
	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 1</p>

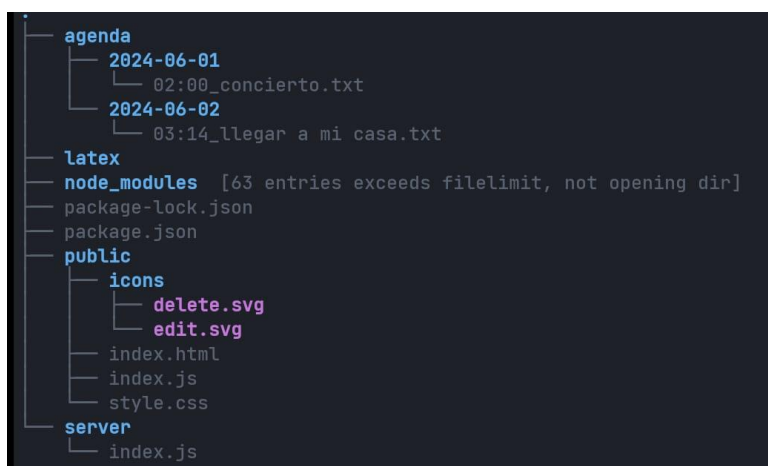
INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	<i>Laboratorio pweb</i>				
TÍTULO DE LA PRÁCTICA:	NodeJS + Express				
NÚMERO DE PRÁCTICA:	<i>04</i>	AÑO LECTIVO:	<i>Segundo</i>	NRO. SEMESTRE:	<i>Primero</i>
FECHA DE PRESENTACIÓN	<i>25/05/2024</i>	HORA DE PRESENTACIÓN	-		
INTEGRANTE (s) <i>Carbajal Gonzales Diego Alejandro</i>				NOTA (0-20)	
DOCENTE(s):					

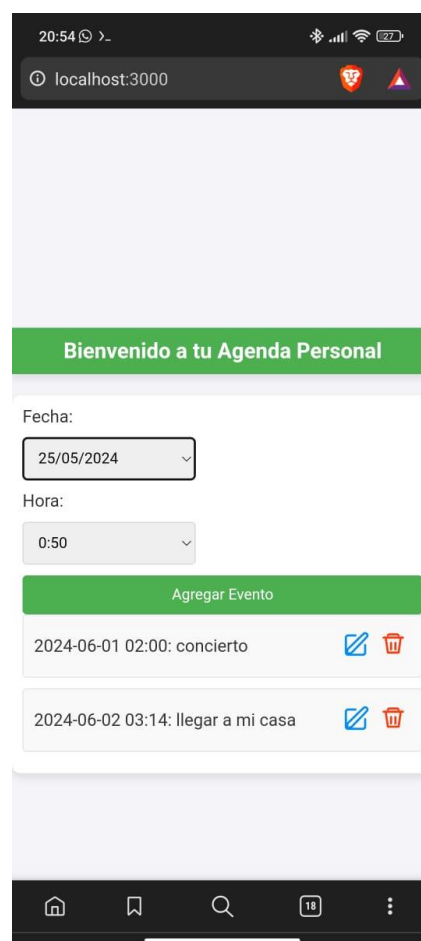
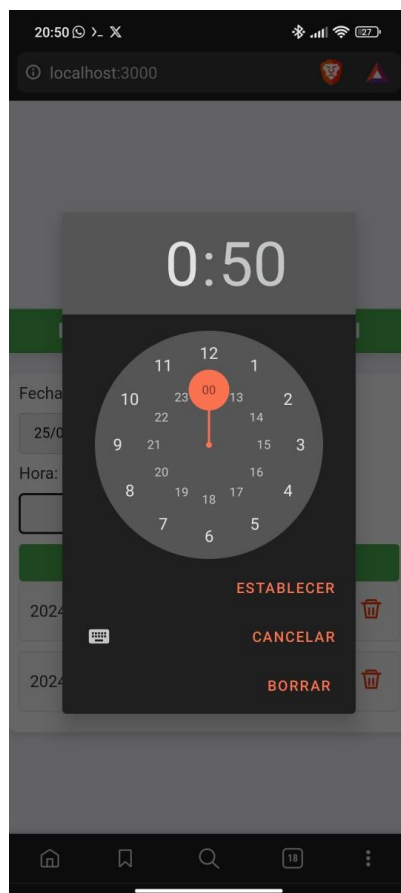
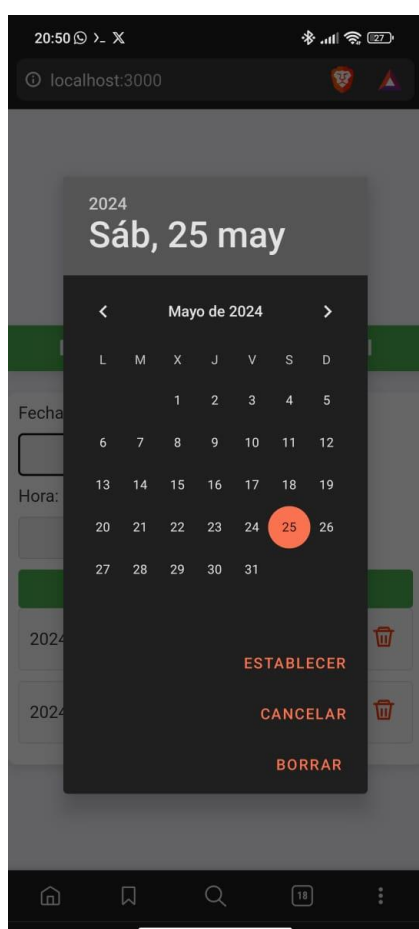
RESULTADOS Y PRUEBAS
<p>I. EJERCICIOS RESUELTOS:</p> <ul style="list-style-type: none"> ○ Cree una aplicación NodeJS con express, para administrar una agenda personal. ○ Home ("/") : Página Principal ○ Seguir el ejemplo de estructura dado en el documento ○ Trabaje todo en una misma interfaz. ○ La aplicación debe permitir: <ul style="list-style-type: none"> ● 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 <p>Toda el proceso de elaboración del laboratorio esta documentado en : https://github.com/L0rD1ego/Tareas_pweb.git</p>



II. PRUEBAS

Se siguió la jerarquía de archivos:



La aplicación cumple con los requerimientos:



	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 3

III. CUESTIONARIO:

Las conexiones asíncronas usando el objeto XMLHttpRequest, jQuery.ajax, y Fetch difieren en la forma en que se gestionan y se interactúa con ellas.

XMLHttpRequest (XHR): Es un objeto nativo de JavaScript que permite realizar solicitudes HTTP asíncronas. Requiere más código para configurar y manejar, ya que se debe crear una instancia del objeto XMLHttpRequest y configurar manualmente los eventos.

jQuery.ajax: jQuery es una biblioteca de JavaScript que simplifica la manipulación del DOM y la interacción con el servidor. jQuery.ajax es un método que simplifica la realización de solicitudes HTTP asíncronas. Proporciona una sintaxis más sencilla y opciones configurables para manejar las solicitudes y respuestas.



Fetch: Es una API moderna de JavaScript para realizar solicitudes HTTP asíncronas. Utiliza promesas para manejar las respuestas de manera más eficiente y proporciona una sintaxis clara y concisa para realizar solicitudes. Es la opción más moderna y preferida para realizar conexiones asíncronas en aplicaciones web modernas.

CONCLUSIONES

En esta práctica, hemos explorado diferentes enfoques para realizar conexiones asíncronas en aplicaciones web utilizando `XMLHttpRequest`, `jQuery.ajax`, y `Fetch`. Hemos visto que cada opción tiene sus propias características y ventajas. `XMLHttpRequest` es la opción más antigua y requiere más código para configurar y manejar, mientras que `jQuery.ajax` proporciona una sintaxis más sencilla pero aún depende de la biblioteca jQuery. Por otro lado, `Fetch` es la API más moderna y eficiente, ofreciendo una sintaxis clara y concisa, además de ser compatible con promesas. En conclusión, la elección entre estas opciones depende de las necesidades específicas del proyecto y del entorno en el que se esté trabajando, pero `Fetch` emerge como la opción preferida para aplicaciones web modernas debido a su simplicidad y eficiencia.

METODOLOGÍA DE TRABAJO

Se trabajo paso a paso estructurando los archivos y mejorando continuamente

	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 4

REFERENCIAS Y BIBLIOGRAFÍA

<https://nodejs.org/en>

Contenido y demostracion		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 onservación)	4	X	3	
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
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	X	2	
4. 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. (Se descontará puntos por error encontrado)	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados a partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	4	
TOTAL		20		16	