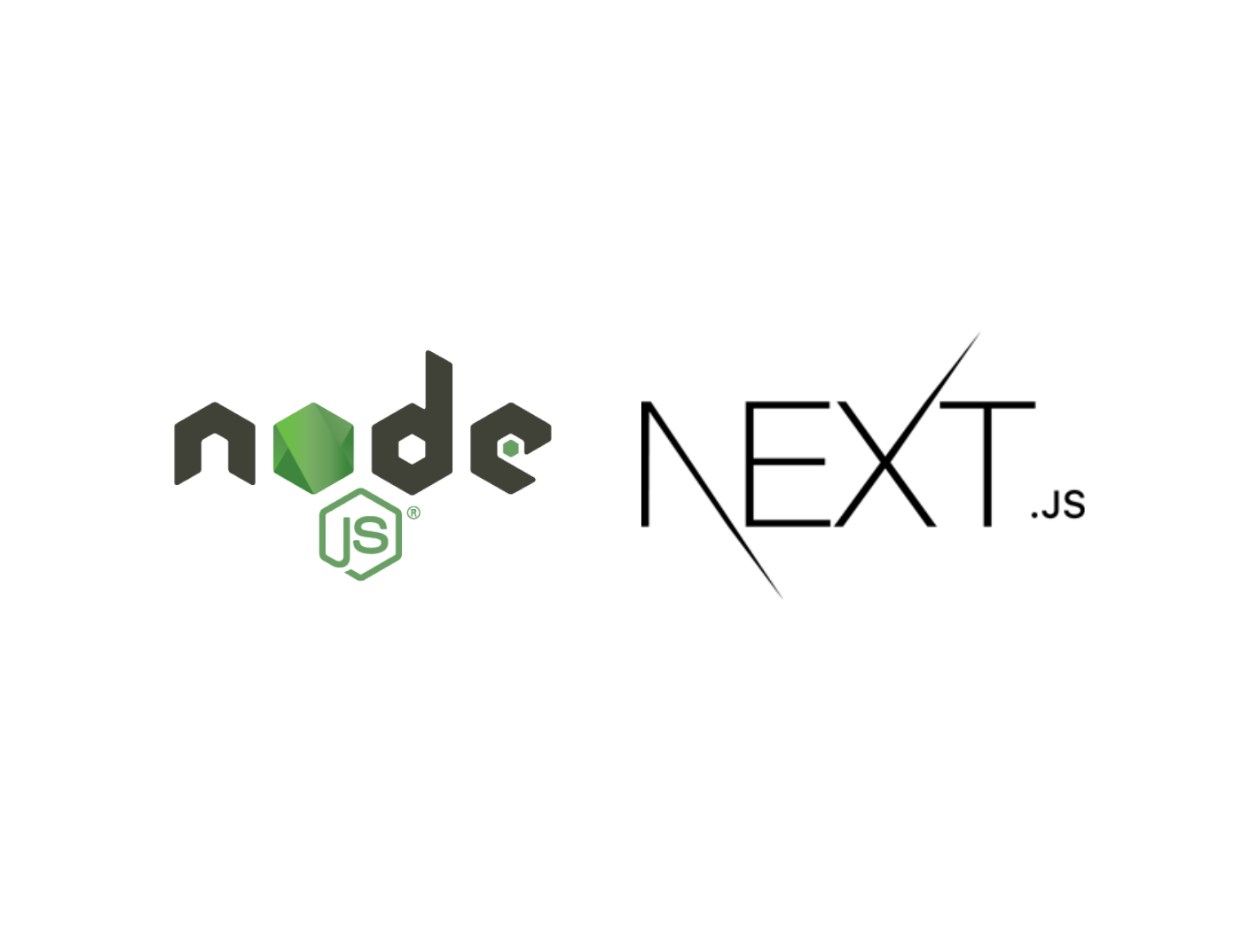
Logotipo, nombre de la empresa

Descripción generada automáticamente

**DESARROLLO DE APLICACIONES WEB AVANZADO**

**LABORATORIO N° 6**

**INTRODUCCIÓN A LA BASE DE DATOS NO SQL.**



|  |  |  |
| --- | --- | --- |
| **Alumnos** | | **Nota** |
|  | |  |
| **Grupo** |  | |
| **Fecha de Entrega** |  | |
| **Docente** | **Renato Usnayo Cáceres** | |

**OBJETIVOS:**

Identifica los requerimientos para la conexión entre Node.js y MongoDB

**SEGURIDAD:**

|  |  |
| --- | --- |
|  | **Advertencia:**  **En este laboratorio está prohibida la manipulación del hardware, conexiones eléctricas o de red; así como la ingestión de alimentos o bebidas.** |

**FUNDAMENTO TEÓRICO:**

* Revisar el texto guía que está en el campus Virtual.

**NORMAS EMPLEADAS:**

* No aplica

**RECURSOS:**

* En este laboratorio cada alumno trabajará con un equipo con Windows 11.

**METODOLOGÍA PARA EL DESARROLLO DE LA TAREA:**

* El desarrollo del laboratorio es individual

**PROCEDIMIENTO:**

**Nota:**

|  |
| --- |
| *Las secciones en azul y cursiva brindan una explicación teórica o del código* |

**Procedimiento:**

**Vista General de los Drivers de MongoDB para Node.js**

MongoDB ofrece varios drivers para conectarse desde Node.js:

Principales Drivers de MongoDB

|  |  |
| --- | --- |
| Driver | Descripción |
| MongoDB Native Driver | Driver oficial de MongoDB para Node.js. |
| Mongoose | ODM (Object Data Modeling) que facilita el manejo de datos con modelos y validaciones. |
| TypeORM (con MongoDB) | ORM basado en TypeScript, compatible con MongoDB. |

Diferencias clave:

* MongoDB Native Driver: Más flexible y directo, pero requiere escribir más código.
* Mongoose: Proporciona validaciones y esquemas para modelar datos fácilmente.
* TypeORM: Recomendado para aplicaciones con TypeScript.

Instalación del Driver de MongoDB en Node.js

Crear un nuevo proyecto de Node.js

|  |
| --- |
| mkdir mongo-node-lab  cd mongo-node-lab  npm init -y |

Instalar el MongoDB Native Driver

|  |
| --- |
| npm install mongodb dotenv |

*mongodb: Driver oficial para interactuar con MongoDB.*

*dotenv: Para manejar variables de entorno con credenciales.*

**Configurar la Conexión a MongoDB**

Crear un archivo .env para las credenciales

Crea un archivo .env en la raíz del proyecto y agrega:

|  |
| --- |
| MONGO\_URI=mongodb://localhost:27017/miBaseDeDatos |

Crear un archivo database.js para gestionar la conexión

|  |
| --- |
| require("dotenv").config();  const { MongoClient } = require("mongodb");  const uri = process.env.MONGO\_URI;  const client = new MongoClient(uri);  async function connectDB() {  try {  await client.connect();  console.log("✅ Conectado a MongoDB");  return client.db();  } catch (error) {  console.error("❌ Error al conectar a MongoDB:", error);  }  }  module.exports = { connectDB, client }; |

Probar la conexión en index.js

|  |
| --- |
| const { connectDB } = require("./database");  async function main() {  const db = await connectDB();  if (db) {  console.log("🎯 Base de datos lista para usar.");  }  }  main(); |

Ejecutar el script

|  |
| --- |
| node index.js |

Si la conexión es exitosa, verás:

|  |
| --- |
|  |

**Crear y Manipular Datos en MongoDB**

Crear una colección y agregar datos

Modifica index.js para insertar datos en una colección productos:

|  |
| --- |
| async function insertProducto() {  const db = await connectDB();  const productos = db.collection("productos");  const nuevoProducto = { nombre: "Laptop", precio: 1200, stock: 10 };  const resultado = await productos.insertOne(nuevoProducto);  console.log(" Producto insertado con ID:", resultado.insertedId);  }  insertProducto(); |

Leer datos de MongoDB

|  |
| --- |
| async function leerProductos() {  const db = await connectDB();  const productos = db.collection("productos");  const lista = await productos.find().toArray();  console.log("Lista de productos:", lista);  }  leerProductos(); |

Actualizar un producto

|  |
| --- |
| async function actualizarProducto() {  const db = await connectDB();  const productos = db.collection("productos");  const resultado = await productos.updateOne(  { nombre: "Laptop" },  { $set: { precio: 1100 } }  );  console.log("Documentos actualizados:", resultado.modifiedCount);  }  actualizarProducto(); |

Eliminar un producto

|  |
| --- |
| async function eliminarProducto() {  const db = await connectDB();  const productos = db.collection("productos");  const resultado = await productos.deleteOne({ nombre: "Laptop" });  console.log("Producto eliminado:", resultado.deletedCount);  }  eliminarProducto(); |

**Alternativa con Mongoose**

Instalar Mongoose

|  |
| --- |
| npm install mongoose |

Crear un modelo de Producto con Mongoose

|  |
| --- |
| const mongoose = require("mongoose");  mongoose.connect(process.env.MONGO\_URI)  .then(() => console.log("Conectado a MongoDB con Mongoose"))  .catch(error => console.error(" Error:", error));  const ProductoSchema = new mongoose.Schema({  nombre: String,  precio: Number,  stock: Number  });  const Producto = mongoose.model("Producto", ProductoSchema);  // Insertar un producto  async function crearProducto() {  const producto = new Producto({ nombre: "Mouse", precio: 50, stock: 20 });  await producto.save();  console.log("Producto guardado:", producto);  }  crearProducto(); |

Captura del resultado

|  |
| --- |
|  |

**TAREA**

**Consultar datos con filtros y proyección**

Objetivo: Aprender a filtrar y proyectar datos en consultas MongoDB.

Instrucciones:

* Implementa una consulta que busque clientes mayores de 30 años.
* Muestra solo los campos nombre y correo usando proyección.
* Prueba la consulta en código y en MongoDB Compass.

**Actualizar datos en MongoDB**

Objetivo: Practicar la actualización de documentos en una colección.

Instrucciones:

* Modifica la ciudad de un cliente en la colección clientes.
* Usa updateOne() para cambiar la ciudad de un cliente específico.
* Usa updateMany() para cambiar la ciudad de todos los clientes que viven en "Bogotá".

**Eliminar documentos de la base de datos**

Objetivo: Aprender a eliminar documentos en MongoDB de manera segura.

Instrucciones:

* Implementa una función para eliminar un cliente por su nombre usando deleteOne().
* Implementa otra función para eliminar todos los clientes de una ciudad con deleteMany().
* Confirma que los datos fueron eliminados correctamente.

***Colocar capturas del código y capturas de ejecución***

***Grabar explicando el código y ejecución de este en máximo 4 minutos***

**OBSERVACIONES:** *(Las observaciones son las notas aclaratorias, objeciones y problemas que se pudo presentar en el desarrollo del laboratorio)*

* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CONCLUSIONES:** *(Las conclusiones son una opinión sobre tu trabajo, explicar cómo resolviste las dudas o problemas presentados en el laboratorio. Además de aportar una opinión critica de lo realizado)*

* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_