

Informe de Practica 04

Tema: Ajax y NodeJS

Nota

Integrantes	Escuela	Asignatura
Roni Companocca Checco Franco Jesus Cahua Soto Rodrigo Ferdinand Quispe Huaman Sebastian A Riveros Valeriano	Escuela Profesional de Ingeniería de Sistemas	Semestre: III Código:

Practica	Tema	Duración
04	Ajax y NodeJS	02 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 09 de Mayo 2024	Al 18 Mayo 2024

1. URL DE REPOSITORIO GITHUB Y EL VIDEO

- URL para el Repositorio GitHub.
- <https://github.com/SebastianRiveros/PW2-Lab4-A-ProyectoGrupal>

2. TAREA

- En grupos de 3 a 5 personas (los grupos se definirán en clase) implemente una aplicación web que navegue sobre archivos Markdown y permita:
- Listar los archivos Markdown disponibles
- Ver el contenido de un archivo Markdown traducido a HTML
- Crear nuevos archivos MarkDown y almacenarlos en el servidor La comunicación entre el cliente y el servidor tiene que ser usando JSON solamente.
- El cliente debe usar AJAX para sus peticiones
- El servidor debe usar NodeJS
- Su aplicación debe ser de página única, es decir que sólo habrá un archivo index.html y nada más.
- Si los enlaces proporcionado en esta guía no le son suficientes, puede revisar códigos en Internet que le ayuden con cosas como ejemplos: listar un directorio en NodeJS; pero deberá incluir los enlaces correspondientes en sus archivos como comentarios y sólo podrá usar código de stackoverflow, incluir código de cualquier otra fuente está prohibido y se considerará actitud deshonestas.

3. SOLUCION AL EJERCICIO PROPUESTO

3.1. Configuración del Entorno:

- Objetivo: Preparar el entorno de desarrollo para crear una aplicación web usando Node.js.
- Acciones: Instalamos Node.js y npm, creamos un directorio de proyecto, y lo inicializamos con npm init.

```
289 sudo apt update
290 sudo apt install nodejs
291 sudo apt install npm
292 node -v
293 npm -v
294 ls
295 cd PW2-Lab-A/
296 ls
297 mkdir Proyecto-Lab4
298 ls
299 cd Proyecto-Lab4/
300 npm init -y
```

3.2. Instalación de Dependencias:

- Objetivo: Añadir las bibliotecas necesarias para construir el servidor y manejar archivos Markdown.
- Acciones: Instalamos Express para el servidor web, body-parser para manejar datos JSON, fs para trabajar con el sistema de archivos, y marked para convertir Markdown a HTML.

```
zevaz@zevaz-HP-245-G8-Notebook-  
o04$ tree ProyectoGrupal -L 1  
ProyectoGrupal  
├── index.html  
├── markdown  
├── node_modules  
├── package.json  
├── package-lock.json  
└── server.js
```

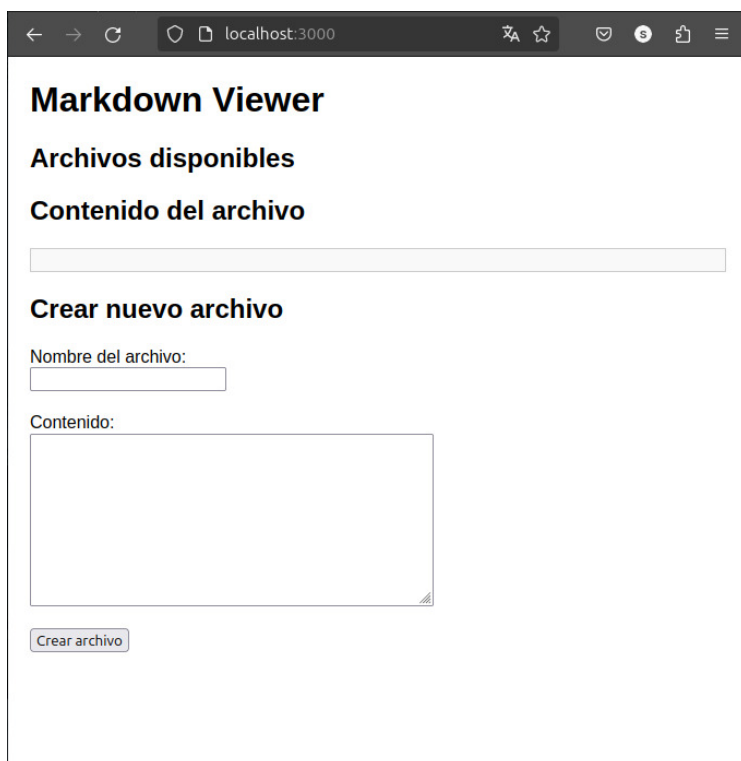
3.3. Creación del Servidor Node.js:

- Objetivo: Configurar un servidor que maneje solicitudes HTTP y sirva el contenido necesario.
- Acciones: Escribimos el archivo server.js para: Configurar rutas para listar, leer y crear archivos Markdown. Utilizar middleware para manejar datos JSON y servir archivos estáticos.

```
zevaz@zevaz-HP-245-G8-Notebook-PC:~/semestre3/PW2-Lab-A/PW2-Laboratorio-A/La  
o04/ProyectoGrupal$ node server.js  
Servidor escuchando en http://localhost:3000
```

3.4. Desarrollo de la Interfaz de Usuario:

- Objetivo: Proporcionar una página web donde los usuarios puedan interactuar con la aplicación.
- Acciones: Creamos index.html con: Secciones para listar archivos Markdown, mostrar contenido traducido a HTML, y crear nuevos archivos Markdown. JavaScript para manejar la interacción del usuario y hacer solicitudes AJAX al servidor.
- Pagina inicial:



← → ↻ localhost:3000

Markdown Viewer

Archivos disponibles

Contenido del archivo

Crear nuevo archivo

Nombre del archivo:

Contenido:

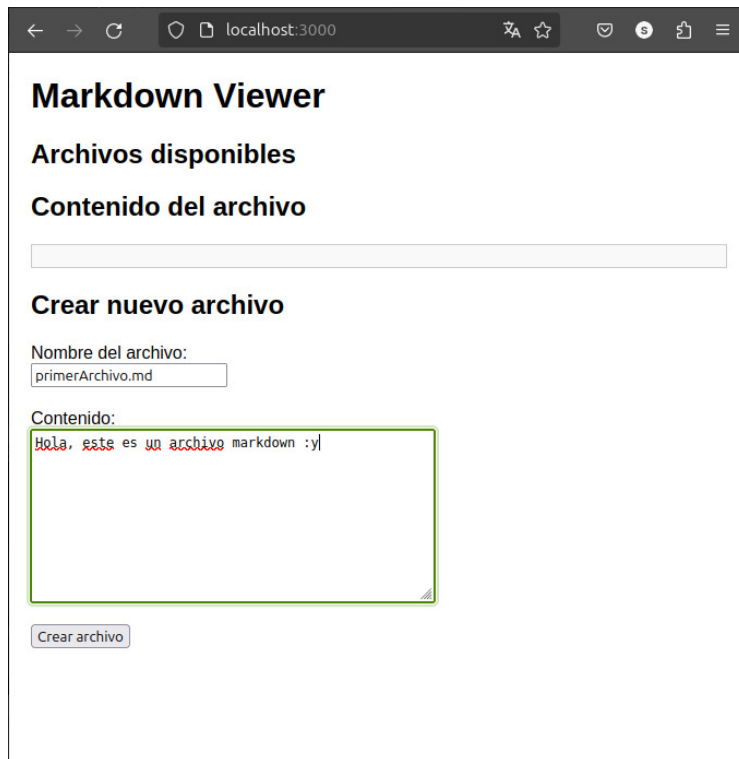
Crear archivo

3.5. Creación del Directorio Markdown:

- Objetivo: Definir un lugar en el servidor donde se almacenarán los archivos Markdown.
- Acciones: Creamos un directorio markdown en el proyecto.

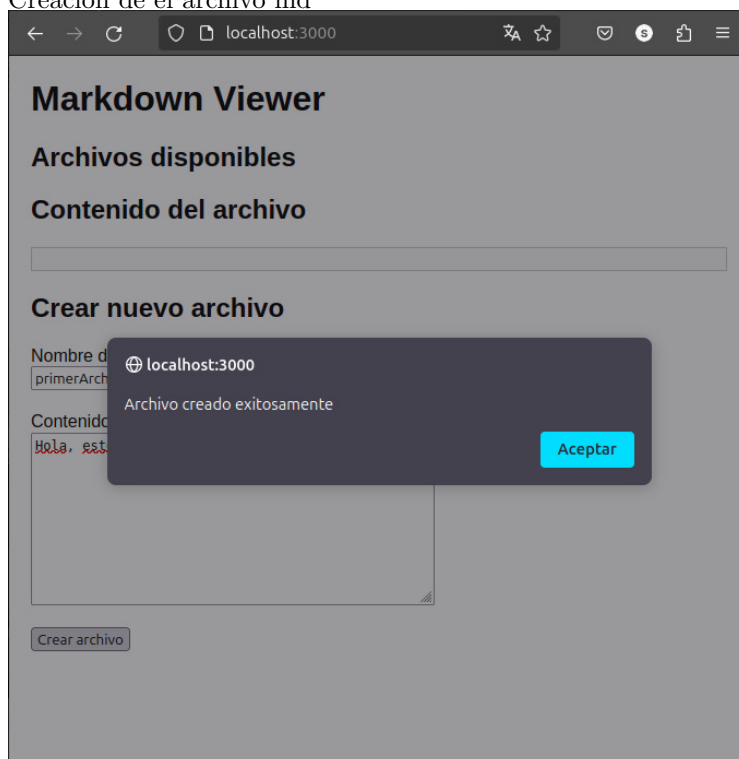
3.6. Pruebas y Ejecución:

- Objetivo: Asegurarnos de que todo funcione correctamente.
- Acciones: Ejecutamos el servidor con node server.js y probamos la aplicación en un navegador web.
- Escritura de un archivo md



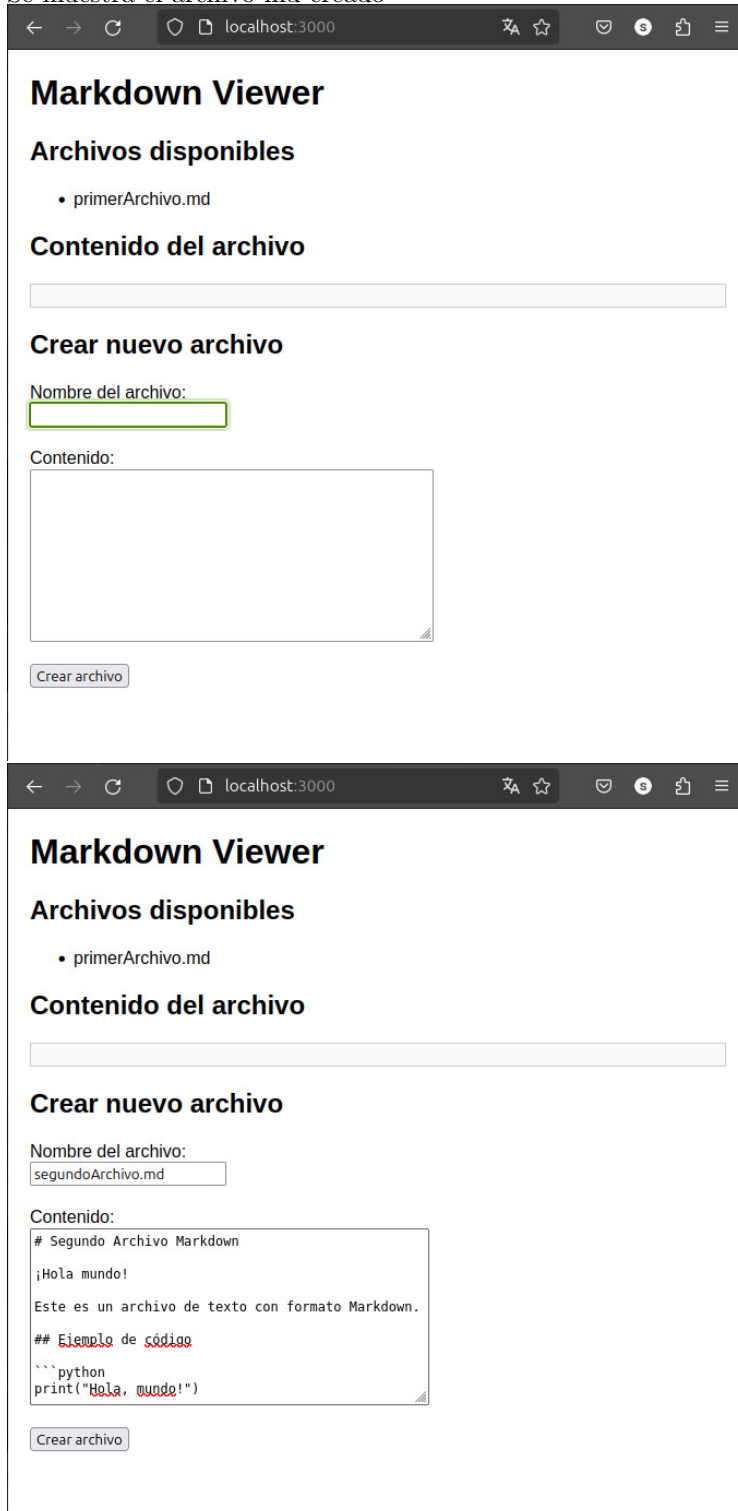
The screenshot shows a web browser window with the address bar displaying 'localhost:3000'. The page title is 'Markdown Viewer'. Below the title, there are three sections: 'Archivos disponibles', 'Contenido del archivo', and 'Crear nuevo archivo'. The 'Crear nuevo archivo' section contains a form with two fields: 'Nombre del archivo:' with the value 'primerArchivo.md' and 'Contenido:' with the value 'Hola, este es un archivo markdown :)' and a 'Crear archivo' button.

■ Creacion de el archivo md



The screenshot shows the same web browser window as the previous one, but with a dark overlay message box in the center. The message box contains the text 'localhost:3000' and 'Archivo creado exitosamente' with an 'Aceptar' button. The background of the page is dimmed.

- Se muestra el archivo md creado



← → ↻ 🛡️ 📄 localhost:3000 🔍 ☆ 📧 📄 📄 ☰

Markdown Viewer

Archivos disponibles

- primerArchivo.md

Contenido del archivo

Crear nuevo archivo

Nombre del archivo:

Contenido:

```
# Segundo Archivo Markdown
¡Hola mundo!
Este es un archivo de texto con formato Markdown.
## Ejemplo de código
```python
print("Hola, mundo!")
```

Crear archivo

### 3.7. Por Qué Hicimos Cada Paso

- Configurar el entorno y dependencias: Para tener las herramientas necesarias que nos permiten desarrollar y gestionar el proyecto.
- Crear el servidor Node.js: Para manejar las solicitudes del cliente, procesar datos y servir contenido dinámico.
- Desarrollar la interfaz de usuario: Para proporcionar una experiencia interactiva donde los usuarios puedan ver y manipular archivos Markdown.
- Crear el directorio Markdown: Para organizar y gestionar los archivos de contenido que la aplicación manejará.
- Probar y ejecutar: Para verificar que la aplicación funcione correctamente y cumpla con los requisitos especificados.
- Código de server.js

```
JS server.js x
ProyectoGrupal > JS server.js > ...
1 const express = require('express');
2 const bodyParser = require('body-parser');
3 const fs = require('fs');
4 const path = require('path');
5 const marked = require('marked');
6
7 const app = express();
8 const PORT = 3000;
9
10 // Middleware para analizar JSON
11 app.use(bodyParser.json());
12
13 // Servir archivos estáticos (index.html)
14 app.use(express.static(path.join(__dirname)));
15
16 // Ruta para listar archivos Markdown
17 app.get('/api/archivos', (req, res) => {
18 fs.readdir('./markdown', (err, files) => {
19 if (err) {
20 return res.status(500).json({ error: 'Error al leer el directorio' });
21 }
22 const markdownFiles = files.filter(file => file.endsWith('.md'));
23 res.json(markdownFiles);
24 });
25 });
26
27 // Ruta para obtener el contenido de un archivo Markdown
28 app.get('/api/archivo/:nombre', (req, res) => {
29 const fileName = req.params.nombre;
30 const filePath = path.join(__dirname, 'markdown', fileName);
31 }
```

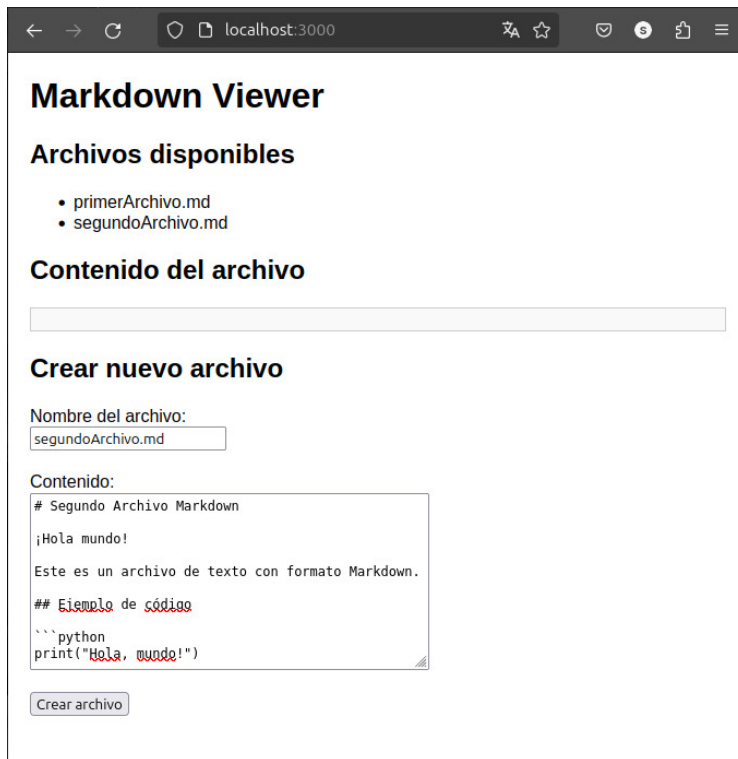
- código de index.html

```
<> index.html x
ProyectoGrupal > <> index.html > html > body > script > document.addEventListener("DOMContentLoaded"
2 <html lang="en">
22 <body>
36 <script>
37 document.addEventListener('DOMContentLoaded', function () {
38 const archivoLista = document.getElementById('archivo-lista');
39 const archivoContenido = document.getElementById('archivo-contenido');
40 const formNuevoArchivo = document.getElementById('form-nuevo-archivo');
41
42 // Función para listar archivos Markdown
43 function listarArchivos() {
44 fetch('/api/archivos')
45 .then(response => response.json())
46 .then(data => {
47 archivoLista.innerHTML = '';
48 data.forEach(file => {
49 const li = document.createElement('li');
50 li.textContent = file;
51 li.addEventListener('click', () => verArchivo(file));
52 archivoLista.appendChild(li);
53 });
54 });
55 }
56
57 // Función para ver el contenido de un archivo Markdown
58 function verArchivo(nombre) {
59 fetch(`/api/archivo/${nombre}`)
60 .then(response => response.json())
61 .then(data => {
62 archivoContenido.innerHTML = data.content;
63 });
64 }

```

- Este es la ejecución Final





## 4. REFERENCIAS

- JavaScript code using the latest ECMAScript. Packt Publishing Ltd, 2018.
- Greg Lim. Beginning Node.js, Express MongoDB Development. Amazon, 2019.