

# Sistemas de Gestión de Base de Datos con Python Mongo DB

Tito Reátegui  
`treateguig@bsginstitute.com`

2025-09-19

## 1. Introducción

**NoSQL** es una familia de sistemas de gestión de bases de datos no relacionales diseñados para manejar grandes volúmenes de datos, escalabilidad horizontal y flexibilidad en el modelado de información. Se utilizan ampliamente en aplicaciones web modernas, big data, IoT y soluciones que requieren respuesta en tiempo real.

## 2. Tipos de Bases de Datos NoSQL

- ▶ **Documentales:** Almacenan datos en documentos generalmente con formato JSON o BSON. Ejemplo: MongoDB, CouchDB.
- ▶ **Clave-Valor:** Guardan datos como pares clave-valor. Ejemplo: Redis, DynamoDB.
- ▶ **Columnares:** Organizan los datos por columnas, ideales para grandes volúmenes analíticos. Ejemplo: Cassandra, HBase.
- ▶ **Grafos:** Optimizadas para relaciones complejas entre datos. Ejemplo: Neo4j, Amazon Neptune.

### 3. Ventajas de NoSQL

- ▶ Escalabilidad horizontal (fácil añadir servidores)
- ▶ Modelado flexible y sin esquemas estrictos
- ▶ Ideal para big data y datos no estructurados
- ▶ Optimización para aplicaciones distribuidas y en la nube<sup>[1]</sup>.

## 4. Mongo DB

**Objetivo:** Proporcionar una introducción práctica a MongoDB, desde la instalación hasta operaciones avanzadas como índices, con ejemplos por tema. El curso se enfoca en MongoDB como base de datos NoSQL, y se menciona brevemente PostgreSQL (base de datos SQL) para comparación en secciones relevantes, ya que el usuario lo solicitó. Sin embargo, el énfasis principal es en MongoDB.

**Requisitos:** Computadora con acceso a internet, sistema operativo (Windows, macOS o Linux).

**Herramientas:** MongoDB Community Edition, MongoDB Shell (mongosh), MongoDB Compass (opcional).

## 4. Mongo DB

### 4.1. Instalación de MongoDB

En esta sección, instalamos MongoDB y verificamos su funcionamiento. MongoDB es una base de datos NoSQL orientada a documentos, a diferencia de PostgreSQL, que es relacional y usa tablas con esquemas fijos.

## 4. Mongo DB

### Pasos de Instalación:

- ▶ **Descarga:** Ve al sitio oficial de MongoDB (<https://www.mongodb.com/try/download/community>) y selecciona la versión Community Edition para tu SO.
- ▶ **Instalación en Windows/macOS/Linux:**
  - ▶ Windows: Ejecuta el instalador MSI y sigue las instrucciones. Instala como servicio.
  - ▶ macOS: Usa Homebrew: `brew tap mongodb/brew` y `brew install mongodb-community`.
  - ▶ Linux (Ubuntu): Agrega el repositorio y ejecuta `sudo apt install mongodb-org`.
- ▶ **Iniciar el servidor:** Ejecuta `mongod` en la terminal (o inicia el servicio).
- ▶ **Instalar MongoDB Shell (mongosh):** Descárgalo desde <https://www.mongodb.com/try/download/shell>.
- ▶ **Verificación:** Abre la shell con `mongosh` y ejecuta `db.version()` para confirmar.

## 4. Mongo DB

### Comparación breve con PostgreSQL:

PostgreSQL se instala similarmente (e.g., `sudo apt install postgresql` en Linux). Para trabajar con PostgreSQL: Conéctate con `psql -U postgres`, crea una BD con `CREATE DATABASE mi_bd;`. MongoDB es más flexible sin esquemas estrictos, mientras PostgreSQL requiere definiciones de tablas.

### Ejemplo en MongoDB:

```
# Iniciar servidor (en otra terminal)
```

```
mongod
```

```
# En mongosh
```

```
mongosh
```

```
> db.test.insertOne({ mensaje: "Instalación exitosa" })
```



## 4. Mongo DB

### 4.2. Introducción a MongoDB

MongoDB almacena datos en documentos JSON-like (BSON).  
Conceptos clave: Bases de datos, colecciones (como tablas), documentos (como filas).

**Diferencia con PostgreSQL:** PostgreSQL usa SQL y relaciones; MongoDB usa consultas en JS y es escalable horizontalmente.

**Ejemplo:**

```
// En mongosh  
use mi_base_de_datos; // Crea o usa la BD  
db.mi_coleccion.insertOne({ nombre: "Ejemplo", edad: 30 });
```

## 4. Mongo DB

### 4.3. Entendiendo los Básicos y Operaciones CRUD

CRUD: Create (Crear), Read (Leer), Update (Actualizar), Delete (Eliminar). En MongoDB, se realizan en colecciones.

**Comparación con PostgreSQL:** En PostgreSQL, usa INSERT, SELECT, UPDATE, DELETE con SQL.

## 4. Mongo DB

### 4.3. Entendiendo los Básicos y Operaciones CRUD

#### Ejemplos por operación:

##### ► Create:

```
db.productos.insertMany([  
  { nombre: "Laptop", precio: 1000 },  
  { nombre: "Teléfono", precio: 500 }  
]);
```

## 4. Mongo DB

### 4.3. Entendiendo los Básicos y Operaciones CRUD

#### ► Read:

```
db.productos.find({ precio: { $gt: 600 } }).pretty();
```

#### ► Update:

```
db.productos.updateOne({ nombre: "Laptop" }, { $set: {
```

## 4. Mongo DB

### 4.3. Entendiendo los Básicos y Operaciones CRUD

#### ► Delete:

```
db.productos.deleteOne({ nombre: "Teléfono" });
```

## 4. Mongo DB

### 4.4. Esquemas y Relaciones: Cómo Estructurar Documentos

MongoDB no fuerza esquemas, pero puedes usar Mongoose para validación. Relaciones: Embebidas o referenciadas.

**Comparación con PostgreSQL:** PostgreSQL usa claves foráneas para joins; MongoDB usa agregaciones o refs manuales.

## 4. Mongo DB

### 4.4. Esquemas y Relaciones: Cómo Estructurar Documentos

#### **Ejemplo de esquema embebido:**

```
db.usuarios.insertOne({  
  nombre: "Juan",  
  direcciones: [  
    { ciudad: "Madrid", pais: "España" }  
  ]  
});
```

## 4. Mongo DB

### 4.4. Esquemas y Relaciones: Cómo Estructurar Documentos

#### Ejemplo de referencia:

```
// Colección posts
db.posts.insertOne({ titulo: "Post1",
                    autor_id: ObjectId("autor_id_aquí") });

// Consulta con agregación
db.posts.aggregate([
  { $lookup: { from: "usuarios",
               localField: "autor_id", foreignField: "_id", as: "
]};
```

## 4. Mongo DB

### 4.5. Explorando el Shell y el Servidor

El shell (mongosh) es para comandos interactivos. El servidor (mongod) maneja conexiones.

#### **Cómo trabajar con PostgreSQL (comparación):**

En PostgreSQL, usa `psql` como shell: `SELECT * FROM tabla;`.

#### **Ejemplos en Shell:**

- ▶ Conectar a servidor: `mongosh --host localhost --port 27017`
- ▶ Listar BDs: `show dbs;`
- ▶ Monitorear servidor: Usa `db.serverStatus()` para stats.



## 4. Mongo DB

### 4.6. Usando MongoDB Compass para Explorar Datos Visualmente

Compass es una GUI para visualizar datos.

**Instalación:** Descarga desde

<https://www.mongodb.com/products/tools/compass>. - Conecta a `mongodb://localhost:27017`. - Explora colecciones gráficamente.

**Ejemplo:** En Compass, inserta un documento vía interfaz y filtra queries.

**Comparación:** PostgreSQL tiene pgAdmin como GUI equivalente.

## 4. Mongo DB

### 4.7. Profundizando en Operaciones de Creación

Métodos: `insertOne()`, `insertMany()`.

**Ejemplo avanzado:**

```
db.inventario.insertMany([
  { item: "libro", qty: 10, tags: ["educación"] },
  { item: "pluma", qty: 50, tags: ["oficina"] }
], { ordered: false });
```

## 4. Mongo DB

### 4.8. Operaciones de Lectura - Un Vistazo Más Cercano

Usa find(), proyecciones, cursores.

**Ejemplo con filtro y sorting:**

```
db.inventario.find({ qty: { $lt: 20 } },  
                   { item: 1, qty: 1 }).sort({ qty: -1 }).limit(10)
```

**Agregación:**

```
db.ventas.aggregate([  
  { $group: { _id: "$producto",  
              total: { $sum: "$cantidad" } } }  
]);
```

## 4. Mongo DB

### 4.9. Operaciones de Actualización

Usa \$set, \$inc, etc.

**Ejemplo:**

```
db.usuarios.updateMany(  
  { edad: { $lt: 18 } },  
  { $set: { estado: "menor" }, $inc: { puntos: 5 } }  
);
```

## 4. Mongo DB

### 4.10. Entendiendo Operaciones de Eliminación

`deleteOne()`, `deleteMany()`.

**Ejemplo:**

```
db.productos.deleteMany({ precio: { $lt: 100 } });
```

## 4. Mongo DB

### 4.11. Trabajando con Índices

Índices mejoran queries. Crea con `createIndex()`.

**Ejemplo:**

```
db.productos.createIndex({ nombre: 1 }); // Índice ascendente  
db.productos.createIndex({ ubicacion: "2dsphere" }); // Geoespacial
```

**Explicar query:**

```
db.productos.explain().find({ nombre: "Laptop" });
```

# Notas Finales:

- ▶ **Práctica:** Dedicar tiempo a ejecutar ejemplos.
- ▶ **Comparación con PostgreSQL:** MongoDB es ideal para datos no estructurados; PostgreSQL para transacciones ACID estrictas. Para PostgreSQL, usa SQL: e.g., `UPDATE tabla SET columna = valor WHERE condicion;`
- ▶ **Recursos adicionales:** Documentación oficial de MongoDB ([docs.mongodb.com](https://docs.mongodb.com)). Para extender el curso, agrega ejercicios.