

Informe de Laboratorio 4

Tema: NodeJS y AJAX

Nota

Estudiante	Escuela	Asignatura
Chambilla Perca Ricardo Mauricio rchambillap@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programacion Web 2 Semestre: III Código: 1701213

Laboratorio	Tema	Duración
4	Nodejs y AJAX	4

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 21 de Mayo 2024	Al 24 de Mayo 2024

1 Ejercicios propuestos

- 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.

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

La elaboracion de la pagina web usando el servidor nodejs y peticiones ajax esta documentado en mi repositorio publico que se encuentra en el siguiente enlace: https://github.com/rikich3/pweb2_lab.git

```
npm init
npm install express
npm install --local cors
```

1.1 Home (“/”) : Página Principal

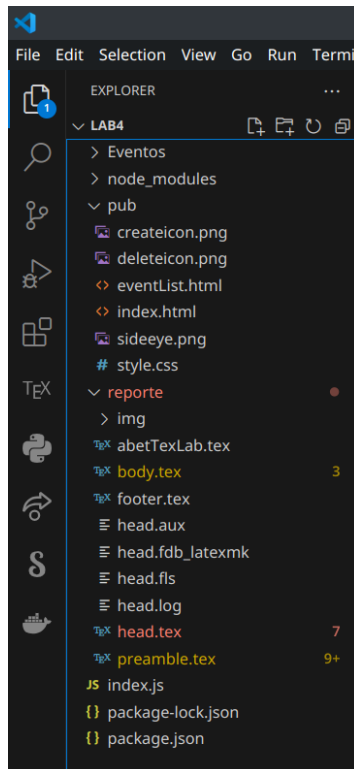
Para la mejor separación de los elementos públicos y carpetas privadas del servidor se está sirviendo desde una carpeta de elementos estáticos.

```

1      app.use(express.static('pub'));
2      \dots
3      app.get('/', (request, response) => {
4          response.sendFile(path.resolve(__dirname, 'pub', 'index.html'));
5      });

```

El directorio principal termina viéndose así:



1.2 Trabaja todo en una misma interfaz.

Toda la aplicación web estará en una sola página web, en index.html dentro de la carpeta pub. El layout se ve así:

Mi agenda

Últimas 2 operaciones realizadas

Crea un evento!

Date (YYYY-MM-DD):
mm / dd / yyyy

Hour (HH-MM):
-- : -- --

Título:

Descripción:

Elimina un evento

Date (YYYY-MM-DD):
mm / dd / yyyy

Hour (HH-MM):
-- : -- --

Los forms son para definir los eventos a crear o eliminar y los botones en la parte de abajo para realizar estas acciones.

1.3 PREGUNTA: 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.

Fetch API es la opción más moderna y preferida para realizar solicitudes asíncronas en el navegador debido a su sintaxis simple y su uso de promesas. JQuery.ajax() sigue siendo útil en proyectos que ya utilizan JQuery, pero para nuevos proyectos, Fetch API es la opción recomendada. El objeto XMLHttpRequest es la opción más antigua y estándar, pero su uso directo es menos común debido a la disponibilidad de alternativas más modernas y convenientes.

2 Rúbricas

2.1 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.

2.2 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

Puntos	Nivel			
	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	2	
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	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas.	4	X	4	
Total		20		17	

3 Referencias

- https://github.com/rikich3/pweb2_lab.git
-