

Configuración de MongoDB en Proyecto Node.js

1. Configuración en MongoDB Atlas (Cloud)

Crear cuenta y cluster

1. Ir a [MongoDB Atlas](#)
2. Crear cuenta gratuita
3. Crear nuevo proyecto
4. Crear cluster gratuito (M0)

Configurar acceso

1. Database Access:

- Add New Database User
- Username: (o el que prefieras)
- Password: generar automáticamente o crear una
- Database User Privileges:

2. Network Access:

- Add IP Address
- Allow access from anywhere:
- (Para desarrollo - en producción usar IPs específicas)

Obtener Connection String

1. En el cluster, click "Connect"
2. Choose "Connect your application"
3. Copy connection string:

```
mongodb+srv://admin:<password>@cluster.mongodb.net/?retryWrites=true&w=majority
```

2. Configuración en el Proyecto Node.js

Instalar dependencias

```
bash
```

```
# Mongoose para conectar con MongoDB
```

```
npm install mongoose
```

```
# dotenv para variables de entorno
```

```
npm install dotenv
```

Crear archivo .env

```
env
```

```
MONGODB_URI=mongodb+srv://admin:tu_password@cluster.mongodb.net/nombre_base_datos?retryWrites=true&
```



Importante:

- Reemplazar `tu_password` con la password real
- Reemplazar `nombre_base_datos` con el nombre deseado (ej: `phoneBook_db`)
- Reemplazar `cluster.mongodb.net` con tu cluster real

Configurar .gitignore

```
gitignore
```

```
node_modules/
```

```
.env
```

Archivo de conexión básica (mongo.js)

javascript

```
// Cargar variables de entorno
require('dotenv').config()
const mongoose = require('mongoose')

// Obtener URI desde variables de entorno
const url = process.env.MONGODB_URI

// Configurar conexión
mongoose.set('strictQuery', false)
mongoose.connect(url)

// Definir schema (estructura de datos)
const contactSchema = new mongoose.Schema({
  name: String,
  number: String
})

// Crear modelo
const Contact = mongoose.model('Contact', contactSchema)

// Crear documento de prueba
const contact = new Contact({
  name: 'Miguel',
  number: '123-456-789'
})

// Guardar y cerrar conexión
contact.save().then((result) => {
  console.log('contact saved successfully')
  mongoose.connection.close()
})
```

3. Estructura de Archivos Final

```
proyecto/
├── .env          # Variables de entorno (NO subir a Git)
├── .gitignore    # Incluye .env y node_modules/
├── mongo.js      # Script de prueba de conexión
├── package.json  # Dependencias del proyecto
└── node_modules/ # Librerías instaladas
```

4. Prueba de Conexión

bash

Ejecutar script de prueba

node mongo.js

Resultado esperado:

contact saved successfully

5. Conceptos Importantes

Jerarquía MongoDB

- **Base de datos:** Contenedor principal (ej: `phoneBook_db`)
- **Colección:** Grupo de documentos similares (ej: `contacts`)
- **Documento:** Registro individual en formato JSON

Convenciones Mongoose

- Modelo singular (`Contact`) → Colección plural (`contacts`)
- Si no especificas base de datos → usa `test` por defecto

Variables de Entorno

- **Desarrollo:** Usar archivo `.env`
- **Producción:** Configurar en plataforma de deployment (Render, Heroku, etc.)
- **Nunca** subir credenciales a Git

6. Siguiendo Pasos

1. Migrar API de array en memoria a MongoDB
2. Implementar operaciones CRUD con Mongoose
3. Agregar validaciones y manejo de errores
4. Configurar variables de entorno para producción

Notas de Seguridad:

- Cambiar passwords antes de deployment
- Usar IPs específicas en producción
- Nunca hardcodear credenciales en el código