**INTRODUCCIÓN BASES DE DATOS NOSQL- MONGODB**

1. Consulte y realice un cuadro comparativo sobre los tipos de dato que se pueden usar dentro de Mongo DB, indicando el nombre, su descripción o uso y un ejemplo de representación de este.

|  |  |  |
| --- | --- | --- |
| Tipo de Dato | Descripción / Uso | Ejemplo de Representación |
| String | Cadena de texto: Almacena Texto | { nombre: "Carlos" } |
| Integer | Números enteros: Almacena valores numéricos sin decimales. | {edad: 30} |
| Double | Números de punto flotante. Almacena valores numéricos con decimales. | {precio: 19.99} |
| Bool | Verdadero o falso. | {activo: true} |
| Date | Fecha y hora: Almacena fecha y hora con milisegundos. | { fecha: ISODate("2023-08-01T00:00:00Z") } |
| Object | Documento embebido: Representa una estructura anidada. | {direccion: {ciudad: "Lima", pais: "Perú"}} |
| Array | Lista ordenada de valores: Puede contener distintos tipos. | { materias: ["Matemáticas", "Inglés"] } |
| ObjectId | Identificador único de documento: Id unico generado por Mongo | { \_id: ObjectId("64e0...") } |
| Null | Representa un valor nulo o vacío. | {comentario:null} |
| Binary Data | Datos binarios como imágenes o archivos codificados. | { archivo: BinData(0,"base64...") } |
| Code | Código JavaScript que puede ejecutarse en el servidor. | { funcion: function() { return true; } } |
| Timestamp | Marca de tiempo especial para control interno. | |  | | --- | |  |  |  | | --- | | { creadoEn: Timestamp(1, 100) } | |

1. Consulte el nombre, la definición y uso de cada uno de los operadores propuestos para trabajo con consultas en Mongo DB.

|  |  |  |
| --- | --- | --- |
| **OPERADOR** | **NOMBRE** | **DEFINICIÓN** |
| $eq | Igual | db.productos.find({precio: {$eq: 60000}}) |
| $gt | Mayor que | db.productos.find({precio: {$gt: 100000}}) |
| $gte | Mayor o igual que | db.productos.find({cantidad: {$gte: 20}}) |
| $lt | Menor que | db.productos.find({precio: {$lt: 100000}}) |
| $lte | Menor o igual que | db.productos.find({cantidad: {$lte: 15}}) |
| $ne | Distinto de | db.productos.find({nombre: {$ne: "USB"}}) |
| $in | Dentro de una lista | db.productos.find({nombre: {$in: ["USB", "Laptop"]}}) |
| $nin | No en una lista | db.productos.find({nombre: {$nin: ["Mouse", "USB"]}}) |
| $or | O | db.productos.find({$or: [{precio: {$lt: 90000}}, {cantidad: {$gt: 50}}]}) |
| $and | Y | db.productos.find({$and: [{precio: {$gt: 100000}}, {cantidad: {$lt: 20}}]}) |
| $not | No (niega condición) | db.productos.find({precio: {$not: {$lt: 100000}}}) |
| $nor | Ni una ni otra | db.productos.find({$nor: [{precio: {$gt: 100000}}, {cantidad: {$gt: 50}}]}) |
| $exists | Existe campo | db.productos.find({cantidad: {$exists: true}}) |
| $type | Tipo del dato | db.productos.find({precio: {$type: "number"}}) |
| $mod | Modulo (residuo división) | db.productos.find({precio: {$mod: [2000, 0]}}) |
| $regex | Expresión regular | db.productos.find({nombre: {$regex: /top/i}}); |
| $elemMatch | Coincidencia elementos (array) | db.estudiantes.find({notas: {$elemMatch: {nota: {$gt: 4.5}}}}) |
| $size | Tamaño (array) | db.estudiantes.find({materias: { $size: 3 }}) |

**Base de datos Usada:**

use Tienda

db.createCollection(“productos”)

db.productos.insertMany([

{nombre:"Mouse",precio:60000,cantidad:45},

{nombre:"Teclados",precio:100000,cantidad:15},

{nombre:"USB",precio:130000,cantidad:19},

{nombre:"Laptop",precio:120000,cantidad:59}

])

db.createCollection(“estudiantes”)

db.estudiantes.insertMany([

{ nombre: "Laura Gómez", notas: [ {materia: "Matemáticas", nota: 4.8}, {materia: "Español", nota: 4.2}, {materia: "Inglés", nota: 3.9}]},

{nombre: "Carlos Ruiz", notas: [{materia: "Matemáticas", nota: 3.2},{materia: "Español", nota: 2.8},{materia: "Inglés", nota: 3.5}]}

]);