



LENGUAJE CLIENTE-SERVIDOR

1. Portada

- **Título de la tarea:** LENGUAJE CLIENTE-SERVIDOR
- **Asignatura:** DESARROLLO WEB AVANZADO
- **Nombre de los estudiantes:** ANTHONY NESTOR VILLARREAL MACÍAS, ARIEL LEONIDAS REYES CERON
- **Fecha de entrega:** 25-11-2024
- **Nombre del profesor o docente:** DORIS KARINA CHICAIZA ANGAMARCA
- **Universidad:** ESPE

2. Objetivo General

Diseñar e implementar un sistema web funcional que permita gestionar reservas de manera eficiente, utilizando tecnologías modernas de desarrollo de software y asegurando una experiencia de usuario agradable mediante una interfaz intuitiva.

3. Desarrollo

- **Concepto base:**

Un sistema de reservas es una herramienta digital diseñada para gestionar la programación y administración de servicios, productos o recursos. Estos sistemas permiten registrar, modificar, consultar y eliminar reservas, optimizando procesos y garantizando la organización de información clave.

En este proyecto, se desarrolló un sistema web completo que consta de un BackEnd para gestionar la lógica de negocio y la comunicación con la base de datos, y un FrontEnd que ofrece una interfaz amigable para los usuarios. Este desarrollo es importante porque permite integrar tecnologías y conocimientos aprendidos en clase, fomentando la práctica de habilidades de programación en un entorno real.

- **Herramientas de desarrollo:**

- Backend:

- Node.js: Entorno de ejecución para JavaScript, usado para construir el servidor y manejar las rutas.
 - Express.js: Framework de Node.js que facilita la creación de aplicaciones web y APIs RESTful.
 - MySQL: Sistema de gestión de bases de datos relacional, empleado para almacenar la información de las reservas.
 - Axios: Librería para realizar solicitudes HTTP desde el cliente al servidor.

- FrontEnd:



- React.js: Biblioteca de JavaScript usada para construir la interfaz de usuario interactiva.
- CSS: Lenguaje de estilos empleado para mejorar el diseño visual del FrontEnd.
- Otras herramientas:
 - Visual Studio Code: Editor de texto utilizado para escribir y organizar el código del proyecto.
 - POSTMAN: Herramienta para probar y depurar las rutas del BackEnd.
- **Cuerpo o Desarrollo:**
 - **Diseño del BackEnd**

El BackEnd sigue una arquitectura estructurada, dividiéndose en las siguientes capas:

- Configuración de la base de datos:

Se estableció un pool de conexiones con MySQL para garantizar una gestión eficiente de las solicitudes concurrentes.

```
1  require('dotenv').config();
2  const mysql = require('mysql2');
3
4  // Crear el pool de conexión a la base de datos
5  const pool = mysql.createPool({
6    host: process.env.DB_HOST,
7    user: process.env.DB_USER,
8    password: process.env.DB_PASSWORD,
9    database: process.env.DB_NAME,
10  });
11
12  // Verificar la conexión al inicializar
13  pool.getConnection((err) => {
14    if (err) {
15      console.error('Error al conectar con la base de datos:', err.message);
16    } else {
17      console.log('Conectado a la base de datos MySQL');
18    }
19  });
20
21  // Exportar el pool para que pueda ser usado en otras partes de la aplicación
22  module.exports = pool.promise();
```

- **Controladores, servicios y repositorios:**

Esta división permite separar responsabilidades y facilitar el mantenimiento del código. Por ejemplo, los controladores manejan las solicitudes del cliente, los servicios contienen la lógica de negocio y los repositorios interactúan directamente con la base de datos.



- **Endpoints del API REST:**

El sistema soporta las siguientes operaciones:

- GET: Obtener todas las reservas o una específica por su ID.
- POST: Crear una nueva reserva.
- PUT: Actualizar una reserva existente.
- DELETE: Eliminar una reserva.

```
1 router.get('/', ReservaController.obtenerTodas);  
2 router.post('/', ReservaController.crear); // Debe existir esta ruta  
3 router.get('/:id', ReservaController.obtenerPorId);  
4 router.put('/:id', ReservaController.actualizar);  
5 router.delete('/:id', ReservaController.eliminar);
```

- **Diseño del FrontEnd**

El FrontEnd se implementó con React.js y se dividió en componentes reutilizables.

- Lista de reservas:

Se desarrolló un componente (ReservaList) que muestra las reservas en un cuadro con un diseño atractivo.



```
1 <div className="reserva-list-container">
2   <h2>Lista de Reservas</h2>
3   <Link to="/nueva" className="create-btn">
4     Crear nueva reserva
5   </Link>
6   <table className="reserva-table">
7     <thead>
8       <tr>
9         <th>Nombre Cliente</th>
10        <th>Tipo</th>
11        <th>Fecha</th>
12        <th>Hora</th>
13        <th>Acciones</th>
14      </tr>
15    </thead>
16    <tbody>
17      {reservas.map((reserva) => (
18        <tr key={reserva.id}>
19          <td>{reserva.nombre_cliente}</td>
20          <td>{reserva.tipo}</td>
21          <td>{reserva.fecha}</td>
22          <td>{reserva.hora}</td>
23          <td>
24            <Link to={`/editar/${reserva.id}`} className="edit-btn">
25              Editar
26            </Link>
27            <button onClick={() => deleteReserva(reserva.id)} className="delete-btn">
28              Eliminar
29            </button>
30          </td>
31        </tr>
32      )
33    </tbody>
34  </table>
35 </div>
```

Estilo en CSS



```
1  .reserva-list-container {
2    margin: 20px;
3    padding: 20px;
4    background-color: #f9f9f9;
5    border-radius: 8px;
6    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
7  }
8
9  h2 {
10   margin-bottom: 20px;
11   color: #333;
12 }
13
14 .create-btn {
15   display: inline-block;
16   margin-bottom: 15px;
17   padding: 10px 15px;
18   background-color: #007bff;
19   color: #fff;
20   text-decoration: none;
21   border-radius: 5px;
22 }
23
24 .create-btn:hover {
25   background-color: #0056b3;
26 }
27
28 .reserva-table {
29   width: 100%;
30   border-collapse: collapse;
31 }
32
33 .reserva-table th,
34 .reserva-table td {
35   padding: 12px 15px;
36   text-align: left;
37   border-bottom: 1px solid #ddd;
38 }
39
40 .reserva-table th {
41   background-color: #007bff;
42   color: white;
43 }
44
45 .reserva-table tr:hover {
46   background-color: #f1f1f1;
47 }
48
49 .edit-btn,
50 .delete-btn {
51   padding: 5px 10px;
52   border: none;
53   border-radius: 5px;
54   cursor: pointer;
55 }
56
57 .edit-btn {
58   background-color: #28a745;
59   color: white;
60   margin-right: 10px;
61 }
62
63 .edit-btn:hover {
64   background-color: #218838;
65 }
66
67 .delete-btn {
68   background-color: #dc3545;
69   color: white;
70 }
71
72 .delete-btn:hover {
73   background-color: #c82333;
74 }
75
```

- **Formulario de reservas:**



Se creó un formulario (ReservaForm) para gestionar la creación y edición de reservas. Este formulario incluye validación de campos y estilos personalizados.

```
1 <div className="form-container">
2   <form onSubmit={handleSubmit} className="reserva-form">
3     <h2{id ? 'Editar Reserva' : 'Nueva Reserva'}></h2>
4     <div className="form-group">
5       <label htmlFor="tipo">Tipo</label>
6       <input
7         type="text"
8         id="tipo"
9         name="tipo"
10        placeholder="Tipo de reserva"
11        value={reserva.tipo}
12        onChange={handleChange}
13      />
14    </div>
15    <div className="form-group">
16      <label htmlFor="nombre_cliente">Nombre Cliente</label>
17      <input
18        type="text"
19        id="nombre_cliente"
20        name="nombre_cliente"
21        placeholder="Nombre del cliente"
22        value={reserva.nombre_cliente}
23        onChange={handleChange}
24      />
25    </div>
26    <div className="form-group">
27      <label htmlFor="fecha">Fecha</label>
28      <input type="date" id="fecha" name="fecha" value={reserva.fecha} onChange={handleChange} />
29    </div>
30    <div className="form-group">
31      <label htmlFor="hora">Hora</label>
32      <input type="time" id="hora" name="hora" value={reserva.hora} onChange={handleChange} />
33    </div>
34    <div className="form-group">
35      <label htmlFor="detalles">Detalles</label>
36      <textarea
37        id="detalles"
38        name="detalles"
39        placeholder="Detalles adicionales"
40        value={reserva.detalles}
41        onChange={handleChange}
42      ></textarea>
43    </div>
44    <div className="form-buttons">
45      <button type="submit" className="btn-submit">
46        {id ? 'Actualizar' : 'Crear'}
47      </button>
48      <button type="button" className="btn-cancel" onClick={() => navigate('/')}>
49        Cancelar
50      </button>
51    </div>
52  </form>
53 </div>
```

Estilo en CSS



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

```
1  .form-container {
2    margin: 20px auto;
3    padding: 20px;
4    max-width: 500px;
5    background-color: #f9f9f9;
6    border-radius: 10px;
7    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
8  }
9
10 .reserva-form {
11   display: flex;
12   flex-direction: column;
13 }
14
15 h2 {
16   text-align: center;
17   margin-bottom: 20px;
18   color: #333;
19 }
20
21 .form-group {
22   margin-bottom: 15px;
23 }
24
25 .form-group label {
26   display: block;
27   margin-bottom: 5px;
28   color: #555;
29 }
30
31 .form-group input,
32 .form-group textarea {
33   width: 100%;
34   padding: 10px;
35   border: 1px solid #ddd;
36   border-radius: 5px;
37   font-size: 16px;
38 }
39
40 .form-group textarea {
41   resize: none;
42   height: 80px;
43 }
44
45 .form-group input:focus,
46 .form-group textarea:focus {
47   border-color: #007bff;
48   outline: none;
49   box-shadow: 0 0 5px rgba(0, 123, 255, 0.3);
50 }
51
52 .form-buttons {
53   display: flex;
54   justify-content: space-between;
55   margin-top: 20px;
56 }
57
58 .btn-submit {
59   padding: 10px 15px;
60   background-color: #007bff;
61   color: white;
62   border: none;
63   border-radius: 5px;
64   cursor: pointer;
65 }
66
67 .btn-submit:hover {
68   background-color: #0056b3;
69 }
70
71 .btn-cancel {
72   padding: 10px 15px;
73   background-color: #dc3545;
74   color: white;
75   border: none;
76   border-radius: 5px;
77   cursor: pointer;
78 }
79
80 .btn-cancel:hover {
81   background-color: #c82333;
82 }
```



- Apoyo visual:
 - Captura del formulario:

Sistema de Reservas

Lista de Reservas

Crear nueva reserva

Nombre Cliente	Tipo	Fecha	Hora	Acciones
Mariano Monge	habitacion	2024-11-20T05:00:00.000Z	15:00:00	<div>EditarEliminar</div>
Doris	boleto	2024-11-14T05:00:00.000Z	04:12:00	<div>EditarEliminar</div>
Anthony Villarreal	habitacion	2001-09-12T05:00:00.000Z	12:00:00	<div>EditarEliminar</div>

Sistema de Reservas

Nueva Reserva

Tipo

Tipo de reserva

Nombre Cliente

Nombre del cliente

Fecha

dd/mm/aaaa

Hora

--:--

Detalles

Detalles adicionales

Crear

Cancelar

Sistema de Reservas

Editar Reserva

Tipo

habitacion

Nombre Cliente

Mariano Monge

Fecha

dd/mm/aaaa

Hora

15:00

Detalles

Habitación doble con vista al mar

Actualizar

Cancelar



4. Conclusión

El sistema de reservas desarrollado integrar un BackEnd eficiente y un FrontEnd interactivo, logrando una aplicación funcional que gestiona reservas de forma intuitiva. Este proyecto permitió aplicar conocimientos de desarrollo web, arquitectura de software y diseño, además de promover habilidades de resolución de problemas.

5. Referencias

- W3Schools. (2024). *CSS Tutorial*. Recuperado de <https://www.w3schools.com/css/>
- Node.js Foundation. (2024). *Node.js Documentation*. Recuperado de <https://nodejs.org/>
- React. (2024). *React Documentation*. Recuperado de <https://reactjs.org/>
- <https://github.com/ANTHONYNESTORVILLARREALMACIAS/Desarrollo-Web-Avanzado-AnthonyV-ArielR.git>