



**UNIVERSIDAD NACIONAL DEL ALTIPLANO**



**FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,  
ELECTRÓNICA Y SISTEMAS**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

## **Funciones Continuas**

**CURSO:**

**ESTRUCTURA DE DATOS AVANZADAS**

**DOCENTE:**

**COLLAMQUI MARTINEZ FREDY**

**ESTUDIANTE:**

**MAMANI TURPO ELFER**

**SEMESTRE:**

**VI**

**GRUPO**

**C**

**PUNO 2024**

# Sistema para una Biblioteca

## PROBLEMA PRÁCTICO

Problema: Supongamos que estamos desarrollando un sistema para una biblioteca. Queremos implementar una funcionalidad que permita buscar libros por su ID de manera eficiente. Cada libro tiene un ID único, un título y un autor. Queremos poder insertar nuevos libros, buscar libros por su ID y eliminar libros de la biblioteca.

### 1. Configuración del Entorno y Tecnologías Utilizadas

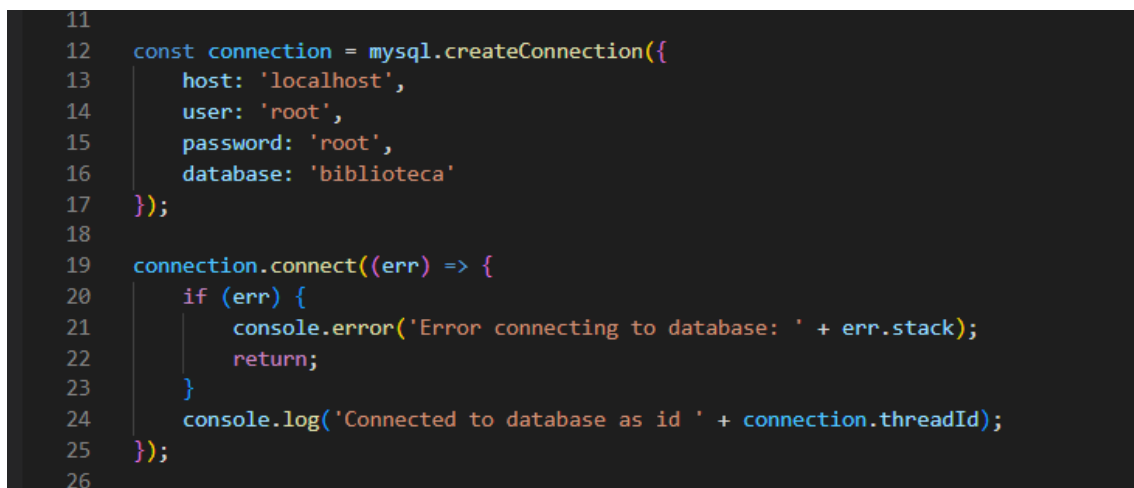
- **Node.js y Express:** Código donde se configura el servidor Node.js usando Express para manejar las rutas y las solicitudes HTTP.



The screenshot shows the VS Code interface with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'BIBLIOTECA1' with files like 'node\_modules', 'views', 'libros.ejs', 'package-lock.json', 'package.json', and 'server.js'. The code editor shows the content of 'server.js' with the following code:

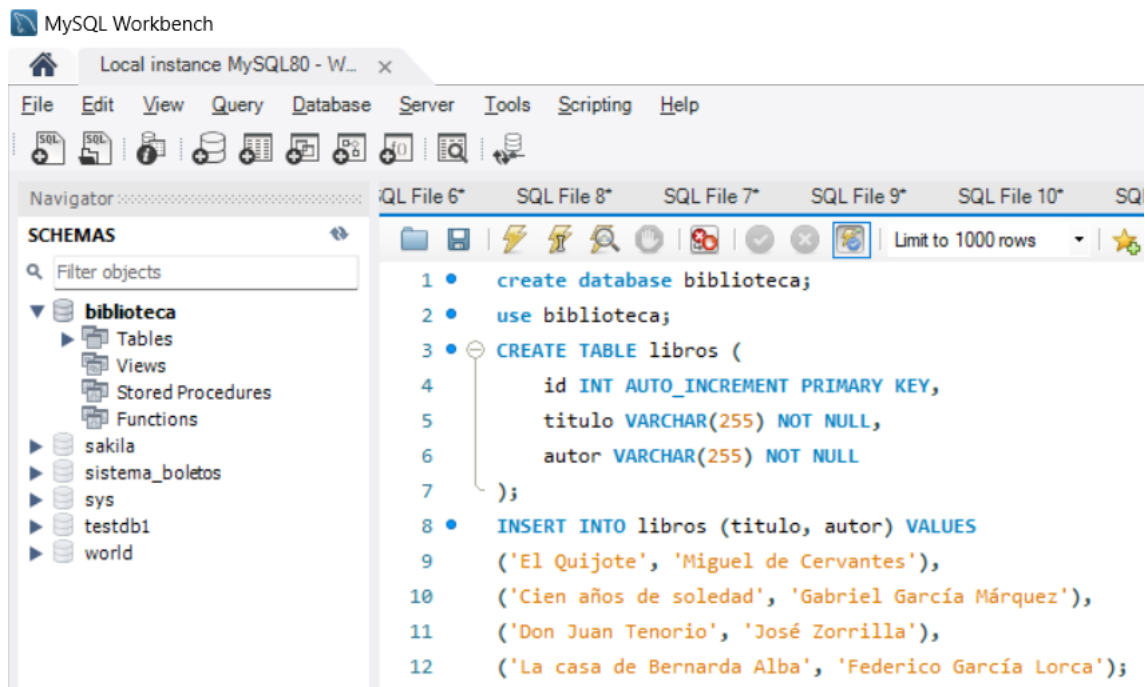
```
1 const express = require('express');
2 const mysql = require('mysql2');
3 const ejs = require('ejs');
4 const bodyParser = require('body-parser');
5
6 const app = express();
7 const port = 3006;
8
9 app.use(bodyParser.urlencoded({ extended: true }));
10 app.use(bodyParser.json());
11
```

- **MySQL:** Configuración de la conexión a la base de datos MySQL, incluyendo cómo se establece la conexión y se ejecutan consultas SQL.



The screenshot shows the continuation of the 'server.js' file in the code editor, specifically the MySQL connection setup. The code is as follows:

```
11
12 const connection = mysql.createConnection({
13   host: 'localhost',
14   user: 'root',
15   password: 'root',
16   database: 'biblioteca'
17 });
18
19 connection.connect((err) => {
20   if (err) {
21     console.error('Error connecting to database: ' + err.stack);
22     return;
23   }
24   console.log('Connected to database as id ' + connection.threadId);
25 });
26
```



- **EJS (Embedded JavaScript):** Uso de EJS como motor de plantillas para renderizar las vistas dinámicas en el servidor.

```

20
27 // Establecer el motor de vistas y la carpeta de vistas
28 app.set('view engine', 'ejs');
29 app.set('views', './views');
30

```

## 2. Implementación de la Lógica de Negocio

- **Rutas y Controladores Binary Search Tree (BST):** Las definiciones de las rutas (GET, POST, DELETE) y sus respectivos controladores en Express para manejar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de los libros.

```

// Ruta para mostrar todos los libros
app.get('/libros', (req, res) => {
  const SELECT_ALL_BOOKS_QUERY = 'SELECT * FROM libros';
  connection.query(SELECT_ALL_BOOKS_QUERY, (err, results) => {
    if (err) {
      console.error('Error fetching books: ' + err.stack);
      res.status(500).send('Error fetching books');
      return;
    }
    res.render('libros', { libros: results });
  });
});

// Ruta para buscar un libro por ID
app.get('/buscar/:id', (req, res) => {
  const id = req.params.id;
  const SELECT_BOOK_BY_ID_QUERY = 'SELECT * FROM libros WHERE id = ?';
  connection.query(SELECT_BOOK_BY_ID_QUERY, [id], (err, results) => {
    {

```

```

    if (err) {
      console.error('Error fetching book by ID: ' + err.stack);
      res.status(500).send('Error fetching book by ID');
      return;
    }
    res.json(results);
    //res.render('libros', { libros: results }); // Renderiza la
    vista 'libros.ejs' con los resultados
  });
});

// Ruta para insertar un nuevo libro
app.post('/insertar', (req, res) => {
  const { titulo, autor } = req.body;
  const INSERT_BOOK_QUERY = 'INSERT INTO libros (titulo, autor)
VALUES (?, ?)';
  connection.query(INSERT_BOOK_QUERY, [titulo, autor], (err,
results) => {
    if (err) {
      console.error('Error inserting book: ' + err.stack);
      res.status(500).send('Error inserting book');
      return;
    }
    res.send(`Libro "${titulo}" insertado correctamente.`);
  });
});

// Ruta para eliminar un libro por ID
app.delete('/eliminar/:id', (req, res) => {
  const id = req.params.id;
  const DELETE_BOOK_BY_ID_QUERY = 'DELETE FROM libros WHERE id = ?';
  connection.query(DELETE_BOOK_BY_ID_QUERY, [id], (err, results) =>
{
    if (err) {
      console.error('Error deleting book by ID: ' + err.stack);
      res.status(500).send('Error deleting book by ID');
      return;
    }
    res.send(`Libro con ID ${id} eliminado correctamente.`);
  });
});
});

```

- **Operaciones con la Base de Datos:** Los bloques de código donde se ejecutan consultas SQL para insertar, buscar y eliminar libros en la base de datos MySQL.

```

18 • INSERT INTO libros (titulo, autor) VALUES
19 ('La Odisea', 'Homero'),
20 ('La Cabaña del Tío Tom', 'Harriet Beecher Stowe'),
21 ('Frankenstein', 'Mary Shelley'),
22 ('1984', 'George Orwell, 1949'),
23 ('Todo Se Derrumba', 'Chinua Achebe'),
24 ('Las Mil y Una Noches', 'Varios autores'),
25 ('El Quijote', 'Miguel de Cervantes Saavedra'),
26 ('Hamlet', 'William Shakespeare'),
27 ('Cien Años de Soledad', 'Gabriel García Márquez'),
28 ('La Ilíada', 'Homero'),
29 ('Beloved', 'Toni Morrison'),
30 ('La Divina Comedia', 'Dante Alighieri'),
31 ('Romeo y Julieta', 'William Shakespeare'),

67 ('Edipo Rey', 'Sófocles'),
68 ('La Metamorfosis', 'Franz Kafka'),
69 ('La Orestíada', 'Esquilo siglo V a.C.'),
70 ('Cenicienta', 'Autor y fecha desconocidos'),
71 ('Aullido', 'Allen Ginsberg'),
72 ('Middlemarch', 'George Eliot'),
73 ('Toba Tek Singh', 'Saadat Hasan Manto');
74
75 • INSERT INTO libros (titulo, autor) VALUES
76 ('Panchatantra', 'Atribuido a Vishnu Sharma'),
77 ('Memorias póstumas de Blas Cubas', 'Joaquim Machado de Assis'),
78 ('La plenitud de la señorita Brodie', 'Muriel Spark'),
79 ('Los filántropos en harapos', 'Robert Tressell'),
80 ('La canción de Lawino', 'Okot p\Bitek'),

104 ('Canto a mí mismo', 'Walt Whitman'),
105 ('Las aventuras de Huckleberry Finn', 'Mark Twain'),
106 ('Las Aventuras de Tom Sawyer', 'Mark Twain'),
107 ('El Aleph', 'Jorge Luis Borges'),
108 ('Historia del Campesino Elocuente', 'Cuento tradicional egipcio'),
109 ('El traje nuevo del emperador', 'Hans Christian Andersen'),
110 ('La Jungla', 'Upton Sinclair'),
111 ('Khamriyyat', 'Abu Nuwas'),
112 ('La marcha Radetzky', 'Joseph Roth'),
113 ('El Cuervo', 'Edgar Allan Poe'),
114 ('Los versos satánicos', 'Salman Rushdie'),
115 ('El secreto', 'Donna Tartt'),
116 ('Un día de nieve', 'Ezra Jack Keats'),
117 ('Toba Tek Singh', 'Saadat Hasan Manto');

```

	id	titulo	autor
	91	Canto a mí mismo	Walt Whitman
	92	Las aventuras de Huckle...	Mark Twain
	93	Las Aventuras de Tom S...	Mark Twain
	94	El Aleph	Jorge Luis Borges
	95	Historia del Campesino El...	Cuento tradicional egi...
	96	El traje nuevo del emper...	Hans Christian Andersen
	97	La Jungla	Upton Sinclair
	98	Khamriyyat	Abu Nuwas
	99	La marcha Radetzky	Joseph Roth
	100	El Cuervo	Edgar Allan Poe
	101	Los versos satánicos	Salman Rushdie
	102	El secreto	Donna Tartt
	103	Un día de nieve	Ezra Jack Keats
	104	Toba Tek Singh	Saadat Hasan Manto
✱	NULL	NULL	NULL

### 3. Interfaz de Usuario (Front-end)

- **Vistas HTML y CSS:** Código HTML y CSS utilizados para diseñar y estilizar la interfaz de usuario, incluyendo formularios de inserción, listado de libros y botones de acción (eliminar).

```

66 <body>
67 <h1>Lista de Libros</h1>
68 <h2>Buscar Libro por ID</h2>
69 <form action="/buscar" method="GET">
70 <label>ID del Libro: </label>
71 <input type="number" name="id">
72 <button type="submit">Buscar</button>
73 </form>
74 <h2>Insertar Nuevo Libro</h2>
75 <form action="/insertar" method="POST">
76 <label>Titulo del Libro: </label>
77 <input type="text" name="titulo" required>
78 <label>Autor del Libro: </label>
79 <input type="text" name="autor" required>
80 <button type="submit">Insertar</button>
81 </form>
82 <h2>Lista de Libros</h2>
83 <ul>
84 <% libros.forEach(libro => { %>
85 <li>
86 <div>
87 <strong>ID:</strong> <%= libro.id %><br>
88 <strong>Titulo:</strong> <%= libro.titulo %><br>
89 <strong>Autor:</strong> <%= libro.autor %>
90 </div>
91 <form class="delete-form" action="/eliminar/<%= libro.id %>" method="POST">
92 <button type="submit" class="delete-button">Eliminar</button>
93 </form>
94 </li>
95 <% }); %>
96 </ul>
97 <script>
98 // JavaScript para eliminar un libro
99 const deleteForms = document.querySelectorAll('.delete-form');
100 deleteForms.forEach(form => {
101 form.addEventListener('submit', event => {
102 event.preventDefault();
103 const confirmar = confirm('¿Estás seguro de que quieres eliminar este libro?');
104 if (confirmar) {
105 const id = form.getAttribute('action').split('/')[2];
106 fetch('/eliminar/${id}', {
107 method: 'DELETE'
108 })
109 .then(response => response.text())
110 .then(message => {
111 alert(message);
112 location.reload();
113 })
114 .catch(error => console.error('Error:', error));
115 }
116 });
117 });
118 </script>

```

## 4. Integración y Funcionalidad Completa

- **Capturas de Pantalla Funcionales:** Capturas de pantalla que muestren la aplicación en acción, como la página principal con la lista de libros, el formulario de inserción de libros y la confirmación de eliminación.

Lista de Libros

Buscar Libro por ID

ID del Libro:

Eliminar

Insertar Nuevo Libro

Título del Libro:

Autor del Libro:

Insertar

Lista de Libros

ID: 1  
Título: El Quijote  
Autor: Miguel de Cervantes

ID: 2  
Título: Cien años de soledad  
Autor: Gabriel García Márquez

ID: 3  
Título: Don Juan Tenorio  
Autor: José Zorrilla

ID: 4  
Título: La casa de Bernarda Alba  
Autor: Federico García Lorca

ID: 5  
Título: La Odisea  
Autor: Homero

ID: 7

Búsqueda:

localhost:3006/buscar/100

Gmail YouTube Traducir Aula Virtual Vikiviki.comhttps://... ChatGPT Perple

Lista de Libros

Buscar Libro por ID

ID del Libro:

Eliminar

Insertar Nuevo Libro

Título del Libro:

Autor del Libro:

Insertar

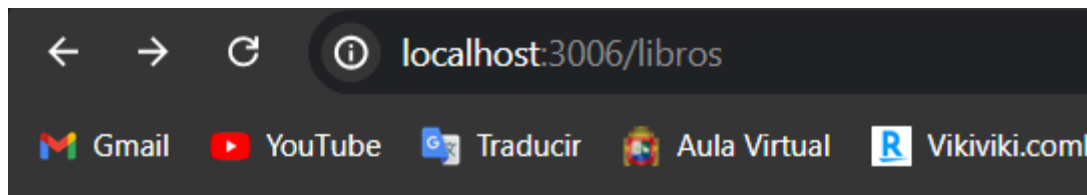
Lista de Libros

ID: 100  
Título: El Cuervo  
Autor: Edgar Allan Poe

98	Khamriyyat	Abu Nuwas
99	La marcha Radetzky	Joseph Roth
100	El Cuervo	Edgar Allan Poe
101	Los versos satánicos	Salman Rushdie



Insertar:



## Lista de Libros

### Buscar Libro por ID

ID del Libro:

Buscar

### Insertar Nuevo Libro

Título del Libro:

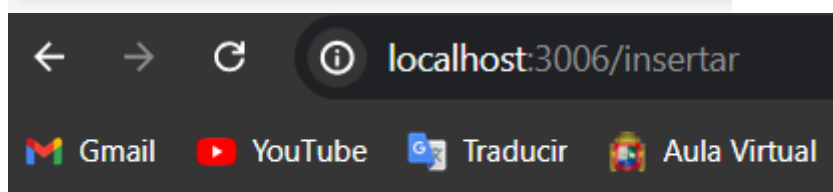
Autor del Libro:

Insertar

**ID:** 105

**Título:** Domingo

**Autor:** Solorio



Libro "Pedro Páramo" insertado correctamente.

**ID:** 105

**Título:** Domingo

**Autor:** Solorio

**ID:** 106

**Título:** Pedro Páramo

**Autor:** Juan Rulfo

119

120 • `SELECT * FROM libros WHERE id IN (105, 106);`

121

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	id	titulo	autor
▶	105	Domingo	Solorio
	106	Pedro Páramo	Juan Rulfo
*	NULL	NULL	NULL

Eliminación: Id 105

localhost:3006/buscar/105

Lista de Libros

Buscar Libro por ID

ID del Libro:

Buscar

Insertar Nuevo Libro

Título del Libro:

Autor del Libro:

Insertar

Lista de Libros

ID: 105

Título: Domingo

Autor: Solorio

Eliminar

localhost:3006 dice

¿Estás seguro de que quieres eliminar este libro?

Aceptar

Cancelar

localhost:3006/buscar/105

Lista de Libros

Buscar Libro por ID

ID del Libro:

Buscar

Insertar Nuevo Libro

Título del Libro:

Autor del Libro:

Insertar

Lista de Libros

ID: 105

Título: Domingo

Autor: Solorio

Eliminar

localhost:3006 dice

Libro con ID 105 eliminado correctamente.

Aceptar

120 • `SELECT * FROM libros WHERE id IN (105, 106);`

121 • `SELECT * FROM libros WHERE id IN (104,105,106);`

122

Result Grid

Filter Rows:

Edit:

Exp

	id	titulo	autor
▶	104	Toba Tek Singh	Saadat Hasan Manto
	106	Pedro Páramo	Juan Rulfo
*	NULL	NULL	NULL

## **Conclusión:**

Implementar esta aplicación de gestión de libros ha sido una experiencia enriquecedora y, a la vez, un reto fascinante. Desde configurar el servidor en Node.js hasta manejar las rutas y la conexión con la base de datos MySQL, cada paso ha sido un aprendizaje continuo. Ver cómo cada línea de código cobra vida y permite insertar, buscar y eliminar libros ha sido muy gratificante.

Aunque hubo momentos de frustración, especialmente cuando surgían errores inesperados, superar esos obstáculos me hizo apreciar aún más el proceso de desarrollo. La integración de EJS para renderizar las vistas y la implementación de las solicitudes HTTP correctas me han ayudado a comprender mejor cómo funcionan las aplicaciones web en su totalidad.

Este proyecto no solo ha mejorado mis habilidades técnicas, sino que también ha reforzado mi paciencia y perseverancia. Es increíble ver cómo un concepto puede transformarse en una aplicación funcional, y estoy emocionado por las posibilidades futuras y las mejoras que puedo seguir implementando.