

Informe de Laboratorio 03

Tema: NodeJS + Express

Nota

Estudiante	Escuela	Asignatura
Miguel Angel Alvarez Choque malvarezcho	Escuela Profesional de Ingeniería de Sistemas	Programación Web II Semestre: I Código: 20230477

Laboratorio	Tema	Duración
04	NodeJS + Express	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 22 de mayo 2024	Al 24 de mayo 2024

1. Actividades

- Cree una aplicación NodeJS con express, para administrar una agenda personal.
- Home (“/”) : Página Principal
- Trabaje todo en una misma interfaz.
- Ejemplo de estructura de la agenda cuando se explora “Eventos”.

Listing 1: ejemplo

```
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.
- Utilice DockerFile para realizar operaciones automatizadas en Docker (incluido arrancar el servidor web nginx a través de un puerto y copiar el proyecto web para acceder desde la máquina anfitrión.)
- Producción acceder a la aplicación NodeJS+Express a través de un servidor web robusto (Nginx).
- Ejemplo: `http://127.0.0.1:8084/lab04/`
- URL del Repositorio GitHub para clonar o recuperar.
- `https://github.com/miguelnodjan/pw2_24a.git`
- URL para el laboratorio 3 en el Repositorio GitHub.
- `https://github.com/miguelnodjan/pw2_24a/tree/main/lab_04`

2. Actividad:

2.1. creación de archivos *html*, *css*:

- Para esta parte se creó un archivo *html* base que se irá adaptando según avancemos con el laboratorio.
- Adicional a ello, se necesita de un *css* para que de estilo al *html*.

Agenda de Eventos

Crear Evento

Título:

Descripción:

Fecha:

Hora:

Guardar Evento

Eventos

Listing 2: css

```
* {  
  body {  
    font-family: Arial, sans-serif;  
    background-color: #f4f4f4;  
    margin: 0;  
    padding: 0;  
  }  
  
  .agenda-container {  
    max-width: 800px;  
    margin: 20px auto;  
    padding: 20px;  
    background-color: #fff;  
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
  }  
  
  header {  
    text-align: center;  
    margin-bottom: 20px;  
  }  
  
  header h1 {  
    margin: 0;  
    font-size: 2em;  
    color: #333;  
  }  
  
  main {  
    display: flex;  
    flex-direction: column;  
    gap: 15px;  
  }  
  
  form {  
    display: flex;  
    flex-direction: column;  
    gap: 10px;  
  }  
  
  form label {  
    font-weight: bold;  
  }  
  
  form input, form textarea, form button {  
    padding: 10px;  
    font-size: 1em;  
    margin-top: 5px;  
  }  
  
  #events-list ul {  
    list-style-type: none;  
    padding: 0;  
  }  
  
  #events-list li {
```

```
padding: 10px;
border: 1px solid #ddd;
border-radius: 5px;
background-color: #fafafa;
margin-bottom: 5px;
}
```

2.2. Inicializar *Node.js* y *Express*:

- Como lo pide el laboratorio, se procede a inicializar un proyecto de *Node.js*
- El archivo resultante es el siguiente que tienen como funciones:
- Se prosigue a compartir una captura del momento de inicialización:

```
PS D:\UNSA\Repositorios\Pweb2\pw2_24a\lab_04> npm init -y
Wrote to D:\UNSA\Repositorios\Pweb2\pw2_24a\lab_04\package.json:

{
  "name": "lab_04",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}

PS D:\UNSA\Repositorios\Pweb2\pw2_24a\lab_04> npm install express body-parser
added 64 packages, and audited 65 packages in 3s

12 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\UNSA\Repositorios\Pweb2\pw2_24a\lab_04> |
```

2.3. creando archivo *JavaScript*

- Se añadió el código al *js* para que tenga funcionalidad nuestra agenda

Listing 3: scripts.js

```
const express = require('express');
const bodyParser = require('body-parser');
const fs = require('fs');
const path = require('path');

const app = express();
```

```
const PORT = 3001;

const baseDir = path.join(__dirname, 'agenda');

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static(path.join(__dirname, '.')));

if (!fs.existsSync(baseDir)) {
  fs.mkdirSync(baseDir);
}

app.get('/events', (req, res) => {
  const events = [];

  fs.readdirSync(baseDir).forEach(dateDir => {
    const datePath = path.join(baseDir, dateDir);
    if (fs.statSync(datePath).isDirectory()) {
      fs.readdirSync(datePath).forEach(file => {
        const filePath = path.join(datePath, file);
        const [hour, minute] = file.replace('.txt', '').split('-');
        const content = fs.readFileSync(filePath, 'utf8').split('\n');
        const title = content[0];
        const description = content.slice(1).join('\n');
        events.push({
          title,
          description,
          date: dateDir,
          time: `${hour}:${minute}`
        });
      });
    }
  });

  res.json(events);
});

app.post('/events', (req, res) => {
  const { title, description, date, time } = req.body;
  const [hour, minute] = time.split(':');
  const dateDir = path.join(baseDir, date);
  const eventFilePath = path.join(dateDir, `${hour}-${minute}.txt`);

  if (!fs.existsSync(dateDir)) {
    fs.mkdirSync(dateDir);
  }

  if (fs.existsSync(eventFilePath)) {
    return res.status(400).json({ message: 'el evento ya existe' });
  }

  const eventContent = `${title}\n${description}`;
  fs.writeFileSync(eventFilePath, eventContent);

  res.status(201).json({ title, description, date, time });
});
```

```
});

app.put('/events/:date/:time', (req, res) => {
  const { date, time } = req.params;
  const { title, description } = req.body;
  const [hour, minute] = time.split(':');
  const eventFilePath = path.join(baseDir, date, `${hour}-${minute}.txt`);

  if (!fs.existsSync(eventFilePath)) {
    return res.status(404).json({ message: 'No se encontr el evento' });
  }

  const eventContent = `${title}\n${description}`;
  fs.writeFileSync(eventFilePath, eventContent);

  res.json({ title, description, date, time });
});

app.delete('/events/:date/:time', (req, res) => {
  const { date, time } = req.params;
  const [hour, minute] = time.split(':');
  const eventFilePath = path.join(baseDir, date, `${hour}-${minute}.txt`);

  if (!fs.existsSync(eventFilePath)) {
    return res.status(404).json({ message: 'No se encontr el evento' });
  }

  fs.unlinkSync(eventFilePath);
  res.status(204).end();
});

app.listen(PORT, () => {
  console.log(` http://localhost:${PORT}`);
});
```

Descripción:

Fecha:

Hora:

Guardar Evento

Eventos
Examen

2024-05-24 - 18:58

importante examen

Editar

Eliminar








Evaluación

2024-06-07 - 21:04

examen parcial

Editar

Eliminar

<div>  > Este equipo > Nuevo vol (D:) > UNSA > Repositorios > Pweb2 > pw2_24a > lab_04 > agenda > </div>				
<div>     Ordenar Ver ... </div>				
Nombre	Fecha de modificación	Tipo	Tamaño	
 2024-05-24	24/05/2024 15:58	Carpeta de archivos		
 2024-06-07	24/05/2024 15:59	Carpeta de archivos		

2.4. Acceder a la aplicación NodeJS+Express a través de un servidor web robusto (Nginx).

- Se realizaron una serie de comandos y operaciones para la creación de la imagen base y luego establecer nuestra agenda como un contenedor para poder acceder:

Listing 4: procedimiento

```
const express = require('express');
PS D:\UNSA\Repositorios\Pweb2\pw2_24a\lab_04> docker pull node:alpine
alpine: Pulling from library/node
d25f557d7f31: Pull complete
64975b5da3f3: Pull complete
346f957dbc37: Pull complete
fda0c25af7a5: Pull complete
Digest: sha256:e07a6a6c965e052163ece23eda2e4505c44b46dcf76a106320f7f8b40d7827b0
Status: Downloaded newer image for node:alpine
docker.io/library/node:alpine

What's Next?
View a summary of image vulnerabilities and recommendations  docker scout quickview
node:alpine
PS D:\UNSA\Repositorios\Pweb2\pw2_24a\lab_04> docker run -it --name agenda-container -v
${PWD}:/app -w /app node:alpine /bin/sh
/app #
/app #
/app # npm install

up to date, audited 65 packages in 1s

12 packages are looking for funding
run 'npm fund' for details

found 0 vulnerabilities
/app # node index.js
http://localhost:3001

PS D:\UNSA\Repositorios\Pweb2\pw2_24a\lab_04> docker run -p 3001:3001 -v ${PWD}:/app -w
/app node:alpine sh -c "npm install && node index.js"

up to date, audited 65 packages in 936ms

12 packages are looking for funding
run 'npm fund' for details

found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 10.7.0 -> 10.8.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.0
npm notice To update run: npm install -g npm@10.8.0
npm notice
http://localhost:3001
```

- Las evidencias se proceden a compartir:

Container Name	ID	Image	Status	Ports	Uptime	Actions
agenda-container	4e2a2cd877ae	node:alpine	Exited (137)		0% 3 minutes ago	Stop, Restart, Logs
awesome_franklin	171b594278f6	node:alpine	Running	3001:3001	0% 26 seconds ago	Stop, Restart, Logs

```
PS D:\UNSA\Repositorios\Pweb2\pw2_24a\lab_04> docker run -p 3001:3001 -v ${PWD}:/app -w /app node:alpine sh -c "npm install && node index.js"
up to date, audited 65 packages in 936ms
12 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
npm notice
npm notice New minor version of npm available! 10.7.0 -> 10.8.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.0
npm notice To update run: npm install -g npm@10.8.0
npm notice
npm notice http://localhost:3001
```

2.5. Commits:

- A continuación se muestra una captura de pantalla de los principales *commits* para la creación de la página web

```
añadiendo funcionalidad con JavaScript
migueldn committed 18 minutes ago c4ea5a1

inicializamos Node.js y Express
migueldn committed 1 hour ago b44d99d

mejorando el html del lab 4
migueldn committed 1 hour ago 9b739d7

añadiendo la primera version de la agenda con el html y css respectivo
migueldn committed 1 hour ago 1f34fd5

añadiendo la
migueldn committed 1 hour ago 9d41d2e

Commits on May 22, 2024

creando archivos para laboratorio 4
migueldn committed 2 days ago 5f8488d

Commits on May 17, 2024
```

2.6. Cuestionario

- Mencione la diferencia entre conexiones asíncronas usando el objeto XMLHttpRequest, JQuery.ajax y Fetch. Justifique su respuesta con un ejemplo muy básico. Eje: Hola Mundo, IMC, etc
 - **XMLHttpRequest** es el objeto nativo de *JavaScript* para solicitudes http, muy compatible pero que requiere de mayor configuración para manejar las respuestas o errores que sucedan.
 - **JQuery.ajax** proviene de la biblioteca *jQuery*, tiene una sintaxis sencilla y fácil
 - **Fetch** es la interfaz moderna de *JavaScript*, posee la sintaxis más sencilla, limpia y fácil de usar de los tres tipos de objetos para conexión asíncrona.
 - Se prosigue a compartir diferentes versiones de los tres objetos para la obtención de contenido de archivos de texto:

Listing 5: XMLHttpRequest

```
const xhr = new XMLHttpRequest();
```

```
xhr.onreadystatechange = function() {  
  if (xhr.readyState === XMLHttpRequest.DONE) {  
    if (xhr.status === 200) {  
      console.log(xhr.responseText); // Manejar la respuesta  
    } else {  
      console.error('Error:', xhr.status); // Manejar errores  
    }  
  }  
};  
xhr.open('GET', 'example.txt', true);  
xhr.send();
```

Listing 6: jQuery.ajax:

```
$.ajax({  
  url: 'example.txt',  
  method: 'GET',  
  success: function(response) {  
    console.log(response); // Manejar la respuesta  
  },  
  error: function(xhr, status, error) {  
    console.error('Error:', status); // Manejar errores  
  }  
});
```

Listing 7: Fetch API:

```
fetch('example.txt')  
  .then(response => {  
    if (!response.ok) {  
      throw new Error('Network response was not ok');  
    }  
    return response.text();  
  })  
  .then(data => {  
    console.log(data); // Manejar la respuesta  
  })  
  .catch(error => {  
    console.error('Error:', error); // Manejar errores  
  });
```

2.7. Estructura de laboratorio 2

```
lab_4  
|----- index.html  
|-----index.js  
|-----package-lock.json  
|-----package.json  
|  
|-----agenda  
|       |-----2024-05-24  
|       |       |----- 18-58.txt  
|       |  
|       |
```

```
|      |-----2024-06-07
|      |----- 21-04.txt
|
|-----css
|      |----- style.css
|
|-----ejercicios
|      |----- index.html
|      |----- index.js
|
|-----latex
|-----node_modules
```

3. Rúbricas

3.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe		
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y fácil de leer.	Nota
Observaciones	Respetar la estructura de organización para la ubicación de los entregables. Por cada observación dentro del informe se le descontará puntos. Se debe incluir el código fuente latex del informe	

3.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

Nivel				
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	1	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	1	
8. Madurez	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	3	
Total		20		14	

4. Bibliografía:

- <http://mally.stanford.edu/~sr/computing/basic-unix.html>
- <https://www.technology.pitt.edu/help-desk/how-to-documents/basic-unix-commands>