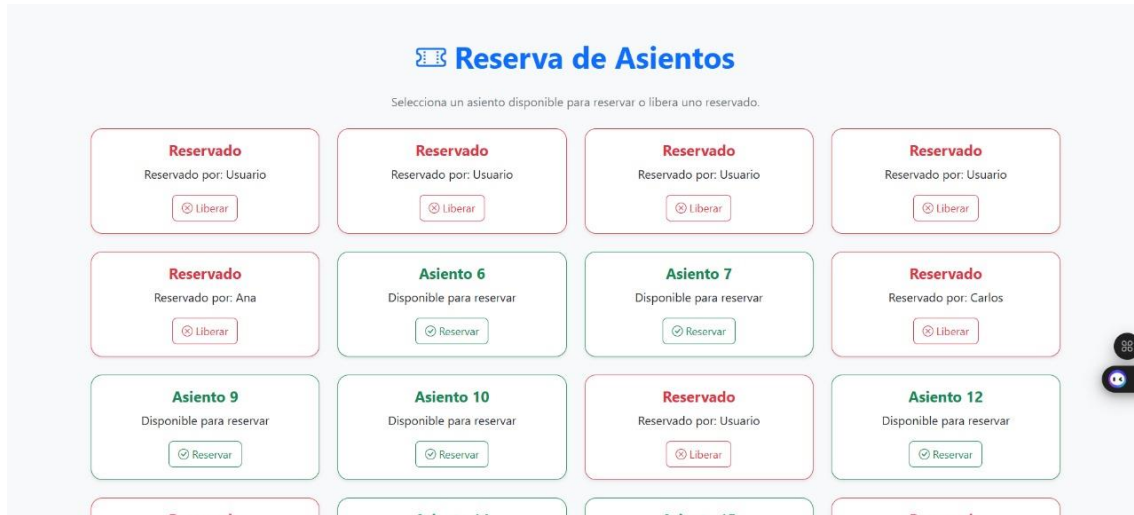


Universidad de las Fuerzas Armadas ESPE

Nombre: Ariel Reyes, Anthony Villarreal

Imágenes de la ejecución



localhost:3000 dice

Asiento 2 reservado con éxito.

Aceptar

Selecciona un asiento disponible para reservar o libera uno reservado.

Asiento 2

Disponible para reservar

✓ Reservar

Reservado

Reservado por: Usuario

⊗ Liberar

Código de Backend:

1. Index.js

```
1  const express = require("express");
2  const bodyParser = require("body-parser");
3  const cors = require("cors");
4  const mysql = require("mysql");
5
6  const app = express();
7  app.use(cors());
8  app.use(bodyParser.json());
9
10 // Configuración de la base de datos
11 const db = mysql.createConnection({
12   host: "localhost",
13   user: "root",
14   password: "", // Cambia según tu configuración
15   database: "reservas_db", // Nombre de la base de datos
16 });
17
18 db.connect((err) => {
19   if (err) {
20     console.error("Error al conectar a la base de datos:", err);
21     return;
22   }
23   console.log("Conectado a la base de datos MySQL");
24 });
25
26 // Obtener todos los asientos
27 app.get("/api/asientos", (req, res) => {
28   const query = "SELECT * FROM asientos";
29   db.query(query, (err, results) => {
30     if (err) {
31       res.status(500).send({ error: "Error al obtener los asientos" });
32       return;
33     }
34     res.json(results);
35   });
36 });
37
38 // Reservar un asiento
39 app.post("/api/asientos/reservar", (req, res) => {
40   const { numero, reservadoPor } = req.body;
41   const query = "UPDATE asientos SET Disponible = 0, reservadoPor = ? WHERE Numero = ? AND Disponible = 1";
42   db.query(query, [reservadoPor, numero], (err, results) => {
43     if (err || results.affectedRows === 0) {
44       res.status(400).send({ message: "El asiento no pudo ser reservado." });
45       return;
46     }
47     res.send({ message: `Asiento ${numero} reservado con éxito.` });
48   });
49 });
50
51 // Liberar un asiento
52 app.post("/api/asientos/liberar", (req, res) => {
53   const { numero } = req.body;
54   const query = "UPDATE asientos SET Disponible = 1, reservadoPor = NULL WHERE Numero = ?";
55   db.query(query, [numero], (err, results) => {
56     if (err || results.affectedRows === 0) {
57       res.status(400).send({ message: "El asiento no pudo ser liberado." });
58       return;
59     }
60     res.send({ message: `Asiento ${numero} liberado con éxito.` });
61   });
62 });
63
64 // Iniciar el servidor
65 const PORT = 5000;
66 app.listen(PORT, () => {
67   console.log(`Servidor ejecutándose en http://localhost:${PORT}`);
68 });
69
```

2. Ejecucion BackEnd

```
PS C:\Users\antho\Desktop\PERSONAL\9_SEMESTRE\DESARROLLO WEB AVANZADO\APPS\GITHUB\Examen_1P_Reyes_Villarreal\backend> node index.js
Servidor ejecutándose en http://localhost:5000
Conectado a la base de datos MySQL
```

Código FrontEnd:

App.js

```
1 import React, { useState, useEffect } from "react";
2 import axios from "axios";
3 import "bootstrap/dist/css/bootstrap.min.css";
4 import "./App.css"; // Puedes agregar más estilos personalizados aquí
5
6 const App = () => {
7   const [asientos, setAsientos] = useState([]); // Estado inicial vacío para los asientos
8
9   // Obtener los asientos desde el backend
10  useEffect(() => {
11    const fetchAsientos = async () => {
12      try {
13        const response = await axios.get("http://localhost:5000/api/asientos");
14        setAsientos(response.data);
15      } catch (error) {
16        console.error("Error al obtener los asientos:", error.message);
17        alert("No se pudieron cargar los asientos.");
18      }
19    };
20
21    fetchAsientos(); // Llamar a la función al montar el componente
22  }, []);
23
24  // Reservar un asiento
25  const reservarAsiento = async (numero) => {
26    try {
27      const response = await axios.post(
28        "http://localhost:5000/api/asientos/reservar",
29        { numero, reservadoPor: "Usuario" }
30      );
31      alert(response.data.message);
32      setAsientos((prev) =>
33        prev.map((asiento) =>
34          asiento.Numero === numero
35            ? { ...asiento, Disponible: 0, reservadoPor: "Usuario" }
36            : asiento
37        )
38      );
39    } catch (error) {
40      console.error("Error al reservar el asiento:", error.message);
41      alert("El asiento no pudo ser reservado.");
42    }
43  };
44
45  // Liberar un asiento
46  const liberarAsiento = async (numero) => {
47    try {
48      const response = await axios.post(
49        "http://localhost:5000/api/asientos/liberar",
50        { numero }
51      );
52      alert(response.data.message);
53      setAsientos((prev) =>
54        prev.map((asiento) =>
55          asiento.Numero === numero
56            ? { ...asiento, Disponible: 1, reservadoPor: null }
57            : asiento
58        )
59      );
60    } catch (error) {
61      console.error("Error al liberar el asiento:", error.message);
62      alert("El asiento no pudo ser liberado.");
63    }
64  };
65 }
```

```

65
66   return (
67     <div className="container mt-5">
68       <h1 className="text-center mb-4 text-primary fw-bold">
69         <i className="bi bi-ticket-perforated"></i> Reserva de Asientos
70       </h1>
71       <p className="text-center text-secondary mb-4">
72         Selecciona un asiento disponible para reservar o libera uno reservado.
73       </p>
74       <div className="row">
75         {asientos.map((asiento) => (
76           <div key={asiento.Numero} className="col-md-4 col-lg-3 mb-4">
77             <div
78               className={`card shadow-sm ${
79                 asiento.Disponible ? "border-success" : "border-danger"
80               }`}
81             >
82               <div className="card-body text-center">
83                 <h5
84                   className={`card-title fw-bold ${
85                     asiento.Disponible ? "text-success" : "text-danger"
86                   }`}
87                 >
88                   {asiento.Disponible
89                     ? `Asiento ${asiento.Numero}`
90                     : `Reservado`}
91                 </h5>
92                 <p className="card-text">
93                   {asiento.Disponible
94                     ? "Disponible para reservar"
95                     : `Reservado por: ${asiento.reservadoPor || "N/A"}`}
96                 </p>
97                 <button
98                   className={`btn ${
99                     asiento.Disponible ? "btn-outline-success" : "btn-outline-danger"
100                   }`}
101                   onClick={() =>
102                     asiento.Disponible
103                       ? reservarAsiento(asiento.Numero)
104                       : liberarAsiento(asiento.Numero)
105                   }
106                 >
107                   {asiento.Disponible ? (
108                     <>
109                       <i className="bi bi-check-circle"></i> Reservar
110                     </>
111                     ) : (
112                     <>
113                       <i className="bi bi-x-circle"></i> Liberar
114                     </>
115                   )}
116                 </button>
117               </div>
118             </div>
119           </div>
120         ))}
121       </div>
122     </div>
123   );
124 };
125
126 export default App;
127

```

App.css

```
1  .card {  
2    border-radius: 15px;  
3    transition: transform 0.2s, box-shadow 0.2s;  
4  }  
5  
6  .card:hover {  
7    transform: scale(1.05);  
8    box-shadow: 0px 4px 15px rgba(0, 0, 0, 0.2);  
9  }  
10  
11 .btn {  
12   font-size: 0.9rem;  
13 }  
14  
15 .text-primary {  
16   font-size: 2.5rem;  
17 }  
18
```

public/index.html

```
1  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons/font/bootstrap-icons.css" rel="stylesheet" />
```

Ejecución FrontEnd

```
Compiled successfully!  
  
You can now view frontend in the browser.  
  
http://localhost:3000  
  
Note that the development build is not optimized.  
To create a production build, use npm run build.  
  
webpack compiled successfully
```

Enlace del repositorio:

<https://github.com/ANTHONYNESTORVILLARREALMACIAS/Desarrollo-Web-Avanzado-AnthonyV-ArielR.git>