



Bootcamp de Desarrollo Web

**Caso
Práctico 4
Integración JS - MySQL**

1



Objetivo

- Practicar la conexión a una base de datos MySQL desde una aplicación JavaScript utilizando Node.js.
- Aprender a realizar consultas SQL básicas y manejar los resultados en una aplicación Node.js.
- Entender los conceptos básicos de la interacción entre Node.js y una base de datos relacional como MySQL.

2



Ejercicio: Conexión a una base de datos MySQL con Node.js

1- Configuración del entorno:

Asegúrate de tener Node.js instalado en tu sistema. Puedes descargarlo e instalarlo desde el sitio web oficial de Node.js. Instala el módulo mysql de Node.js para interactuar con la base de datos MySQL. Puedes instalarlo ejecutando `npm install mysql` en tu terminal.



Ejercicio: Conexión a una base de datos MySQL con Node.js

2 - Creación de una base de datos MySQL:

Utiliza una herramienta como phpMyAdmin o MySQL Workbench para crear una base de datos en MySQL, o bien, puedes crearla directamente desde la línea de comandos de MySQL.

Por ejemplo, puedes crear una base de datos llamada `ejemplo_db` con una tabla llamada `usuarios` que tenga columnas como `id`, `nombre`, `email`, etc.



Ejercicio: Conexión a una base de datos MySQL con Node.js

3 - Crear un archivo JavaScript:

Crea un nuevo archivo JavaScript llamado `conexion_db.js`.

4 - Conectar a la base de datos desde Node.js:

En el archivo `conexion_db.js`, importa el módulo `mysql`.

Crea una conexión a la base de datos utilizando las credenciales de acceso y la información de conexión adecuadas.

Por ejemplo:

Introducción a la programación

5



Ejercicio: Conexión a una base de datos MySQL con Node.js

```
const mysql = require('mysql');

const conexion = mysql.createConnection({
  host: 'localhost',
  user: 'tu_usuario_mysql',
  password: 'tu_contraseña_mysql',
  database: 'ejemplo_db'
});

conexion.connect((error) => {
  if (error) {
    console.error('Error al conectar a la base de datos: ', error);
    return;
  }
  console.log('Conexión a la base de datos exitosa!');
});

// Ahora puedes realizar consultas SQL desde aqui
```

Introducción a la programación

6



Ejercicio: Conexión a una base de datos MySQL con Node.js

Realizar consultas SQL:

Después de establecer la conexión, puedes ejecutar consultas SQL utilizando el objeto `conexion`.

Por ejemplo, puedes realizar una consulta para obtener todos los usuarios de la tabla `usuarios`:



Ejercicio: Conexión a una base de datos MySQL con Node.js

```
conexion.query('SELECT * FROM usuarios', (error, resultados, campos) => {  
  if (error) {  
    console.error('Error al ejecutar la consulta: ', error);  
    return;  
  }  
  console.log('Resultados de la consulta: ', resultados);  
});
```



Ejercicio: Conexión a una base de datos MySQL con Node.js

Cerrar la conexión:

No te olvides de cerrar la conexión una vez que hayas terminado de interactuar con la base de datos. Puedes hacerlo llamando al método `end()` del objeto `conexion`.



Notas

- No olvides subir el código a tu repositorio de github
- Adjunta un archivo **pdf** con las capturas de proyecto en la plataforma
- Los archivos a usar se encuentran adjuntos en la plataforma.



Bootcamp de Desarrollo Web

**Caso
Práctico 4
Integración JS - MySQL**