Taller Práctico: Integración FullStack

Julian F. Latorre Bootcamp Desarrollo Web Full Stack

i Información del Taller

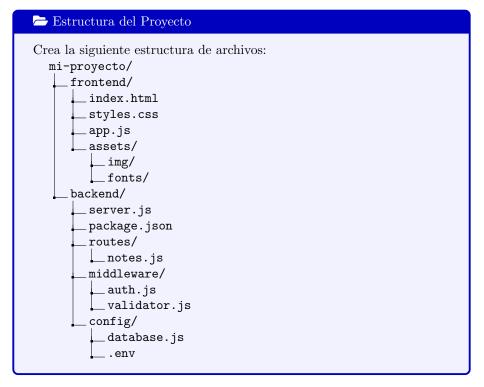
Este taller está diseñado para practicar los conceptos fundamentales de integración FullStack. Trabajaremos en la creación de una aplicación simple de notas donde implementaremos la comunicación frontend-backend, buenas prácticas y preparación para despliegue.

1. 💥 Herramientas Necesarias

- ✓ Editor de código (VS Code recomendado)
- nede Node.js instalado
- git Git instalado
- Navegador web moderno

2. ₹**≡** Ejercicios Prácticos

2.1. Ejercicio 1: Configuración Inicial



```
⟨⟩ Código Base Frontend
Archivo: index.html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,</pre>
   initial-scale=1.0">
    <title>Mi App de Notas</title>
    <link rel="stylesheet" href="styles.css">
<body>
    <div class="container">
        <h1>Mis Notas</h1>
        <div id="notas-form">
            <input type="text" id="nota-titulo" placeholder="</pre>
   Título">
            <textarea id="nota-contenido" placeholder="</pre>
    Contenido"></textarea>
            <button onclick="crearNota()">Crear Nota</button>
        </div>
        <div id="notas-lista"></div>
    <script src="app.js"></script>
</body>
</html>
```

2.2. Ejercicio 2: Implementación del Frontend

```
Archivo: styles.css

.container {
    max-width: 800px;
    margin: 0 auto;
    padding: 20px;
}

.nota {
    border: 1px solid #ddd;
    padding: 10px;
    margin: 10px 0;
    border-radius: 5px;
}

/* Añade más estilos según necesites */
```

Js Lógica JavaScript

```
Archivo: app.js
// Variables globales
const API_URL = 'http://localhost:3000';
// Función para crear una nota
async function crearNota() {
   const titulo = document.getElementById('nota-titulo').
   value;
   const contenido = document.getElementById('nota-contenido
   ').value;
   try {
        const response = await fetch('${API_URL}/notas', {
            method: 'POST',
            headers: {
               'Content-Type': 'application/json'
            body: JSON.stringify({ titulo, contenido })
        if (response.ok) {
            cargarNotas();
            limpiarFormulario();
   } catch (error) {
       console.error('Error:', error);
}
// Completar las funciones cargarNotas() y limpiarFormulario
    ()
```

2.3. Ejercicio 3: Implementación del Backend

```
➡ Servidor Express
Archivo: server.js
const express = require('express');
const cors = require('cors');
const app = express();
app.use(cors());
app.use(express.json());
// Array para almacenar las notas (en memoria)
let notas = [];
// Ruta para crear notas
app.post('/notas', (req, res) => {
   const { titulo, contenido } = req.body;
    // Completar la lógica
});
// Ruta para obtener notas
app.get(,/notas, (req, res) => {
    // Completar la lógica
app.listen(3000, () => {
    console.log('Servidor corriendo en puerto 3000');
```

3. **Y** Retos Adicionales

Implementa la funcionalidad de eliminar notas

Añade validación de datos en el frontend y backend

Implementa persistencia de datos usando localStorage

Mejora el diseño CSS para hacerlo responsivo

4. Preguntas Guía

Responde las siguientes preguntas mientras desarrollas:

- ¿Cómo manejas los errores en las peticiones fetch?
- ¿Qué medidas de seguridad deberías implementar?
- ¿Cómo mejorarías el rendimiento de la aplicación?

5. **?** Tips para el Desarrollo

- **ൂ** Usa console.log() para debuggear
- X Utiliza las herramientas de desarrollo del navegador Valida siempre los datos de entrada
- 2 Mantén el código limpio y organizado

6. 🗣 Preparación para Despliegue

Checklist de Despliegue

- Verifica que todas las rutas estén correctamente configuradas
- Asegúrate de tener un archivo .gitignore apropiado
- Configura las variables de entorno necesarias
- Prueba la aplicación en modo producción

7. Entrega

Deberás entregar:

- Código fuente en un repositorio Git
- URL de la aplicación desplegada
- Documento con respuestas a las preguntas guía
- Breve explicación de los retos implementados