Logotipo, nombre de la empresa

Descripción generada automáticamente

FACULTAD DE TELEMÁTICA

ING. EN TECNOLOGÍAS DE INTERNET

INGENIERIA DE SOFTWARE

LISTA DE PENDIENTES

**UTILIZANDO NODE JS**

Dr. MIGUEL ÁNGEL AGUAYO ORTUÑO

EQUIPO:5

* ESCALIER VILLALVA BETHEL ITZEL
* ROJAS TORRES JESSICA JUDITH

GRUPO: 4°B

**INTRODUCCIÓN**

En la actualidad, el uso de aplicaciones web ha aumentado debido al creciente número de usuarios conectados a la red. Esto ha requerido que las redes de Internet sean más amplias y tengan la capacidad de ofrecer un servicio de calidad a todos los clientes. La programación de sistemas distribuidos de servicios de Internet es una solución para poder gestionar el tráfico de datos. Además, las páginas web deben ser aplicaciones escalables, de alta calidad y altamente eficientes para manejar grandes cantidades de solicitudes simultáneamente.

**DESARROLLO**

1. El sistema debe funcionar en un servidor y ser accesible por IP desde las computadoras del laboratorio. Para este paso, creamos un archivo en JavaScript llamado "leer.js" en el que declaramos el puerto que utilizaremos y la IP donde se encuentra el servidor.
2. El sistema debe tener una caja de texto donde puedas agregar nuevas tareas por hacer y un botón para agregarlas. En un archivo HTML y con ayuda de CSS, creamos la caja de texto y el botón para agregar las tareas del usuario.
3. El sistema debe mostrar las tareas que has agregado. Dentro del mismo archivo, hemos colocado un script que contiene el código JS que muestra en pantalla las tareas creadas.
4. El sistema debe tener un botón para remover las tareas que has agregado y deben tener alguna manera de ser seleccionables (en el ejemplo se hace con una checkbox). El código compara los valores ingresados en cada ciclo "if", donde los valores según sea el caso, ejecutan la acción de "remover" o de "agregar".
5. El sistema debe mostrar las tareas que marques como realizadas. Con ayuda de CSS, creamos las checkbox que indican gráficamente qué tareas ya han sido realizadas.
6. El sistema debe mostrar un mensaje de alerta si la caja de texto donde agregar las tareas se encuentra vacía. El código compara el valor esperado con el valor asignado por el usuario y si este es falso, aparecerá un alert que indica que la caja de texto está vacía.
7. El sistema debe mostrar un mensaje de alerta si se agrega una tarea idéntica a una que ya existe. El código compara el valor ingresado con los valores del arreglo formado por las tareas previamente ingresadas y si alguna es igual al valor ingresado, entonces se activará un alert que indica que la tarea ya existe.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**CONCLUSIONES**

En conclusión, el desarrollo de un sistema distribuido que maneje una lista de pendientes en Internet utilizando JavaScript, Node.JS y la librería Express requiere conocimientos de programación, el funcionamiento de servidores, el manejo de APIs y protocolos de comunicación, así como el dominio de las herramientas necesarias para implementar el sistema. Además, es importante contar con habilidades de diseño para crear una interfaz de usuario amigable y funcional, así como habilidades de resolución de problemas y depuración para garantizar que el sistema funcione correctamente y sin errores.

Gracias a esta práctica, se ha logrado crear una página web que genera una lista de pendientes y puede ser utilizada por varios usuarios al mismo tiempo, desde diferentes dispositivos, simplemente con la dirección IP del servidor. Esto demuestra la capacidad de aplicar los conocimientos y habilidades mencionados para desarrollar un sistema distribuido funcional y práctico para el manejo de tareas en línea.

Para mayor información consultar el siguiente link, donde se encuentra el portafolio de GitHub con el código:

<https://github.com/JessicaRojas24/toDoList.git>