



### Instalar

- MongoDB: [Download MongoDB Community Server | MongoDB](#)
  - o Configurar y probar MongoDB Compass
- MongoDB Shell: [MongoDB Shell Download | MongoDB](#)
  - o Montar autenticación en mongosh
- (opc) MongoDB Atlas (cloud):  
<https://www.mongodb.com/es/cloud/atlas/register>
- MongoDB Tools:  
<https://www.mongodb.com/try/download/database-tools?tck=docs databasetools>

MongoDB utiliza BSON (Binary JSON) internamente para almacenar documentos y realizar operaciones en la base de datos.

Aunque MongoDB utiliza BSON internamente, los desarrolladores generalmente interactúan con la base de datos utilizando JSON. Los controladores de MongoDB se encargan de convertir entre JSON y BSON automáticamente.

[Trabajando con JSON - Aprende desarrollo web | MDN](#)

[MongoDB Tutorial](#)

[MongoDB CRUD Operations](#)

<https://jsonbeautify.com/>  
<https://jsonformatter.org/json-viewer>



Podemos crear una BD desde cero o importando un JSON:

### Importando un JSON

- **Para importar, moverse a la ruta donde esta mongoimport**

```
cd C:\Users\usuario\Documents\mongodb-database-tools-  
windows-x86_64-100.11.0\mongodb-database-tools-  
windows-x86_64-100.11.0
```

- **Ejecutar el siguiente comando**

```
mongoimport --db=monumentosDB --collection=nombre_coleccion  
--file=ubicacion_archivo_json --jsonArray
```

Ejemplo con monumentos.json:

```
mongoimport --db=monumentosDB --collection=monumento --file=  
C:\Users\Javier\Downloads\monumentos.json --jsonArray
```

---

### Desde cero

- **Crear la BD**

```
use blog
```

- **Crear la Colección (grupo de documentos que se almacenan juntos dentro de una base de datos)**

```
db.createCollection("posts")
```

### **ABRIR TERMINAL (CMD)**

```
mongosh
```

### **Mostrar Bases de Datos**

```
show databases
```

### **Usar una Database**

```
use nombre_database
```

### **Mostrar colecciones**

```
show collections
```

### **Create a Database**

```
use blog
```

### **Create Collection**

```
db.createCollection("posts")
```

### Insert

```
db.posts.insertOne({
  title: "Post Title 1",
  body: "Body of post.",
  category: "News",
  likes: 1,
  tags: ["news", "events"],
  date: Date()
})

db.posts.insertMany([
  {
    title: "Post Title 2",
    body: "Body of post.",
    category: "Event",
    likes: 2,
    tags: ["news", "events"],
    date: Date()
  },
  {
    title: "Post Title 3",
    body: "Body of post.",
    category: "Technology",
    likes: 3,
    tags: ["news", "events"],
    date: Date()
  },
  {
    title: "Post Title 4",
    body: "Body of post.",
    category: "Event",
    likes: 4,
    tags: ["news", "events"],
    date: Date()
  }
])
```

### Find Data

```
db.posts.find()
db.posts.findOne()
db.posts.find( {category: "News"} )
db.posts.find({}, {title: 1, date: 1})
db.posts.find({}, {_id: 0, title: 1, date: 1})
db.posts.find({}, {category: 0})
```

### Update Document

```
db.posts.find( { title: "Post Title 1" } )
db.posts.updateOne( { title: "Post Title 1" }, { $set: {
likes: 2 } } )
```

Insert if not found (upsert)

```
db.posts.updateOne(
  { title: "Post Title 5" },
  {
    $set:
      {
        title: "Post Title 5",
        body: "Body of post.",
        category: "Event",
        likes: 5,
        tags: ["news", "events"],
        date: Date()
      }
  },
  { upsert: true }
)

db.posts.updateMany({}, { $inc: { likes: 1 } })
```

### Delete Documents

```
db.posts.deleteOne({ title: "Post Title 5" })
db.posts.deleteMany({ category: "Technology" })
```

### **MongoDB Aggregation**

- \$group
- \$limit
- \$project
- \$sort
- \$match
- \$addFields
- \$count
- \$lookup
- \$out

### **MongoDB Charts**

(investigar)

---