



UNIVERSIDAD NACIONAL DE SAN AGUSTIN
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y
SERVICIOS ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMA-



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 1

INFORME DE LABORATORIO

(formato estudiante)

INFORMACION BASICA

| | | | | | |
|--|--------------------|------------------------|-------------|----------------|------------------------------|
| ASIGNATURA: | Programacion web 2 | | | | |
| TITULO DE LA PRACTICA: | NodeJS + Express | | | | |
| NÚMERO DE PRÁCTICA: | 4 | AÑO LECTIVO: | 2024 | NRO. SEMESTRE: | 3 |
| FECHA DE PRE-SENTACIÓN: | 2024/5/25 | HORA DE PRE-SENTACIÓN: | 7:40 | | |
| INTEGRANTE (s): HUAMANI CONDORI JEANPIERO SIXTO | | | NOTA(0-20): | | Nota colocada por el docente |
| DOCENTE(s): LINO JOSE PINTO OPPE | | | | | |

RESULTADOS Y PRUEBAS



I. EJERCICIOS RESUELTOS:

Descripción 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” 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 traves de un puerto y copiar el proyecto web para acceder desde la máquina anfitrión.) Producción acceder a la aplicación NodeJS+Express a traves de un servidor web robusto (Nginx). Ejemplo: <http://127.0.0.1:8084/lab04/>

codigo express

Commit

crear , eliminar editar evento

 main **jhuamaniCond** committed 53 minutes ago

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

const app = express();

app.use(cors());
app.use(bodyParser.json()); // Para parsear JSON
app.use(bodyParser.urlencoded({ extended: true })); // Para parsear URL-encoded
app.use(express.static('pub'));

app.listen(3000, () => {
  console.log("Escuchando en: http://localhost:3000");
});

app.get('/', (request, response) => {
  response.sendFile(path.resolve(__dirname, 'index.html'));
});

app.get('/agenda', (request, response) => {
  // Obtén Los parámetros de fecha y hora de la consulta
  const { date, time } = request.query;

  // Construye la ruta del archivo utilizando Los parámetros
  const filePath = path.resolve(__dirname, `agenda/${date}/${time}.txt`);

  fs.readFile(filePath, 'utf8', (err, data) => {
    if (err) {
      console.error(err);
      response.status(500).json({
        error: 'message'
      });
      return;
    }
    response.json({
      text: data.replace(/\n/g, '<br>')
    });
  });
});

app.post('/crear-agenda', (request, response) => {
  const { fecha, hora, contenido } = request.body;

  if (!fecha || !hora || !contenido) {
    return response.status(400).json({ error: 'Fecha, hora y contenido son requeridos' });
  }

  const dirPath = path.resolve(__dirname, `agenda/${fecha}`);
  const filePath = path.join(dirPath, `${hora}.txt`);
```

codigo para enviar peticiones

Commit

js para crear el formulario y realizar diversas acciones

main

jhuamaniCond committed 2 hours ago

```
function obtenerAgenda() {  
  // Construye la URL con los parámetros de consulta  
  const url = `http://localhost:3000/agenda?date=${encodeURIComponent(fecha)}&time=${encodeURIComponent(hora)}`;  
  
  fetch(url)  
    .then(response => response.json())  
    .then(data => {  
      datos=data.text.replace(/<br>/g, '\n')  
      const titulo = datos.split("\n")[0];  
      const contenido = datos.substring(datos.indexOf("\n") + 1);  
      console.log("titulo"+titulo+"content"+contenido)  
      editarEventoForm2(fecha,hora,titulo,contenido)  
      document.querySelector("#respuesta").innerHTML = 'encontrado';  
    })  
    .catch(error => {  
      document.querySelector("#respuesta").innerHTML = 'NO se encontro';  
      console.error('Error fetching the agenda:', error);  
    });  
}  
  
function creaEvento() {  
  const fecha = document.querySelector("#fecha").value;  
  const hora = document.querySelector("#hora").value;  
  let contenido = document.querySelector("#contenido").value;  
  const titulo = document.querySelector("#titulo").value;  
  contenido=titulo+"\n"+contenido  
  
  if (!fecha || !hora || !contenido || !titulo) {  
    document.querySelector("#respuesta").innerHTML = "Todos los campos son requeridos.";   
    return;  
  }  
  console.log("fecha"+fecha+"hora"+hora+"cont"+contenido)  
  fetch('http://localhost:3000/crear-agenda', {  
    method: 'POST',  
    headers: {  
      'Content-Type': 'application/json'  
    },  
    body: JSON.stringify({ fecha, hora, contenido })  
  })  
    .then(response => response.json())  
    .then(data => {  
      document.querySelector("#respuesta").innerHTML = data.message || data.error;  
    })  
    .catch(error => {  
      console.error('Error:', error);  
      document.querySelector("#respuesta").innerHTML = 'Ocurrió un error al crear el archivo.';  
    });  
}
```

II. PRUEBAS

Bienvenido al editor de eventos

- Crear evento
- Editar evento
- Eliminar evento
- Ver eventos

Bienvenido al editor de eventos

- Crear evento
- Editar evento
- Eliminar evento
- Ver eventos

Archivo eliminado con éxito



III. CUESTIONARIO:

Mencione la diferencia entre conexiones asíncronas usando el objeto XMLHttpRequest, JQuery.ajax

Objeto XMLHttpRequest:

Sintaxis:

Requiere la creación de una instancia del objeto XMLHttpRequest utilizando el constructor new XMLHttpRequest().

Luego, se configuran los parámetros de la solicitud, como el método HTTP y la URL, utilizando métodos y propiedades del objeto.

Manejo de eventos:

Para manejar los eventos relacionados con la solicitud (como onreadystatechange, onload, onerror, etc.), se deben asignar funciones de devolución de llamada a estas propiedades del objeto.

Compatibilidad con todos los navegadores:

XMLHttpRequest es compatible con la mayoría de los navegadores modernos, incluidos Internet Explorer 7+ y versiones posteriores. Funcionalidad básica:

Proporciona funcionalidades básicas para realizar solicitudes HTTP asíncronas, pero no incluye características avanzadas como manejo de promesas o métodos de abstracción de datos.

JQuery.ajax:

Sintaxis:

Utiliza la función \$.ajax() de JQuery para realizar solicitudes HTTP asíncronas. La sintaxis es más simple y legible en comparación con XMLHttpRequest, ya que no requiere crear una instancia de objeto. Manejo de eventos:

JQuery.ajax utiliza Deferred objects o Promises para manejar eventos de éxito, error y completado. Esto hace que el código sea más estructurado y fácil de manejar. Encapsulamiento de la lógica:

Proporciona métodos de abstracción de datos, como \$.get(), \$.post(), \$.getJSON(), etc., que simplifican la realización de solicitudes HTTP comunes. Compatibilidad con todos los navegadores:

JQuery.ajax garantiza una mayor compatibilidad con navegadores antiguos y nuevos, ya que maneja automáticamente las diferencias en la implementación de XMLHttpRequest entre los navegadores.



CONCLUSIONES

Para las conclusiones tenemos que express , nos ayuda a crear puntos endpoint para poder recibir solicitudes , y a traves de estas solicitudes hacer alguna accion como devolver informacion o hacer alguna operacion .Con fetch enviamos dichas solicitudes al endpoint

METODOLOGÍA DE TRABAJO

Para la metodologia utilizada , fue primero analizar el ejercicio , luego primero crear los end-point donde se realizaran las llamadas del fetch , y luego la implementacion en general

```
jhuamanicond/  
|-- pw2-24a  
|   |-- lab04  
|   |   |-- agenda  
|   |   |   |-- 2024-5-25  
|   |   |   |   |-- 10-00.txt  
|   |   |   |   |-- 10-30.txt  
|   |   |   |-- 2024-12-1  
|   |   |   |   |-- 12-30.txt  
|   |   |-- estilos.css  
|   |   |-- index.html  
|   |   |-- index.js  
|   |   |-- manejadorEventos.js
```



| | Contenido y demostración | Puntos | Checklist | Estudiante | Profesor |
|---------------|---|--------|-----------|------------|----------|
| 1. GitHub | Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error o omisión) | 4 | ✓ | 4 | |
| 2. Commits | Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación). | 4 | ✓ | 2 | |
| 3. Ejecución | Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión) | 4 | ✓ | 2 | |
| 4. Pregunta | Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación). | 2 | ✓ | 2 | |
| 7. Ortografía | El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado) | 2 | ✓ | 1 | |
| 8. Madurez | El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación). | 4 | ✓ | 3 | |
| | Total | | | 14 | |

REFERENCIAS



UNIVERSIDAD NACIONAL DE SAN AGUSTIN
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y
SERVICIOS ESCUELA PROFESIONAL DE
INGENIERÍA DE SISTEMA-



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01

Código: GUIA-PRLE-001

Página: 9

<https://github.com/rescobedoq/backend-js>

<https://medium.com/zero-equals-false/using-cors-in-express-cac7e29b005b>

https://www.w3schools.com/nodejs/nodejs_intro.asp

<https://nodejs.org/en/docs/guides/getting-started-guide>

https://www.w3schools.com/js/js_api_fetch.asp

<https://expressjs.com/es/>

https://developer.mozilla.org/es/docs/Web/API/Fetch_API/Using_Fetch

https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction

<https://nodejs.org/docs/latest-v18.x/api/>