

INGENIERÍA WEB II

ACTIVIDAD 6

DOCENTE:

JOSE HEMBER SOLORZANO PARDO

NRC: 50-55290

S

ERIKA MILENA MARIN VILLAMIZAR - ID: 835814

UNIVERSIDAD MINUTO DE DIOS

FACULTAD INGENIERIA DE SISTEMAS

BOGOTA D.C

2023

Implementación de API REST en Node.js

Se realiza la implementación de API Rest en Node.js usando los métodos get, post, put y delete, para la construcción de una página web.

- Método Get

```
index.get('/clientes', (req, res) => {
  const sql = 'SELECT * FROM clientes';

  connection.query(sql, (error, results) => {
    if (error) throw error;
    if (results.length > 0) {
      res.json(results);
    } else {
      res.send('No hay resultado');
    }
  });
});

index.get('/clientes/:id', (req, res) => {
  const { id } = req.params;
  const sql = 'SELECT * FROM clientes WHERE id = ${id}';
  connection.query(sql, (error, results) => {
    if (error) throw error;
    if (results.length > 0) {
      res.json(results);
    } else {
      res.send('No hay resultado');
    }
  });
});
});
```

- Método Post

```
index.post('/agregar', (req, res) => {
  const sql = 'INSERT INTO clientes SET ?';

  const clientesObj = {
    nombre: req.body.nombre,
    apellido: req.body.apellido,
    email: req.body.email
  }

  connection.query(sql, clientesObj, error => {
    if (error) throw error;
    res.send('Cliente creado')
  });
});
```

- Método put

```
index.put('/actualizar/:id', (req, res) => {
  const {id} = req.params;
  const {nombre, apellido, email} = req.body;
  const sql = `UPDATE clientes SET nombre = '${nombre}', apellido='${apellido}', email='${email}' WHERE id = ${id}`;

  connection.query(sql, error => {
    if (error) throw error;
    res.send('Cliente actualizado');
  });
});
```

- Método delete

```
index.delete('/eliminar/:id', (req, res) => {
  const {id} = req.params;
  const sql = `DELETE FROM clientes WHERE id = ${id}`;

  connection.query(sql, error => {
    if (error) throw error;
    res.send('Cliente eliminado');
  });
});
```

- Rutas

```
//RUTA

index.get('/', (req, res) => {
    res.sendFile(path.resolve(__dirname, 'src', 'index.html'))
});

index.get('/login.html', (req, res) => {
    res.sendFile(path.resolve(__dirname, 'src', 'login.html'))
});

index.get('/registrar.html', (req, res) => {
    res.sendFile(path.resolve(__dirname, 'src', 'registrar.html'))
});

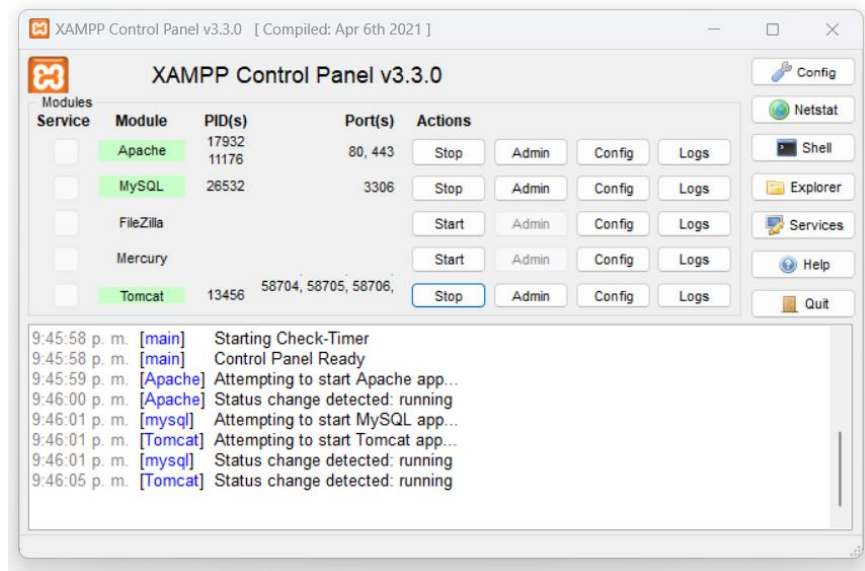
index.get('/registrarElemento.html', (req, res) => {
    res.sendFile(path.resolve(__dirname, 'src', 'registrarElemento.html'))
});

index.get('/listaElemento.html', (req, res) => {
    res.sendFile(path.resolve(__dirname, 'src', 'listaElemento.html'))
});

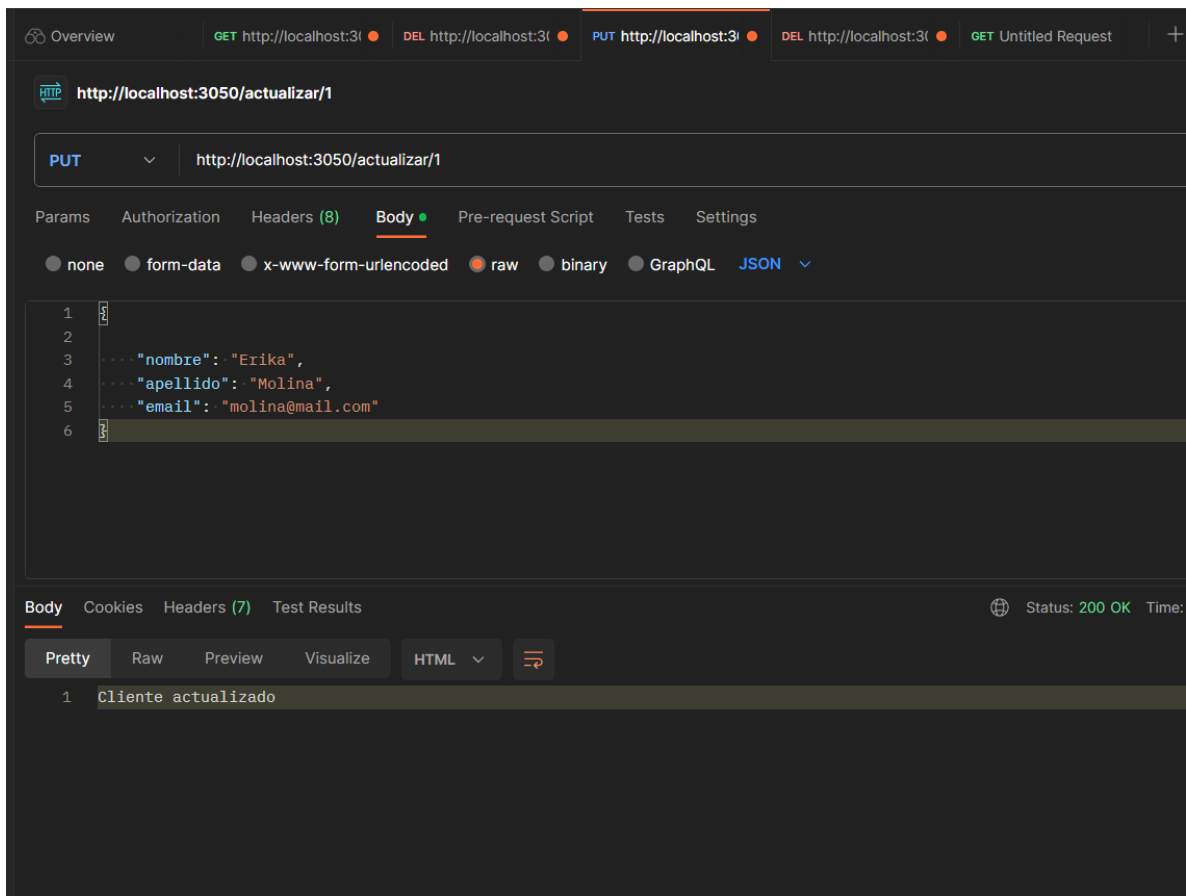
index.get('/prueba.html', (req, res) => {
    res.sendFile(path.resolve(__dirname, 'src', 'prueba.html'))
});

index.get('/editarCliente.html', (req, res) => {
    res.sendFile(path.resolve(__dirname, 'src', 'editarCliente.html'))
});
```

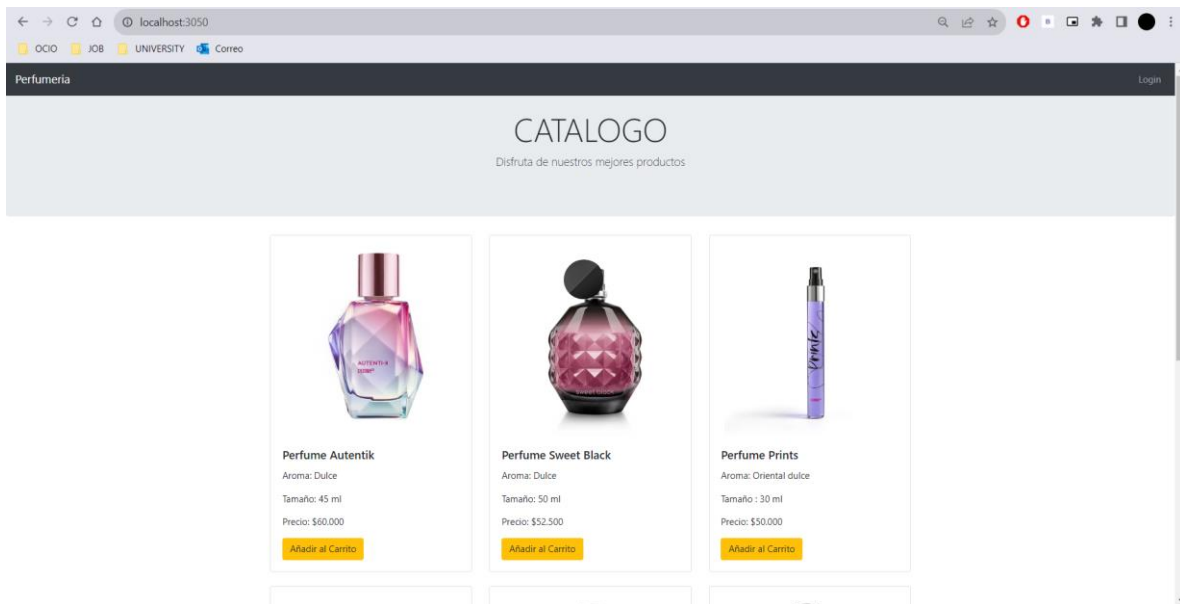
Motor de gestión de base de datos utilizado para la implementación de la página web a nivel local.



Se utiliza Postman para realizar validaciones



Se observa el menú principal de la página web donde se puede observar el catalogo de la tienda local.



```
26     </ul>
27   </div>
28 </nav>
29
30 <header class="jumbotron text-center mt-3 fade-in">
31   <h1 class="display-4">CATALOGO</h1>
32   <p class="lead">Disfruta de nuestros mejores productos</p>
33 </header>
34
35
36
37 <div class="container slide-in">
38   <div class="row">
39     <div class="col-md-4">
40       <div class="card mb-4">
41         
43           <h5 class="card-title">Perfume Autentik</h5>
44           <p class="card-text">
45             Aroma: Dulce
46           </p>
47           <p class="card-text">
48             Tamaño: 45 ml
49           </p>
50           <p class="card-text">
51             Precio: $60.000
52           </p>
53           <button class="btn btn-warning" onclick="agregarP1()">Añadir al Carrito</button>
54         </div>
55       </div>
56     </div>
57     <div class="col-md-4">
58       <div class="card mb-4">
59         
61           <h5 class="card-title">Perfume Sweet Black</h5>
62           <p class="card-text">
```

Pestaña donde se realiza el registro de usuario para miembros del equipo y consultas sobre usuarios.



PERFUMERIA

Esta seccion de la pagina web esta diseñada para el registro del inventario de la perfumeria



Todos los derechos reservados

```
41      Esta seccion de la pagina web esta diseñada para el registro del inventario de la perfumeria
42
43      </p>
44      <!--CARDS-->
45
46
47      <div class="row row-cols-1 row-cols-md-4 g-4">
48
49          <div class=""></div>
50
51          <div class="col">
52              <div class="card h-100">
53                  
54                  <div class="card-body">
55                      <h5 class="card-title text-center">Registro de Usuario</h5>
56                      <p class="card-text text-center">Podra registrar los miembros del equipo de la tienda</p>
57                      <center><a class="btn btn-primary" href="registrar.html" role="button">Registrar</a></center>
58                  </div>
59              </div>
60          </div>
61          <div class="col">
62              <div class="card h-100">
63                  
64                  <div class="card-body">
65                      <h5 class="card-title text-center">Consultas</h5>
66                      <p class="card-text text-center">Podra buscar los usuarios registrados en la tienda</p>
67                      <center><a class="btn btn-primary" href="listaElemento.html" role="button">Buscar</a></center>
68                  </div>
69              </div>
70          </div>
71
72      </div>
73
74      </div>
75
76
```

Formulario de registro para miembros del equipo de la perfumería.



Registro de Usuario

Nombre:

Apellido:

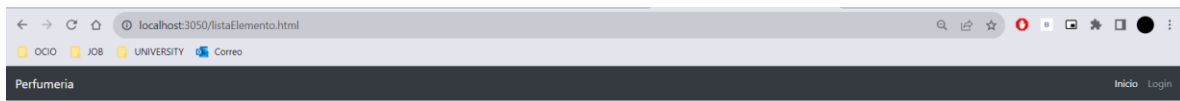
Email:

```
<div class="form-group mx-auto p-2" style="width: 200px;">
  <label for="nombre">Nombre:</label>
  <input type="text" class="form-control" id="nombre" name="nombre" placeholder="Nombre" required>
</div>
<div class="form-group mx-auto p-2" style="width: 200px;">
  <label for="apellido">Apellido:</label>
  <input type="text" class="form-control" id="apellido" name="apellido" placeholder="Apellido" required>
</div>
<div class="form-group mx-auto p-2" style="width: 200px;">
  <label for="email">Email:</label>
  <input type="email" class="form-control" id="email" name="email" placeholder="Email" required>
</div>
<button type="submit" class="btn btn-success">Guardar</button>
</form>
</div>
</center>

<script>
// Agregar un cliente al enviar el formulario
document.getElementById('agregarCliente').addEventListener('submit', function (event) {
  event.preventDefault();
  const nombre = document.getElementById('nombre').value;
  const apellido = document.getElementById('apellido').value;
  const email = document.getElementById('email').value;

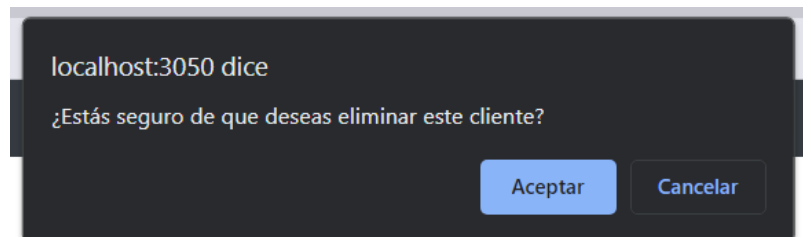
  fetch('/agregar', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({ nombre, apellido, email })
  })
  .then(response => response.text())
  .then(message => {
    alert(message);
  })
});
```


Listado de usuarios registros en la pagina web, donde se puede realizar eliminación y actualización.



Listado de Clientes

ID	NOMBRE	APELLIDO	EMAIL	ACCIONES	
1	Erika	Molina	molina@mail.com	Editar	Eliminar
4	Daniela	Tabares	daniela@mail.com	Editar	Eliminar
6	lucia	fernandez	lucia@mail.com	Editar	Eliminar
7	Alejandro	Tabares	alejandro@mail.com	Editar	Eliminar
10	Maria	Villamizar	mvillamizar@mail.com	Editar	Eliminar



Actualizar Usuario

Nombre:

Apellido:

Email:

```

<script>
  fetch('/clientes')
    .then(response => response.json())
    .then(data => {
      const listaClientes = document.getElementById('listaClientes');
      listaClientes.innerHTML = ''; // Limpiar la lista
      data.forEach(cliente => {
        const row = document.createElement('tr');
        row.innerHTML = `
          <th scope="row">${cliente.id}</th>
          <td>${cliente.nombre}</td>
          <td>${cliente.apellido}</td>
          <td>${cliente.email}</td>
          <td>
            <a href="#" onclick="editarCliente(${cliente.id})" class="btn btn-primary">Editar</a>
            <a href="#" onclick="eliminarCliente(${cliente.id})" class="btn btn-danger">Eliminar</a>
          </td>
        `;
        listaClientes.appendChild(row);
      });
    });

  // Función para editar un usuario
  function editarCliente(id) {
    // Redirige a una página de edición o muestra un formulario de edición
    window.location.href = `/editarCliente.html?id=${id}`;
  }

  // Función para eliminar un usuario
  function eliminarCliente(id) {
    // Muestra un mensaje de confirmación
    const confirmar = confirm('¿Estás seguro de que deseas eliminar este cliente?');

    if (confirmar) {
      // Realiza una solicitud al servidor para eliminar el cliente
      fetch(`/eliminar/${id}`, {

```