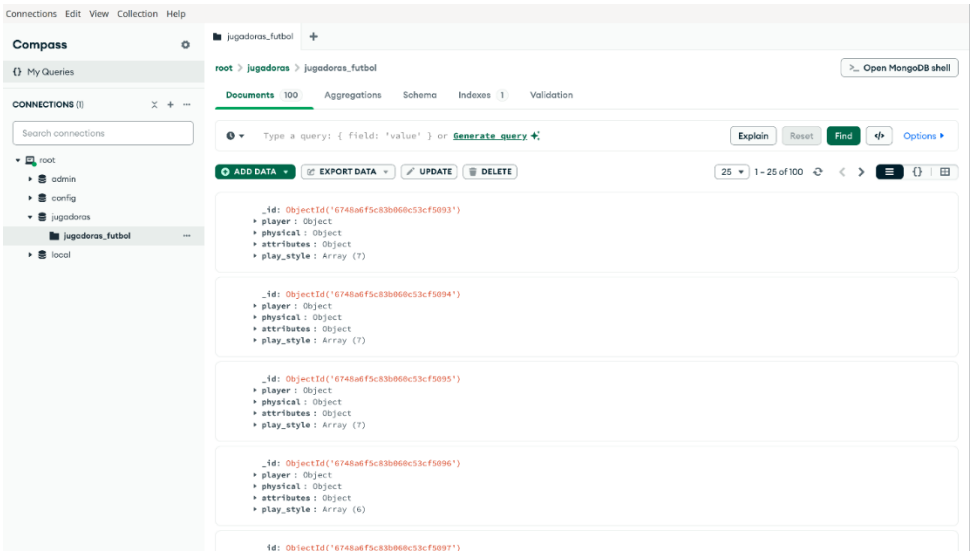
En este modelo se consideran los atributos más relevantes, eliminando aquellos valores nulos o irrelevantes que no aportan información significativa. Las columnas *Height* y *Weight* fueron procesadas para convertir sus valores en formato texto a valores numéricos en centímetros y kilogramos respectivamente.

5. INSERCIÓN DE LOS DATOS

Para cargar los datos en MongoDB, se desarrolló un script en python llamado practica2.py. Este script se encarga de leer un archivo CSV con la información de las jugadoras de fútbol, transformar las filas en documentos JSON siguiendo el modelo definido y, por último, insertar estos documentos en una base de datos MongoDB.

Los pasos que se han seguido para llevar esto a cabo son los siguiente:

- Leer el archivo con la biblioteca pandas y cada fila se transformó en un documento JSON utilizando la función transformar_fila.
- Realizar la conexión con la bbdd de MongoDB utilizando la biblioteca pymongo. Se crea una base de datos football_data con una colección llamada players.
- Se insertan los 100 primeros registros



6. SENTENCIAS DE MODIFICACIÓN Y CONSULTAS

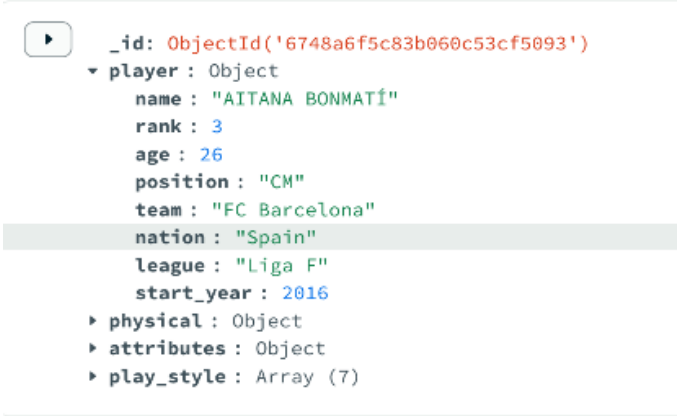
6.1 SENTENCIAS DE MODIFICACIÓN

Para realizar la actualización de los nombres de las jugadoras a mayúsculas en MongoDB, utilizamos el método update_one de la biblioteca pymongo. Buscamos el documento que contiene el nombre de la jugadora y actualizamos ese campo para que su nombre se almacene en mayúsculas.

```
jugadoras = [
    {"name": "Aitana Bonmatí"},
    {"name": "Alexia Putellas"},
]

# modificar el nombre de las jugadoras a mayúsculas
for jugadora in jugadoras:
    collection.update_one(
        {"player.name": jugadora["name"]},
        {"$set": {"player.name": jugadora["name"].upper()}}
    )
```

Resultado:



6.2 CONSULTA 1

Consulta: Filtrando por el año de comienzo en el football mayor de 2020

En esta consulta, se busca obtener las jugadoras cuyo año de comienzo (start_year) sea mayor que 2020. Para ello, se ha utilizado el operador $gt en la consulta, que permite filtrar los registros donde el valor sea mayor que 2020.

```
# ----- CONSULTA 1 : Jugadora con año de inicio mayor a 2020 -----
start_time = time.time()
consulta_1 = collection.find({
    "player.start_year": {"$gt": 2020}
})

# Guardar resultado en una lista
resultado_1 = []
for jugadora in consulta_1:
    resultado_1.append(jugadora)
end_time = time.time()
print(f"Tiempo consulta 1 (jugadoras iniciaron después de 2020): {end_time - start_time:.4f} segundos")
```

6.3 CONSULTA 2

Consulta: Filtrando que el equipo empiece con “Manchester.....”

Esta consulta busca a las jugadoras cuyo equipo comience con la palabra "Manchester". Para realizar esta consulta, podemos utilizar el operador $regex, que permite hacer búsquedas por expresión regular.

```
start_time = time.time()
consulta_2 = collection.find({
    "player.team": {"$regex": "^Manchester", "$options": "i"}
})

# Guardar el resultado en una lista
resultado_2 = []
for jugadora in consulta_2:
    resultado_2.append(jugadora)
end_time = time.time()
print(f"Tiempo consulta 2 (jugadoras cuyo equipo empieza con 'Manchester'): {end_time - start_time:.4f} segundos")
```

6.4 CONSULTA 3

Consulta: Consulta por un país concreto donde juega una jugadora

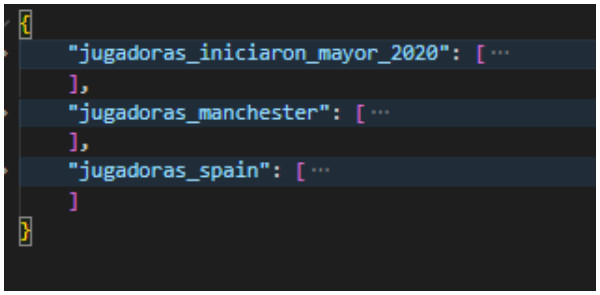
Para esta consulta se filtra por el campo campo “player.nation” sea igual a "Spain".

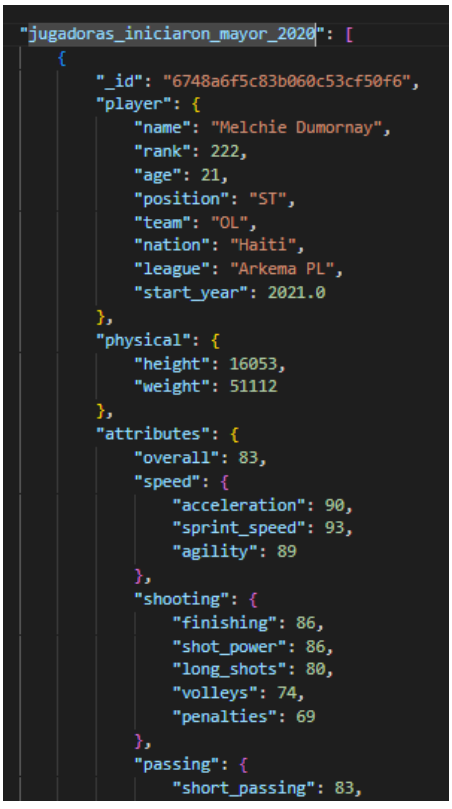
```
# ----- CONSULTA 3 : Jugadora de un país concreto
pais = "Spain"
start_time = time.time()  #captura timepo fin
consulta_3 = collection.find({
    "player.nation": pais
})

# Guardar el resultado en una lista
resultado_3 = []
for jugadora in consulta_3:
    resultado_3.append(jugadora)
end_time = time.time()  #captura timepo fin
print(f"Tiempo consulta 3 (jugadoras de España): {end_time - start_time:.4f} segundos")
```

```
(env) irene@irene:~/Documentos/MUCSI/big_data/PRACTICA2$ python consultas.py
Tiempo consulta 1 (jugadoras iniciaron después de 2020): 0.0233 segundos
Tiempo consulta 2 (jugadoras cuyo equipo empieza con 'Manchester'): 0.0013 segundos
Tiempo consulta 3 (jugadoras de España): 0.0013 segundos
Resultados almacenados en el json
(env) irene@irene:~/Documentos/MUCSI/big_data/PRACTICA2$
```

Para poder ver los resultados de estas consultas de una manera más organizada y visualmente más cómoda que mostrarlas por consola, se ha decidido generar un archivo JSON con todas las respuestas. A continuación, se muestra un imagen de la estructura de este JSON.





7. CÓDIGO

Para facilitar el acceso a esta práctica, todos los archivos utilizados, incluidos los scripts en Python y la generación del archivo JSON con las consultas, se encuentran disponibles en un repositorio de GitHub. Además, se incluye el dataset original utilizado para realizar este análisis.

Para su acceso únicamente es necesario acceder a la url que se me muestra a continuación, no es necesario pedir ningún permiso extra ya que es un repositorio público.

Github: https://github.com/IreeneGG/BigData/tree/master

8. BIBLIOGRAFíA

[1] *Compare Cassandra vs InfluxDB*. (s. f.). PeerSpot. https://www.peerspot.com/products/comparisons/cassandra_vs_influxdb_vs_mongodb

[2] InfluxData. (2021, 10 diciembre). *InfluxDB: Open Source Time Series Database | InfluxData*. https://www.influxdata.com/blog/influxdb-vs-cassandra-time-series/

[3] ChatGPT. (2024, 7 octubre). ChatGPT Español sin registro. https://chatgpt.es/ (**prompt**: https://chatgpt.com/share/6748a835-bdf4-8004-80e3-ec50b7252594)