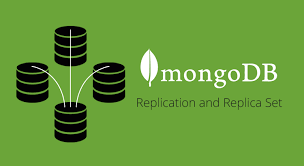
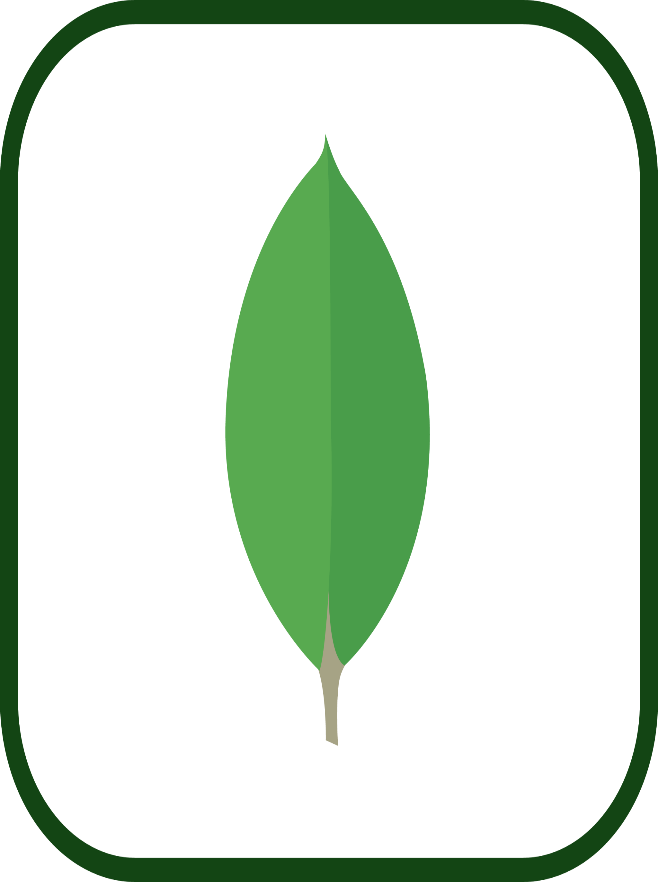
Consultas en MongoDB



Adrián García-Herras Herrero

Víctor Marín Escribano

Ariel Orlando Cahuaza Huanca

# Índice

[Índice 1](#_Toc199171045)

[Introducción 2](#_Toc199171046)

[EJER 1 3](#_Toc199171047)

[EJER 2 3](#_Toc199171048)

[EJER3 4](#_Toc199171049)

[EJER4 4](#_Toc199171050)

[EJER5 5](#_Toc199171051)

[EJER6 5](#_Toc199171052)

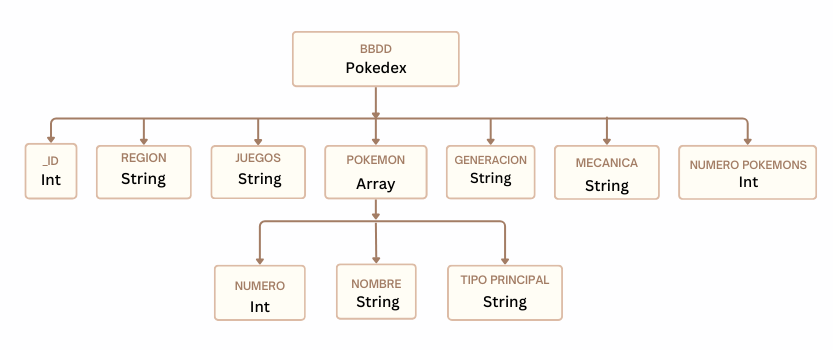
[EJER7 6](#_Toc199171053)

[EJER8 6](#_Toc199171054)

## Introducción

La base de datos de Pokedex tiene 10 documentos de las regiones de los juegos de pokemon, estas están compuestas por el \_id, Región, Juegos, Generación, Mecánica, Numero\_pokemons, Pokémon.

Por orden estas representan, el identificador en la base de datos, el nombre de la región, los juegos en los cuales la historia transcurre en dicha región, generación en la cual aparece, la mecánica o mecánicas que aporto a la saga, el número de pokemons que añadió y sus pokemons iniciales, estos están compuestos por su número de la pokedex, el nombre del pokemon y sus tipos.



Esquema de datos

## EJER 1

En este caso se quiere ordenar todos los pokemon por la región de manera alfabética. Usando find({}) para seleccionar todos para luego usar la proyección para excluir el id e incluir la región, juegos y generación. Y el sort para ordenarlos de manera alfabética.

Resultado:

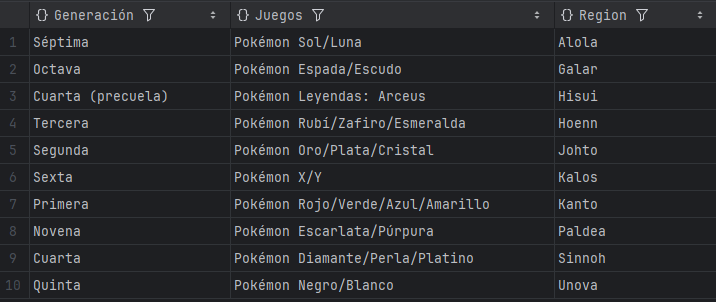


Ilustración 1-Resultado de la consulta del primer ejercicio

## EJER 2

En este caso buscamos la región con más de 150 pokemons. Y lo hacemos utilizando un find() para luego filtrarlo con $gt seleccionado número de pokemons mayor a 150 y en la proyección excluimos el id.

Resultado:

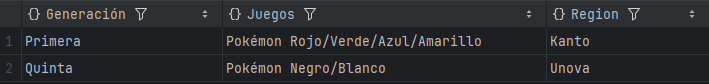


Ilustración 2-Resultado de la consulta del segundo ejercicio.

## EJER3

En este caso encontrar las regiones con los pokemons de tipo Agua y que en pokedex sean superior a 500.

Para ello usamos find() con un $eleMatch para buscarlas en los arrays que sean de tipo: Agua y el Numero con $gt mayor a 500 y excluyendo en la proyección el id y la mecánica.

Resultado:

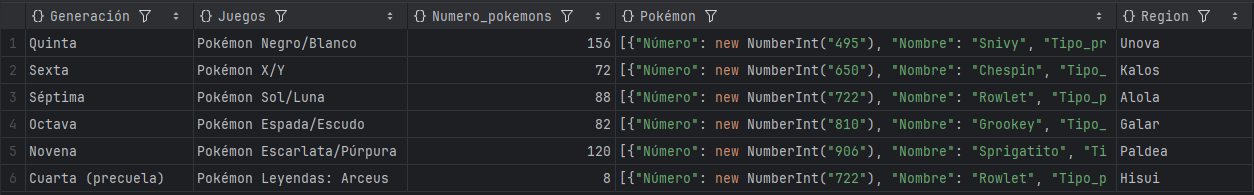


Ilustración 3-Resultado de la consulta del tercer ejercicio

## EJER4

En este caso buscamos obtener la región con más juegos con más de 100 pokemons y donde no aparezcan Hoenn y que no sean de la 1 generación.

Usaremos un find con los operadores $gt , $not ($eq) para seleccionar los que tienen más de 100 pokemon, la región hoenn y la primera generación. Y descartamos el id.

Resultado:

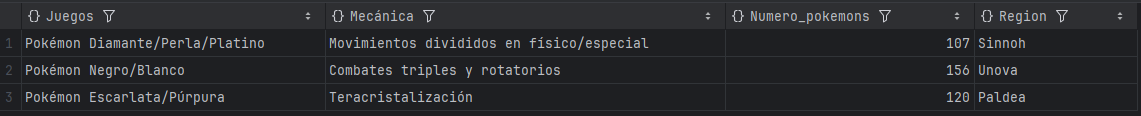


Ilustración 4-Resultado de la consulta del cuarto ejercicio.

## EJER5

Filtra regiones con más de 100 pokémons, ordena por región y limita a 5 resultados

Usamos el aggregate con un match y gt para filtrar el número de pokemons > 100, el sort para ordenarlos por de manera alfabéticamente y el limit mostrar los 5 primeros.

Resultado:

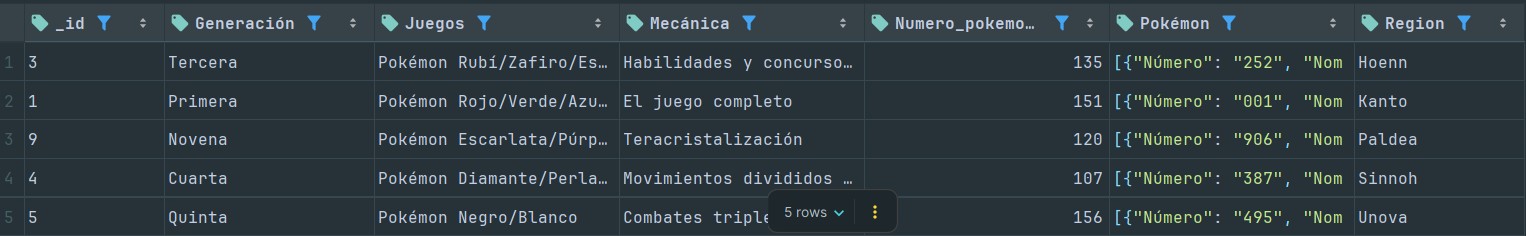


Ilustración 5-Resultado de la consulta del quinto ejercicio.

## EJER6

Divide regiones en pares/impares (según \_id) y calcula media de pokémons por grupo y un total de regiones por grupo

Resultado:

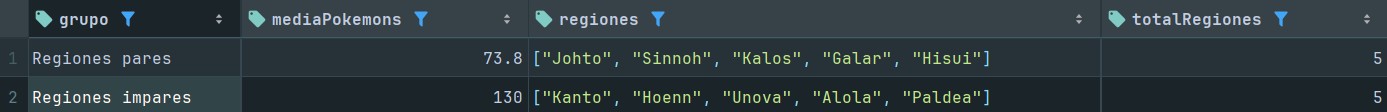


Ilustración 6-Resultado de la consulta del sexto ejercicio.

## 

## EJER7

Descompone array Pokémon y filtra solo tipos de Fuego

En este caso usaremos un aggregate con un unwind para generar un archivo individual de cada pokemon, el match para filtrarlo el tipo fuego. Y con proyect para mostrar la región, nombre y el tipo.

Resultado:



Ilustración 7-Resultado de la consulta del séptimo ejercicio.

## EJER8

Calcula densidad de pokémons (nº Pokémon/100) y crea descripción concatenada indicando la región y la generación

En este caso con proyect creamos dos campos calculados, el primero par calcular la densidad de los pokemon con $divide y la segunda para una descripción con $concat de la región y generación.

Resultado:

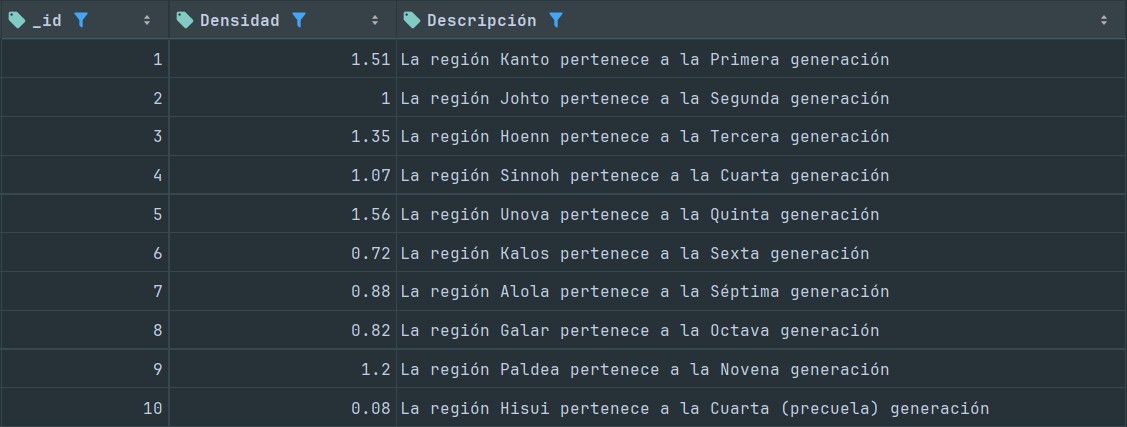


Ilustración 8-Resultado de la consulta del octavo ejercicio.