

# Informe de Laboratorio 03

## Tema: NodeJS

Nota

Estudiante	Escuela	Asignatura
Diego Claudio Nina Suyo dninas@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: I Código: 20190214

Laboratorio	Tema	Duración
03	NodeJS	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - A	Del 3 Junio 2023	Al 5 Junio 2023

## 1. Tarea

Ejercicio 1. Cree una aplicación NodeJS con express, para administrar una agenda personal.

Home (‘/’) : Página Principal

Preferiblemente trabaje todo en una misma interfaz.

Ejemplo de estructura de la agenda cuando se quiere “Ver eventos”.

```
agenda[DIR]
|
|----- 2023-05-15 [DIR]
|           |----- 10-00.txt [FILE]
|           |----- 13-30.txt [FILE]
|----- 2023-05-20 [DIR]
|           |----- 09-10.txt [FILE]
|           |----- 20-40.txt [FILE]
```

La aplicación debe permitir:

- Crear evento: fecha y hora. (Si ya existe el archivo no debería ingresar el evento)(La primera línea es el título del evento, las demás líneas son la descripción del evento.
- Editar evento. (Se muestran el archivo donde esta el detalle del evento)

- Eliminar evento.
- Ver eventos. Utilizar el formato árbol especificado anteriormente, donde debería incluirse sólo el título del evento.

**Recomendaciones:**

Interfaz que trabaja todo un CRUD en una sólo vista. (Se pueden usar ventanas emergentes)

## 2. Equipos, materiales y temas utilizados

- Sistema Operativo Manjaro Linux x86\_64 6.1.31-1-MANJARO
- VIM 9.0.
- Visual Studio Code
- Git 2.40.1
- tree
- Cuenta en GitHub con el correo institucional.
- HTML, CSS.
- Creación de página web
- Javascript
- NodeJS

## 3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.  
<https://github.com/DiegoNiS/pw2-lab-d-23a.git>
- URL para el laboratorio 01 en el Repositorio GitHub.  
<https://github.com/DiegoNiS/pw2-lab-d-23a/tree/main/lab-03>

## 4. Actividades con el repositorio GitHub

### 4.1. Ubicandonos en el repositorio de Github

Como se continúa un trabajo en el mismo repositorio del laboratorio anterior nuevamente es conveniente realizar un `git pull` sobre el repositorio en nuestra máquina anfitriona. Luego de esto ubicarnos en la carpeta raíz del repositorio y crear una parteta especial para el siguiente laboratorio y no trabajar sobre lo el contenido total:

Listing 1: Creación de la carpeta del laboratorio

```
cd ruta/a/la/raiz/pw2-lab-d-23a
git pull
mkdir lab-03 && cd lab-03
```

### 4.2. Creando archivos esenciales para el funcionamiento

Se crea el archivo HTML, y dentro de este colocar el CSS por medio de los tags de style, además de el JavaScript que está incluido tambien en el HTML, por el tag script, esto con fines de facilitar el funcionamiento de NodeJS.

Además se debe crear el archivo `app.js` que debe correrse en el servidor.

Tener en cuenta que se debe tener instalado npm & nodejs, que en este caso se ejecutarán los siguientes comandos en terminal para instalarlos en la terminal de manjaro:

Listing 2: Instalación de paquetes necesarios en linux

```
$ sudo pacman -Syu
$ sudo pacman -S npm nodejs
```

Dentro de la carpeta de trabajo `lab-03` necesitamos descargar localmente algunos paquetes con lo siguiente:

Listing 3: Instalación de paquetes requeridos en este laboratorio

```
$ npm install express fs path
```

### 4.3. Contenido de los archivos creados

Como se mencionó anteriormente el archivo HTML contiene todos los archivos necesarios para el cliente, como se ve a continuación:

Listing 4: Código del HTML CSS y Javascript del cliente

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Agenda Personal</title>
8   <style>
9     body {
10       font-family: Arial, sans-serif;
```

```
11     margin: 20px;
12 }
13
14 h1 {
15     margin-bottom: 20px;
16 }
17
18 #button-section {
19     margin-bottom: 20px;
20 }
21
22 #content {
23     margin-top: 20px;
24     border: 1px solid #ccc;
25     padding: 10px;
26 }
27 </style>
28 </head>
29 <body>
30     <h1>Agenda Personal</h1>
31     <div id="button-section">
32         <button onclick="createEvent()">Crear evento</button>
33         <button onclick="editEvent()">Editar evento</button>
34         <button onclick="deleteEvent()">Eliminar evento</button>
35         <button onclick="viewEvents()">Ver eventos</button>
36     </div>
37     <div id="content"></div>
38
39     <script>
40         function createEvent() {
41             var contentDiv = document.getElementById("content");
42             contentDiv.innerHTML = '';
43
44             var textarea = document.createElement("textarea");
45             contentDiv.appendChild(textarea);
46
47             var saveButton = document.createElement("button");
48             saveButton.innerText = "Guardar";
49             saveButton.onclick = function () {
50                 var note = textarea.value;
51                 saveEvent(note);
52             };
53             contentDiv.appendChild(saveButton);
54         }
55
56         function editEvent() {
57             var contentDiv = document.getElementById("content");
58             contentDiv.innerHTML = '';
59
60             var filenameInput = document.createElement("input");
61             filenameInput.type = "text";
62             contentDiv.appendChild(filenameInput);
63
64             var editButton = document.createElement("button");
65             editButton.innerText = "Editar";
66             editButton.onclick = function () {
```

```
67         var filename = filenameInput.value;
68         loadEventContent(filename);
69     };
70     contentDiv.appendChild(editButton);
71
72     var textarea = document.createElement("textarea");
73     textarea.value = "Contenido del archivo";
74     contentDiv.appendChild(textarea);
75
76     var updateButton = document.createElement("button");
77     updateButton.innerText = "Actualizar";
78     updateButton.onclick = function () {
79         var filename = filenameInput.value;
80         var content = textarea.value;
81         updateEvent(filename, content);
82     };
83     contentDiv.appendChild(updateButton);
84 }
85
86 function deleteEvent() {
87     var contentDiv = document.getElementById("content");
88     contentDiv.innerHTML = '';
89
90     var filenameInput = document.createElement("input");
91     filenameInput.type = "text";
92     contentDiv.appendChild(filenameInput);
93
94     var deleteButton = document.createElement("button");
95     deleteButton.innerText = "Eliminar";
96     deleteButton.onclick = function () {
97         var filename = filenameInput.value;
98         deleteEvent(filename);
99     };
100    contentDiv.appendChild(deleteButton);
101 }
102
103 function viewEvents() {
104     var contentDiv = document.getElementById("content");
105     contentDiv.innerHTML = '';
106     getEventList();
107 }
108
109 function saveEvent(note) {
110     fetch('/events', {
111         method: 'POST',
112         headers: {
113             'Content-Type': 'application/json'
114         },
115         body: JSON.stringify({ note: note })
116     })
117     .then(function (response) {
118         if (response.ok) {
119             console.log("Evento guardado correctamente");
120         } else {
121             console.error("Error al guardar el evento");
122         }
123     })
124 }
```

```
123     })
124     .catch(function (error) {
125         console.error("Error de red:", error);
126     });
127 }
128
129 function loadEventContent(filename) {
130     // Obtener el contenido del evento del servidor y mostrarlo en el textarea
131     fetch('/events/${filename}', {
132         method: 'GET'
133     })
134     .then(function (response) {
135         if (response.ok) {
136             return response.text();
137         } else {
138             console.error("Error al obtener el contenido del evento");
139         }
140     })
141     .then(function (content) {
142         var textarea = document.querySelector("#content textarea");
143         textarea.value = content;
144     })
145     .catch(function (error) {
146         console.error("Error de red:", error);
147     });
148 }
149
150 function updateEvent(filename, content) {
151     fetch('/events/${filename}', {
152         method: 'PUT',
153         headers: {
154             'Content-Type': 'application/json'
155         },
156         body: JSON.stringify({ content: content })
157     })
158     .then(function (response) {
159         if (response.ok) {
160             console.log("Evento actualizado correctamente");
161         } else {
162             console.error("Error al actualizar el evento");
163         }
164     })
165     .catch(function (error) {
166         console.error("Error de red:", error);
167     });
168 }
169
170 function deleteEvent(filename) {
171     fetch('/events/${filename}', {
172         method: 'DELETE'
173     })
174     .then(function (response) {
175         if (response.ok) {
176             console.log("Evento eliminado correctamente");
177         } else {
178             console.error("Error al eliminar el evento");
```

```
179     }
180   })
181   .catch(function (error) {
182     console.error("Error de red:", error);
183   });
184 }
185
186 function getEventList() {
187   fetch('/events', {
188     method: 'GET'
189   })
190   .then(function (response) {
191     if (response.ok) {
192       return response.json();
193     } else {
194       console.error("Error al obtener la lista de eventos");
195     }
196   })
197   .then(function (eventTree) {
198     // Mostrar los eventos en forma de arbol
199     console.log(eventTree);
200   })
201   .catch(function (error) {
202     console.error("Error de red:", error);
203   });
204 }
205 </script>
206 </body>
207 </html>
```

Ahora se presenta el archivo javascript que corre en el servidor llamado app.js:

Listing 5: Código javascript del lado del servidor

```
1  const express = require('express');
2  const fs = require('fs');
3  const path = require('path');
4
5  const app = express();
6  app.use(express.urlencoded({ extended: true }));
7  app.use(express.json());
8
9  const eventsDir = path.join(__dirname, 'agenda');
10
11  // Obtener la lista de eventos
12  app.get('/events', (req, res) => {
13    fs.readdir(eventsDir, (err, files) => {
14      if (err) {
15        console.error('Error al leer el directorio de eventos:', err);
16        return res.status(500).json({ error: 'Error al obtener la lista de eventos' });
17      }
18
19      const eventTree = createEventTree(files);
20      res.json(eventTree);
21    });
22  });
23
```

```
24 // Funcion para crear el arbol de eventos
25 const createEventTree = (files) => {
26   const eventTree = [];
27
28   files.forEach(file => {
29     const filePath = path.join(eventsDir, file);
30     const stats = fs.statSync(filePath);
31
32     if (stats.isDirectory()) {
33       const eventDate = file;
34       const eventFiles = fs.readdirSync(filePath);
35       const eventChildren = [];
36
37       eventFiles.forEach(eventFile => {
38         const eventTitle = eventFile;
39         eventChildren.push({ title: eventTitle });
40       });
41
42       eventTree.push({ title: eventDate, children: eventChildren });
43     }
44   });
45
46   return eventTree;
47 };
48
49 // Crear un evento
50 app.post('/events', (req, res) => {
51   const { date, time, title, description } = req.body;
52   const eventDir = path.join(eventsDir, date);
53
54   if (!fs.existsSync(eventDir)) {
55     fs.mkdirSync(eventDir);
56   }
57
58   const eventFile = path.join(eventDir, `${time}.txt`);
59
60   if (fs.existsSync(eventFile)) {
61     res.status(400).json({ error: 'El evento ya existe' });
62   } else {
63     const eventData = `Titulo: ${title}\nDescripcion: ${description}\n`;
64     fs.writeFileSync(eventFile, eventData);
65     res.sendStatus(200);
66   }
67 });
68
69 // Editar un evento
70 app.put('/events/:date/:time', (req, res) => {
71   const { date, time } = req.params;
72   const { title, description } = req.body;
73   const eventFile = path.join(eventsDir, date, `${time}.txt`);
74
75   if (fs.existsSync(eventFile)) {
76     const eventData = `Titulo: ${title}\nDescripcion: ${description}\n`;
77     fs.writeFileSync(eventFile, eventData);
78     res.sendStatus(200);
79   } else {
```



```
80     res.status(404).json({ error: 'El evento no existe' });
81   }
82 });
83
84 // Eliminar un evento
85 app.delete('/events/:filename', (req, res) => {
86   const { filename } = req.params;
87   const eventFile = path.join(eventsDir, filename);
88
89   if (fs.existsSync(eventFile)) {
90     fs.unlinkSync(eventFile);
91     res.sendStatus(200);
92   } else {
93     res.status(404).json({ error: 'El evento no existe' });
94   }
95 });
96
97 // Ruta principal
98 app.get('/', (req, res) => {
99   res.sendFile(path.join(__dirname, 'index.html'));
100 });
101
102 // Iniciar el servidor
103 app.listen(3000, () => {
104   console.log('Servidor iniciado en http://localhost:3000');
105 });
```

Recordar además que, para correr el archivo `app.js` desde la terminal, se debe ejecutar lo siguiente:

Listing 6: Correr `app.js`

```
$ node app.js
```

Luego colocar lo siguiente en el buscador: `http://localhost:3000`

## 4.4. Commits

Como se trata de un proyecto en el que se trabajó hace algún tiempo y los commits se mezclan en la carpeta `log`, por lo que no podemos hacer `$ git log` simplemente. Para solucionar ello podemos hacer lo siguiente (ubicandonos en la carpeta `root`):

Listing 7: Ver commits los commits que se trabajaron en carpeta en específica

```
$ git log -- lab-03/
commit 86fd8bb7edb2ae8bca1f45f6fa69fc90f9a1db1d (HEAD -> main, origin/main, origin/HEAD)
Author: Diego Nina <dninas@unsa.edu.pe>
Date:   Fri Jun 9 10:08:49 2023 -0500

    solving trouble

commit 394e66d9c6455cf89636aa04c6010567167fa360
Author: Diego Nina <dninas@unsa.edu.pe>
Date:   Sun May 28 15:01:07 2023 -0500

    aniadido de funciones

commit e901aeb693685e2a27b442f407946a6255a742d
```

Author: Diego Nina <dninas@unsa.edu.pe>

Date: Sun May 28 12:37:22 2023 -0500

Mejorada la estructura de arbol en el cliente de los archivos

commit b0743f2f6082acf1820f023fee3558855886a465

Author: Diego Nina <dninas@unsa.edu.pe>

Date: Sun May 28 12:23:00 2023 -0500

lab03 - agenda

commit c5692e49b1a2b2b300ee8cd52ab3010bd9e0c9e3

Author: Gabriela <gmartellv@unsa.edu.pe>

Date: Mon May 15 15:36:06 2023 -0500

avance de clase

## 4.5. Estructura de laboratorio 02

Para poder visualizar la estructura del proyecto utilizaremos `$ tree` desde la carpeta `lab-03`

Listing 8: estructura del laboratorio

```
$ tree -I LaTeX
.
|-- agenda
|-- app.js
|-- index.html
|-- node_modules
|-- package.json
+-- package-lock.json

3 directories, 4 files
```

## 5. Rúbricas

Tabla 1: Rúbrica para contenido del Informe y demostración

	Contenido y demostración	Pts.	CL.	Est.	Prof.
<b>1. GitHub</b>	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
<b>2. Commits</b>	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	4	
<b>3. Código fuente</b>	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
<b>4. Ejecución</b>	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X		
<b>5. Pregunta</b>	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	0	
<b>6. Fechas</b>	Las fechas de modificación del código fuente estan dentro de los plazos de fecha de entrega establecidos.	2			
<b>7. Ortografía</b>	El documento no muestra errores ortográficos.	2	X	2	
<b>8. Madurez</b>	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	2	
<b>Total</b>		20		11	

## 6. Referencias

- <https://www.w3schools.com/>
- <https://www.geeksforgeeks.org/>
- <https://stackoverflow.com>