

Full Stack Bootcamp

Programación & Algorítmica & Proyectos Software

Introducción a BBDD

Bases de Datos avanzadas

Javascript avanzado & EcmaScript 6

MongoDB

Base de datos No Relacionales

- Especifico para aplicaciones con:
 - Grandes volúmenes
 - Baja latencia
 - Modelo de datos flexibles
 - Flexibilidad en coherencia de datos
- Principales características:
 - Flexibilidad
 - Escalabilidad
 - Alto rendimiento
 - Altamente funcional



Azure Cosmos DB



Base de datos No Relacionales (II)

- Base de datos Ecommerce (SQL o Relacional)
 - Tabla Usuarios
 - Tabla Productos (Relacionado con Vendedores id_vendedor)
 - Tabla Vendedores
 - Tabla Ventas (Relacionado con Usuarios id_usuario y Productos id_producto)
- Base de datos Ecommerce (NoSQL o no relacional)
 - ?

Base de datos No Relacionales (III)

- Tipos de base de datos NoSQL
 - Clave-Valor, como el map de JavaScript
 - Documentos, se suele guardar en formato JSON
 - Gráficos, datos no estructurados y conexión de diversas fuentes de datos
 - En memoria, trabajan muy rápido con datos que no necesitan almacenarse
 - Búsqueda, basado en indexación de datos para búsquedas rápidas

NoSQL - Documentos

- Documentos de tipo JSON

```
let usuario = {  
  nombre: "Mariano",  
  apellidos: "Rodriguez Lozano",  
  dirección: "C/ Malaga 12, 4D, 29007"  
}
```

Ejercicio: crear el modelo de la base de datos Ecommerce en documentos JSON.

MongoDB – Instalación

- Seguimos las instrucciones en

<https://docs.mongodb.com/v4.4/tutorial/install-mongodb-on-windows/>

- Instalar como aplicación y no servicio
- No instalar MongoDB Compass

MongoDB – Comandos básicos

- *db*, te muestra la base de datos actual en uso (test por defecto)
- *show dbs*, muestra las bases de datos disponibles
- *use <database>*, cambia a otra base de datos (la crea si no existe)
- *show collections*, muestra las colecciones de la base de datos
- *db.myCollection.insertOne({ x: 1 })*;; inserta un documento en la colección (si la colección no existe la crea vacía)
- *db.myCollection.find()*, lista todos los documentos de la colección myCollection

MongoDB – Nuestra primera base de datos

- Replicar nuestra base de datos imdb en MongoDB
 - Tablas -> Collections
 - Columnas -> clave de cada par clave-valor
 - Filas -> valores de cada par clave-valor

Ejemplo:

```
usuario = { columna1: "fila1-valor",  
            columna2: "fila1-valor"}  
usuario2 = { columna1: "fila2-valor",  
            columna2: "fila2-valor"  
}
```


MongoDB – insertOne e insertMany

- imdb.contenidos.insertOne({ ... })
- imdb.contenidos.insertMany([{ ... }, { ... }])

imdb.contenidos.find();

- Columna ID de cada documento
 - Valores que tiene
 - Se crea por defecto si no se indica

MongoDB – Listar documentos

```
imdb.contenidos.find();
```

```
SELECT * FROM contenidos;
```

```
imdb.contenidos.find({ tipo: "Serie" })
```

```
SELECT * FROM contenidos WHERE tipo like "%Serie%"
```

```
imdb.contenidos.find({ tipo: "Serie", rating: { $lt: 8 } })
```

```
SELECT * FROM contenidos WHERE tipo like "%Serie%" and rating < 8
```

MongoDB – Filtrando documentos

```
db.contenidos.find({ tipo: $in: [ "Serie", "Show" ] });
```

```
SELECT * FROM contenidos WHERE tipo in ("Serie", "Show");
```

```
db.contenidos.find({ $or: [{ tipo: "Serie"}, { rating: { $lt: 9 } } ] } )
```

```
SELECT * FROM contenidos WHERE tipo like "%Serie%" or rating < 8
```

```
db.contenidos.find({ $or: [{ tipo: "Serie" }, { tipo: "Show" }], { rating: { $lt: 10 } } })
```

```
SELECT * FROM contenidos WHERE tipo like "%Serie%" or tipo like "%Show%" and  
rating < 8
```

MongoDB – Base de datos Ecommerce

Ejercicio: Replicar base de datos Ecommerce

Replicar los ejercicios de MySQL

- Obtener el listado de ventas con nombre de usuario y el tipo de pago realizado (si la colección ventas no tiene tipo de pago, incluirlo)
- Sacar por consola lista de productos donde aparezca su nombre y la cantidad vendida
- Sacar por consola Lista de productos que se muestre el average del rating
- Sacar por consola cual es el producto mas vendido y nombre de usuarios han comprado dicho producto

*orderby <https://docs.mongodb.com/manual/reference/operator/meta/orderby/>
limit <https://docs.mongodb.com/manual/reference/operator/aggregation/limit/>*

- Sacar por consola el producto con su nombre y tipo que mayor ganancia ha generado
- Sacar por consola el nombre de usuario que mas artículos ha comprado
- Sacar por consola el nombre de usuario que se ha gastado mas dinero. Incluir el símbolo €.

Para solo mostrar rating y nombre *db.contenidos.find({ tipo: "Serie" }, { _id: 0, nombre: true, rating: true })*

MongoDB – NodeJS y JavaScript

- Pasos para la instalación

```
npm install mongodb
```

```
const { MongoClient } = require('mongodb');

// Conectar a la base de datos
const client = new MongoClient("mongodb://localhost:27017");
await client.connect();
await client.db("ecommerce").command({ ping: 1 });
console.log("Conectado a MongoDB");

// desconectar con la base de datos
await client.close();
console.log("Desconectado de la base de datos");
```

- Solicitar Datos:

```
const usuarios =
client.db("ecommerce").collection(
"usuarios").find();

for (const u of usuarios) {
    console.log(u);
}
```