



INSTITUTO TECNOLÓGICO DEL PUTUMAYO



IES Vigilada por:  
**Educación**

## PROGRAMA INGENIERIA DE SISTEMAS

Página 1 | 18

# Informe Base de Datos NoSQL

Presentado por: Jhon Esteban Josa Quinchoa  
Maicol Sebastián Guerrero López  
Presentado a: Ing. Brayan Ignacio Arcos Burbano

Bases De Datos Y  
Almacenamiento Masivo

El **Saber** como **Arma** de **Vida**

## PROGRAMA INGENIERIA DE SISTEMAS

Página 2 | 18

### Índice

<b>Resumen Ejecutivo .....</b>	<b>4</b>
<b>Introducción .....</b>	<b>5</b>
<b>Metodología .....</b>	<b>6</b>
<b>Herramientas Utilizadas .....</b>	<b>6</b>
1. MongoDB .....	6
2. MongoDB Compas .....	6
<b>Informe de la Base de Datos NoSQL Biblioteca .....</b>	<b>7</b>
1. Creación de la base de datos Biblioteca .....	7
2. Creación de las colecciones libros, usuarios y prestamos .....	7
3. Inserción de Documentos: .....	7
3.1. Libros .....	7
3.2. Usuarios .....	7
3.3. Prestamos .....	8
<b>Descripción de la Base de Datos .....</b>	<b>8</b>
1. Libros .....	8
2. Usuarios .....	9
3. Préstamos .....	9
<b>Consultas NoSQL .....</b>	<b>10</b>
1. Todos los libros que han sido prestados a un usuario específico .....	10
2. Todos los libros que aún no han sido devueltos de un usuario específico .....	11
3. Todos los libros que han sido devueltos de un usuario específico. ....	11
4. Todos los prestamos realizados a los usuarios .....	12
5. Todos los prestamos realizados a los usuarios no devueltos .....	13
6. Historial de préstamos de un libro específico .....	14
<b>Diseño de Base de Datos .....</b>	<b>15</b>
<b>Análisis y Discusión .....</b>	<b>16</b>



INSTITUTO TECNOLÓGICO DEL PUTUMAYO



IES Vigilada por:  
**Educación**

## PROGRAMA INGENIERIA DE SISTEMAS

Página 3 | 18

<b>Conclusiones .....</b>	<b>17</b>
<b>Referencias.....</b>	<b>18</b>



El **Saber** como **Arma** de **Vida**



## PROGRAMA INGENIERIA DE SISTEMAS

Página 4 | 18

### Resumen Ejecutivo

Este informe documenta el proceso de creación de una base de datos no relacional en MongoDB para gestionar el sistema de una biblioteca. Se crearon tres colecciones: Libros, Usuarios y Préstamos. Se describen los datos insertados en cada colección y las consultas realizadas sobre los mismos. Además, se proporcionan los resultados de dichas consultas y se explica su lógica. Finalmente, se discuten las consideraciones de diseño y los resultados obtenidos.



El **Saber** como **Arma** de **Vida**



## PROGRAMA INGENIERIA DE SISTEMAS

Página 5 | 18

### Introducción

El presente informe se realiza con el objetivo de estudiar y aplicar conceptos relacionados con bases de datos no relacionales, utilizando MongoDB como herramienta principal. El tema es relevante debido a la creciente demanda de soluciones de almacenamiento de datos flexibles y escalables en aplicaciones modernas, también cubre aspectos de diseño, consultas y optimización en el contexto de una base de datos no relacional en MongoDB. Se incluyen detalles sobre la estructura de la base de datos, las consultas realizadas y la explicación de la lógica detrás de cada consulta.

El objetivo principal del informe es documentar la creación de una base de datos en MongoDB, incluyendo el diseño, inserción de datos y consultas. Se busca demostrar cómo MongoDB puede ser utilizado para manejar datos en una biblioteca, facilitando el acceso y la gestión eficiente de información.

El **Saber** como **Arma de Vida**

# PROGRAMA INGENIERIA DE SISTEMAS

Página 6 | 18

## Metodología

## Herramientas Utilizadas

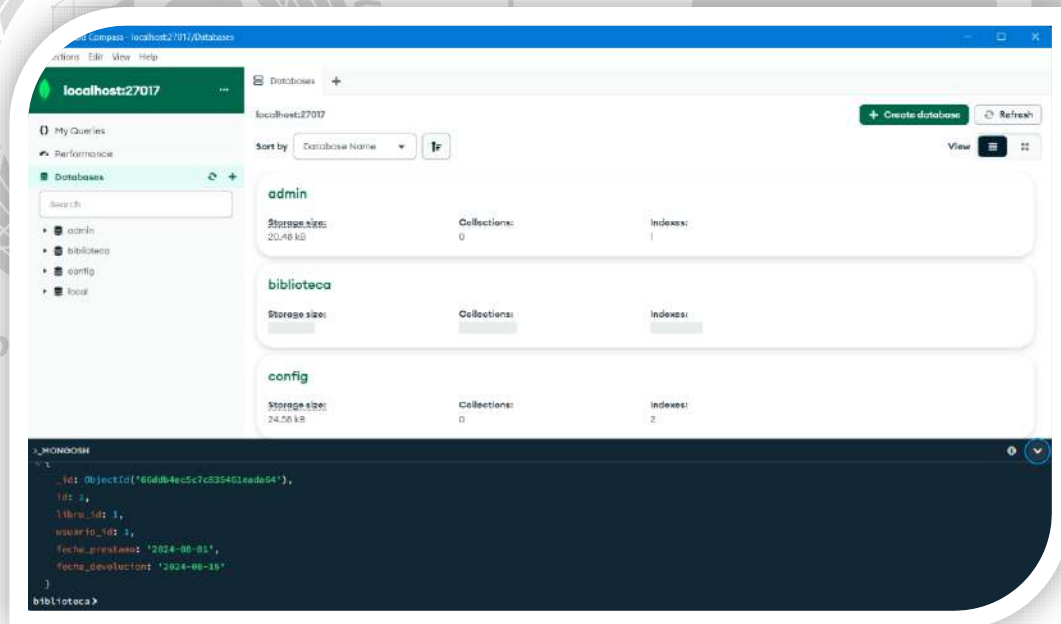
Para llevar a cabo este proyecto se utilizaron las siguientes herramientas:

1. **MongoDB:** Sistema de base de datos no relacional para almacenar los datos.



mongoDB®

2. **MongoDB Compass:** Herramienta gráfica para gestionar y realizar consultas sobre la base de datos.





## PROGRAMA INGENIERIA DE SISTEMAS

Página 7 | 18

### Informe de la Base de Datos NoSQL Biblioteca

#### 1. Creación de la base de datos Biblioteca

#### 2. Creación de las colecciones libros, usuarios y prestamos



```
use biblioteca
```



```
db.createCollection("libros")  
db.createCollection("usuarios")  
db.createCollection("prestamos")
```

#### 3. Inserción de Documentos:

##### 3.1. Libros



```
db.libros.insertMany([  
  {  
    _id: 1,  
    "titulo": "Cien Años de Soledad",  
    "autor": "Gabriel García Márquez",  
    "año_publicacion": 1967,  
    "genero": "Realismo Mágico"  
  },  
  {  
    _id: 2,  
    "titulo": "1984",  
    "autor": "George Orwell",  
    "año_publicacion": 1949,  
    "genero": "Distopía"  
  },  
  {  
    _id: 3,  
    "titulo": "El Gran Gatsby",  
    "autor": "F. Scott Fitzgerald",  
    "año_publicacion": 1925,  
    "genero": "Ficción" }  
]);
```

##### 3.2. Usuarios



```
db.usuarios.insertMany([  
  {  
    _id: 1,  
    "nombre": "Juan Pérez",  
    "email": "juan.perez@gmail.com",  
    "direccion": "Calle 123 A",  
    "telefono": "3001234567"  
  },  
  {  
    _id: 2,  
    "nombre": "Ana Gómez",  
    "email": "ana.gomez@gmail.com",  
    "direccion": "Avenida 45 B",  
    "telefono": "3009876543"  
  },  
  {  
    _id: 3,  
    "nombre": "Luis Rodríguez",  
    "email": "luis.rodriguez@gmail.com",  
    "direccion": "Calle 78 A",  
    "telefono": "3111239876"  
  }  
]);
```

### 3.3. Prestamos

```
db.prestamos.insertMany([
  // Usuario 1
  {
    _id: 1,
    usuario_id: 1,
    libro_id: 1,
    fecha_prestamo: new Date('2024-01-01'),
    fecha_devolucion: new Date('2024-01-10')
  },
  {
    _id: 2,
    usuario_id: 1,
    libro_id: 4,
    fecha_prestamo: new Date('2024-02-05')
  }, // Aún no devuelto
  {
    _id: 3,
    usuario_id: 1,
    libro_id: 7,
    fecha_prestamo: new Date('2024-03-10')
  }, // Aún no devuelto
]);
```

#### Descripción de la Base de Datos

La base de datos consta de las siguientes colecciones:

1. **Libros:** Almacena información sobre los libros disponibles en la biblioteca.

```
{
  "_id": 1,
  "titulo": "Cien Años de Soledad",
  "autor": "Gabriel García Márquez",
  "año_publicacion": 1967,
  "genero": "Realismo Mágico"
}
```



## PROGRAMA INGENIERIA DE SISTEMAS

Página 9 | 18

2. **Usuarios:** Contiene datos sobre los usuarios que tienen acceso a la biblioteca.

```
{
  "_id": 1,
  "nombre": "Juan Pérez",
  "email": "juan.perez@gmail.com",
  "direccion": "Calle 123 A",
  "telefono": "3001234567"
}
```

3. **Préstamos:** Registra los préstamos de libros realizados por los usuarios.

```
{
  "_id": 1,
  "usuario_id": 1,
  "libro_id": 1,
  "fecha_prestamo": {
    "$date": "2024-01-01T00:00:00.000Z"
  },
  "fecha_devolucion": {
    "$date": "2024-01-10T00:00:00.000Z"
  }
}
```

## PROGRAMA INGENIERIA DE SISTEMAS

Página 10 | 18

### Consultas NoSQL

#### 1. Todos los libros que han sido prestados a un usuario específico

```
db.prestamos.aggregate([
  { $match: { usuario_id: 1 } },
  { $lookup: {
    from: "libros",
    localField: "libro_id",
    foreignField: "_id",
    as: "detalle_libro"
  }},
  { $unwind: "$detalle_libro" },
  { $project: {
    _id: 1,
    titulo: "$detalle_libro.titulo",
    fecha_prestamo: 1,
    fecha_devolucion: 1
  }}
]);
```

```
{
  _id: 1,
  fecha_prestamo: 2024-01-01T00:00:00.000Z,
  fecha_devolucion: 2024-01-10T00:00:00.000Z,
  titulo: 'Cien Años de Soledad'
}
{
  _id: 2,
  fecha_prestamo: 2024-02-05T00:00:00.000Z,
  titulo: 'Matar a un Ruiseñor'
}
{
  _id: 3,
  fecha_prestamo: 2024-03-10T00:00:00.000Z,
  titulo: 'Los Pilares de la Tierra'
}
```

- **\$match:** Filtra los préstamos del usuario con usuario\_id: 1.
- **\$lookup:** Realiza una "join" con la colección libros para obtener la información de los libros asociados al préstamo.
- **\$unwind:** Despliega los detalles de cada libro en documentos individuales.
- **\$project:** Selecciona el ID del préstamo, el título del libro, la fecha de préstamo y la fecha de devolución.

## PROGRAMA INGENIERIA DE SISTEMAS

Página 11 | 18

### 2. Todos los libros que aún no han sido devueltos de un usuario específico.

```
db.prestamos.aggregate([
  { $match: { usuario_id: 1,
    fecha_devolucion: { $exists: false } } },
  { $lookup: {
    from: "libros",
    localField: "libro_id",
    foreignField: "_id",
    as: "detalle_libro"
  } },
  { $unwind: "$detalle_libro" },
  { $project: {
    _id: 1,
    titulo: "$detalle_libro.titulo",
    fecha_prestamo: 1
  } }
]);
```

```
{
  _id: 2,
  fecha_prestamo: 2024-02-05T00:00:00.000Z,
  titulo: 'Matar a un Ruiseñor'
}
{
  _id: 3,
  fecha_prestamo: 2024-03-10T00:00:00.000Z,
  titulo: 'Los Pilares de la Tierra'
}
```

### 3. Todos los libros que han sido devueltos de un usuario específico.

```
db.prestamos.aggregate([
  { $match: { usuario_id: 1,
    fecha_devolucion: { $exists: true } } },
  { $lookup: {
    from: "libros",
    localField: "libro_id",
    foreignField: "_id",
    as: "detalle_libro"
  } },
  { $unwind: "$detalle_libro" },
  { $project: {
    _id: 1,
    titulo: "$detalle_libro.titulo",
    fecha_prestamo: 1,
    fecha_devolucion: 1
  } }
]);
```

```
{
  _id: 1,
  fecha_prestamo: 2024-01-01T00:00:00.000Z,
  fecha_devolucion: 2024-01-10T00:00:00.000Z,
  titulo: 'Cien Años de Soledad'
}
```

- **\$match:** Ambas consultas filtran los préstamos del usuario\_id: 1. La diferencia está en el filtro aplicado a fecha\_devolucion:
- **Para los libros no devueltos:** { \$exists: false }
- **Para los libros devueltos:** { \$exists: true }
- **\$lookup, \$unwind y \$project:** Son iguales en ambas consultas, recuperando los detalles del libro y mostrando los datos relevantes del préstamo (fecha de préstamo y devolución, si aplica).

#### 4. Todos los préstamos realizados a los usuarios

```
db.prestamos.aggregate([
  { $group: {
    _id: "$usuario_id",
    total_prestamos: { $sum: 1 },
    libros: { $push: "$libro_id" }
  }},
  { $sort: { total_prestamos: -1 } },
  { $lookup: {
    from: "usuarios",
    localField: "_id",
    foreignField: "_id",
    as: "usuario_info"
  }},
  { $unwind: "$usuario_info" },
  { $lookup: {
    from: "libros",
    localField: "libros",
    foreignField: "_id",
    as: "libro_info"
  }},
  { $project: {
    _id: 0,
    nombre_usuario: "$usuario_info.nombre",
    total_prestamos: 1,
    libros: {
      $map: {
        input: "$libro_info",
        as: "libro",
        in: "$$libro.titulo"
      }
    }
  }
}]);
```

```
{
  total_prestamos: 3,
  nombre_usuario: 'Ana Gómez',
  libros: [
    '1984',
    'El Gran Gatsby',
    'El Hobbit'
  ]
}
{
  total_prestamos: 3,
  nombre_usuario: 'Sofía Castro',
  libros: [
    'Cien Años de Soledad',
    'Crimen y Castigo',
    'El Nombre de la Rosa'
  ]
}
{
  total_prestamos: 3,
  nombre_usuario: 'Luis Rodríguez',
  libros: [
    'Orgullo y Prejuicio',
    'Crimen y Castigo',
    'El Nombre de la Rosa'
  ]
}
```

- **\$group:** Agrupa los préstamos por usuario\_id, contando el total de préstamos y acumulando los IDs de los libros.
- **\$sort:** Ordena los usuarios por el número total de préstamos, de mayor a menor.
- **\$lookup y \$unwind (primer uso):** Realiza una "join" con la colección usuarios para obtener el nombre del usuario.
- **\$lookup (segundo uso):** Une la información de los libros asociados a cada préstamo, obteniendo los títulos.
- **\$project:** Muestra el nombre del usuario, el total de préstamos y los títulos de los libros.



## PROGRAMA INGENIERIA DE SISTEMAS

Página 13 | 18

### 5. Todos los préstamos realizados a los usuarios no devueltos

```
db.prestamos.aggregate([
  { $match: { fecha_devolucion: { $exists: false } } },
  { $group: {
    _id: "$usuario_id",
    total_prestamos: { $sum: 1 },
    libros: { $push: "$libro_id" }
  } },
  { $sort: { total_prestamos: -1 } },
  { $lookup: {
    from: "usuarios",
    localField: "_id",
    foreignField: "_id",
    as: "usuario_info"
  } },
  { $unwind: "$usuario_info" },
  { $lookup: {
    from: "libros",
    localField: "libros",
    foreignField: "_id",
    as: "libro_info"
  } },
  { $project: {
    _id: 0,
    nombre_usuario: "$usuario_info.nombre",
    total_prestamos: 1,
    libros: {
      $map: {
        input: "$libro_info",
        as: "libro",
        in: "$$libro.titulo"
      }
    }
  } }
]);
```

```
{
  total_prestamos: 2,
  nombre_usuario: 'Juan Pérez',
  libros: [
    'Matar a un Ruiseñor',
    'Los Pilares de la Tierra'
  ]
}
{
  total_prestamos: 1,
  nombre_usuario: 'Ana Gómez',
  libros: [
    'El Hobbit'
  ]
}
{
  total_prestamos: 1,
  nombre_usuario: 'Ana Gómez',
  libros: [
    'El Hobbit'
  ]
}
```

- **\$match:** Filtra los préstamos que no han sido devueltos (sin fecha\_devolucion).
- Las demás etapas son iguales a las de la consulta anterior.

El Saber como Arma de Vida



## 6. Historial de préstamos de un libro específico

```
db.prestamos.aggregate([
  { $match: { libro_id: 4 } },
  { $lookup: {
    from: "usuarios",
    localField: "usuario_id",
    foreignField: "_id",
    as: "usuario_info"
  }},
  { $unwind: "$usuario_info" },
  { $project: {
    _id: 1,
    usuario_nombre: "$usuario_info.nombre",
    fecha_prestamo: 1,
    fecha_devolucion: 1
  }}
]);
```

```
{
  _id: 2,
  fecha_prestamo: 2024-02-05T00:00:00.000Z,
  usuario_nombre: 'Juan Pérez'
}
{
  _id: 15,
  fecha_prestamo: 2024-07-15T00:00:00.000Z,
  fecha_devolucion: 2024-07-25T00:00:00.000Z,
  usuario_nombre: 'Carlos García'
}
{
  _id: 22,
  fecha_prestamo: 2024-11-15T00:00:00.000Z,
  usuario_nombre: 'Isabel Díaz'
}
```

- **\$match:** Filtra los préstamos relacionados con el libro con libro\_id: 4.
- **\$lookup y \$unwind:** Obtiene y descompone la información del usuario que solicitó el libro.
- **\$project:** Muestra el nombre del usuario, la fecha del préstamo y la fecha de devolución si está disponible.

El Saber como Arma de Vida



INSTITUTO TECNOLÓGICO DEL PUTUMAYO



IES Vigilada por:  
**Educación**

## PROGRAMA INGENIERIA DE SISTEMAS

Página 15 | 18

### Diseño de Base de Datos

El diseño de la base de datos sigue un modelo NoSQL flexible, donde las colecciones almacenan documentos en formato JSON. Las relaciones entre colecciones, como la relación entre Préstamos y Libros o Usuarios, son gestionadas a través de referencias.



El **Saber** como **Arma** de **Vida**





INSTITUTO TECNOLÓGICO DEL PUTUMAYO



IES Vigilada por:  
**Educación**

## PROGRAMA INGENIERIA DE SISTEMAS

Página 16 | 18

### Análisis y Discusión

Los resultados obtenidos de las consultas permiten obtener información detallada sobre los libros disponibles y los préstamos realizados. La estructura flexible de MongoDB facilita la inserción y recuperación de datos, y permite una escalabilidad que se adapta a las necesidades de una biblioteca en crecimiento.



El **Saber** como **Arma** de **Vida**



## PROGRAMA INGENIERIA DE SISTEMAS

Página 17 | 18

### Conclusiones

El uso de MongoDB para gestionar la información de una biblioteca demostró ser eficiente y flexible. Las consultas realizadas proporcionaron resultados precisos, y la estructura no relacional permitió almacenar datos sin necesidad de una estructura rígida.



El **Saber** como **Arma de Vida**





INSTITUTO TECNOLÓGICO DEL PUTUMAYO



IES Vigilada por:  
**Educación**

## PROGRAMA INGENIERIA DE SISTEMAS

Página 18 | 18

### Referencias

MongoDB, Inc. (2024). MongoDB Documentation. Retrieved from

<https://docs.mongodb.com/>.

Git hub: <https://github.com/esteban2oo1/mongo-EM.git>



El **Saber** como **Arma** de **Vida**

