Camioneta azul en la calle

El contenido generado por IA puede ser incorrecto.

Tutor de la empresa:

Roberto Pascual Ucero

PHARMA-CO(BOT)-LOGY

Realizado por:

Gabriel Brown Ahumada

Gregorio López Salvador

En este documento os vamos a explicar la programación que lleva este proyecto de una manera menos detallada como en otros documentos

## PROGRAMACIÓN DE VISUAL STUDIO CODE:

**Código final - Interfaz 1 - Interfaz 2 - Interfaz 3 - Interfaz 4 – Estilos – Interfaz5 – Código médico**

-En el documento principal se ha explicado las interfaces de la página web y sus estilos, aquí se va a explicar los códigos realizados con ‘JavaScript.’

Códigos en JavaScript que son 2, el código final (app.js) y el Código médico.

-app.js

app.js

/\*

Programa:

Funcionamiento

 · Recibe un JSON de Node-Red via WebSocket, con una estructura, como por ejemplo: {"tipo":"nuevo\_usuario","nombre":"Tito","medicamentos":[{"id":3,"uds":2,"dosis":"500mg","nombre":"Paracetamol"},{"id":4,"uds":1,"dosis":"600mg","nombre":"Ibuprofeno"}]}

 · Dicho JSON: "data" la guardamos en "usuario"

 · Si recibimos en data.tipo "nuevo\_usuario" cargamos medicamentos.html, si no mostramos mensaje en le DOM de index.html de "Usuario no reconocido"

 · Si, en el caso de que estemos en medicamentos.html, esperemos a que esté cagado el DOM y cogemos de "usuario.nombre" el nombreUsuario para mostrarlo en el HTML

 · Se prepara la tabla mediante el método .forEach

 · Si de da a aceptar btnAceptar', entonces se pasar la acción de "aceptar" y mapea con el método .map para enviarlo mediante WS a node-red y se pasa a confirmacion.html

 · En caso de cancelar btnCancelar, entonces se vuelve a index.html

 · En confirmacion.html se hace una cuenta atrás para mostrar una alarma

 \*/

// Función para abrir la llamada en Microsoft Teams

document.addEventListener("DOMContentLoaded", () => {

    // Asegurar que el botón de ayuda existe en la página

    const helpButton = document.querySelector(".help-button");

    if (helpButton) {

        helpButton.addEventListener("click", () => {

            openTeamsCall("g.lopez@padrearamburuburgoso365.educamos.com");

        });

    }

});

function openTeamsCall(targetEmail) {

    window.open(

        `https://teams.microsoft.com/l/call/0/0?users=g.lopez@padrearamburuburgoso365.educamos.com`,

        "\_blank",

        "width=800,height=600"

    );

}

// Establecer conexión WebSocket con Node-RED al entrar en este script\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

var ws = new WebSocket("ws://" + location.hostname + ":1880/ws/farmacia");

var ws2 = new WebSocket("ws://" + location.hostname + ":1880/ws/farmaciaPLC");

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Función para navegar a la página que se indica en cada caso\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

function navegar(pagina) {

    window.location.href = pagina;  //Carga el parámetro "pagina" al que le pasamos como argumento index.html, medicamentos.html...

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Manejar mensajes entrantes del WebSocket\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ws.onmessage = function(event) {

    const data = JSON.parse(event.data);  //La información recibida de node-red, se parsea a JSON y se asigna a el objeto "data"

    if(data.tipo === 'nuevo\_usuario') {      //Si en "data.tipo" se ha obtenido "nuevo\_usuario..."

        sessionStorage.setItem('usuario', JSON.stringify(data)); //...se almacenan los datos recibidos (data) de manera temporal en la keyword usuario

        navegar('medicamentos.html'); //Y se navega al html "medicamentos.html"

    }

    else{

            document.body.insertAdjacentHTML('beforeend', '<div class="container\_aviso"> <h1>Usuario no reconocido</h1></div>');

            ws.send(JSON.stringify({ accion: 'no\_reconocido' }));

    }

};

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// En medicamentos.html\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (window.location.pathname.endsWith('medicamentos.html')) {

    document.addEventListener('DOMContentLoaded', () => {

        const usuario = JSON.parse(sessionStorage.getItem('usuario'));

        if (!usuario) {

            alert("No se ha encontrado el usuario en sessionStorage.");

            return;

        }

        document.getElementById('nombreUsuario').textContent = usuario.nombre;

        const tbody = document.querySelector('#tablaMedicamentos tbody');

        if (usuario.medicamentos && usuario.medicamentos.length > 0) {

            usuario.medicamentos.forEach(med => {

                const row = document.createElement('tr');

                // Celda del checkbox

                const checkboxCell = document.createElement('td');

                const checkbox = document.createElement('input');

                checkbox.type = 'checkbox';

                checkbox.value = med.id;

                checkbox.dataset.nombre = med.nombre;

                checkbox.dataset.uds = med.uds;

                checkboxCell.appendChild(checkbox);

                row.appendChild(checkboxCell);

                // Celdas del medicamento, dosis y unidades

                row.innerHTML += `

                    <td>${med.nombre}</td>

                    <td>${med.dosis}</td>

                `;

                // Celda de unidades con botones + y -

                const udsCell = document.createElement('td');

                const btnMenos = document.createElement('button');

                btnMenos.textContent = '-';

                btnMenos.addEventListener('click', () => {

                    if (med.uds > 0) {

                        med.uds--;

                        udsDisplay.textContent = med.uds;

                    }

                });

                const udsDisplay = document.createElement('span');

                udsDisplay.textContent = med.uds;

                const btnMas = document.createElement('button');

                btnMas.textContent = '+';

                // Almacenar el valor máximo original de uds

                const maxUds = usuario.medicamentos.reduce((max, m) => {

                    if (m.nombre === med.nombre) {

                        return Math.max(max, m.uds);

                    }

                    return max;

                }, 0);

                btnMas.addEventListener('click', () => {

                    if (med.uds < maxUds) { // Limitar al máximo original de uds

                        med.uds++;

                        udsDisplay.textContent = med.uds;

                    }

                });

                udsCell.appendChild(btnMenos);

                udsCell.appendChild(udsDisplay);

                udsCell.appendChild(btnMas);

                row.appendChild(udsCell);

                tbody.appendChild(row);

            });

            // Marcar los checkboxes después de que se hayan renderizado

            setTimeout(() => {

                document.querySelectorAll('#tablaMedicamentos input[type="checkbox"]').forEach(checkbox => {

                    checkbox.checked = true;

                });

            }, 0);

        } else {

            tbody.innerHTML = '<tr><td colspan="4">No hay medicamentos asignados.</td></tr>';

        }

        document.getElementById('btnCancelar').addEventListener('click', () => {

            ws.send({ accion: 'cancelar' });

            navegar('index.html');

        });

        document.getElementById('btnAceptar').addEventListener('click', () => {

            const seleccionados = Array.from(document.querySelectorAll('input[type="checkbox"]:checked'));

            if (seleccionados.length === 0) {

                alert("Por favor, selecciona al menos un medicamento.");

                return;

            }

            const medicamentosSeleccionados = seleccionados.map(checkbox => {

                // Encuentra el objeto med correspondiente en usuario.medicamentos

                const med = usuario.medicamentos.find(m => m.id === parseInt(checkbox.value));

                return {

                    id: checkbox.value,

                    nom: checkbox.dataset.nombre,

                    uds: med.uds // Usa el valor actualizado de med.uds

                };

            });

            ws.send(JSON.stringify({

                accion: 'aceptar',

                medicamentos: medicamentosSeleccionados

            }));

            navegar('robot.html');

        });

    });

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// En robot.html\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (window.location.pathname.endsWith('robot.html')) {

    document.addEventListener('DOMContentLoaded', () => {

        ws2.onmessage = function(event) {

            const data2 = JSON.parse(event.data);  //La información recibida de node-red, se parsea a JSON y se asigna a el objeto "data"

            if(data2.estado === 'robot\_fin') {      //Si en "data.tipo" se ha obtenido "nuevo\_usuario..."

                navegar('recoger.html'); //Y se navega al html "medicamentos.html"

            }

        };

    });

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// En recoger.html\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (window.location.pathname.endsWith('recoger.html')) {

    document.addEventListener('DOMContentLoaded', () => {

        const audio = new Audio('snd/alerta.mp3');

        audio.loop = true; // Configura el bucle para el audio

        reproducirSonido();

        // Configurar un temporizador de 30 segundos para redirigir si no se ha recogido el pedido

        const timeout = setTimeout(() => {

            console.log("⏳ Tiempo agotado. Redirigiendo a index.html...");

            audio.pause(); // Detener el sonido

            navegar('index.html'); // Volver a la página inicial después de 30 segundos

        }, 30000); // 30 segundos (30000 milisegundos)

        ws2.onmessage = function(event) {

            const data2 = JSON.parse(event.data); // Parsear la información recibida como JSON

            if (data2.estado === 'pedido\_recogido' || data2.estado === 'pedido\_no\_recogido') {

                clearTimeout(timeout); // Si se recibe el mensaje, cancelamos el temporizador

                console.log("✅ Pedido recogido o no recogido. Redirigiendo a index.html...");

                navegar('index.html'); // Volver a la página inicial inmediatamente

                audio.pause();         // Detener el audio

            }

        };

        function reproducirSonido() {

            const playPromise = audio.play();

            if (playPromise !== undefined) {

                playPromise.catch(error => console.error("⚠️ Error al reproducir el audio:", error));

            }

        }

    });

}

**Las funciones que realiza este código son las siguientes:**

-Realiza una conexión WebSocket con Node-RED

-Recepción de un usuario nuevo con medicamentos

-Muestra medicamentos y puede seleccionar

-Acepta o cancela

-Activa robot

-Espera recogida

-medico.js

medico.js

let socket = new WebSocket("ws://" + location.hostname + ":1880/ws/farmacia");

let pacientes = JSON.parse(localStorage.getItem("pacientes")) || [];

function guardarPacientes() {

    localStorage.setItem("pacientes", JSON.stringify(pacientes));

}

function filtrarPacientes() {

    let inputNombre = document.getElementById("nombrePaciente");

    let filtro = inputNombre.value.toLowerCase();

    let listaSugerencias = document.getElementById("sugerencias");

    listaSugerencias.innerHTML = "";

    if (filtro.length === 0) {

        return;

    }

    let resultados = pacientes.filter(p => p.nombre.toLowerCase().includes(filtro));

    resultados.forEach(paciente => {

        let opcion = document.createElement("div");

        opcion.textContent = paciente.nombre;

        opcion.onclick = function () {

            inputNombre.value = paciente.nombre;

            document.getElementById("pacienteId").value = paciente.id;

            listaSugerencias.innerHTML = "";

            cargarMedicamentos(paciente.id);

        };

        listaSugerencias.appendChild(opcion);

    });

}

function cargarMedicamentos(pacienteId) {

    let paciente = pacientes.find(p => p.id === pacienteId);

    let medicamentosDiv = document.getElementById("medicamentos");

    medicamentosDiv.innerHTML = "";

    if (paciente && paciente.medicamentos) {

        paciente.medicamentos.forEach(med => {

            agregarMedicamento(med.nombre, med.uds, med.dosis);

        });

    } else {

        // Agregar un campo de medicamento vacío si no hay medicamentos para el paciente

        agregarMedicamento();

    }

}

function obtenerMedicamentos() {

    return new Promise((resolve, reject) => {

        socket.send(JSON.stringify({ action: "obtenerMedicamentos" }));

        socket.onmessage = function (event) {

            try {

                let data = JSON.parse(event.data);

                if (data.action === "medicamentos") {

                    console.log("Medicamentos recibidos:", data.medicamentos);

                    resolve(data.medicamentos);

                }

            } catch (error) {

                console.error("Error al analizar la respuesta de medicamentos:", error);

                reject(error);

            }

        };

        socket.onerror = function (error) {

            console.error("Error en la conexión WebSocket:", error);

            reject(error);

        };

    });

}

async function agregarMedicamento(nombre = "", uds = "", dosis = "") {

    let medicamentosDiv = document.getElementById("medicamentos");

    let medicamentoDiv = document.createElement("div");

    medicamentoDiv.innerHTML = `

        <input type="text" class="nombre" placeholder="Nombre del medicamento" value="${nombre}">

        <input type="number" class="uds" placeholder="Unidades" value="${uds}">

        <input type="text" class="dosis" placeholder="Dosis" value="${dosis}">

        <button onclick="this.parentNode.remove()">Eliminar</button>

        <br><br>

    `;

    medicamentosDiv.appendChild(medicamentoDiv);

}

function guardarMedicamentos() {

    let pacienteId = document.getElementById("pacienteId").value;

    let medicamentosDiv = document.getElementById("medicamentos").children;

    let nuevosMedicamentos = [];

    for (let medicamentoDiv of medicamentosDiv) {

        let nombre = medicamentoDiv.querySelector(".nombre").value;

        let uds = medicamentoDiv.querySelector(".uds").value;

        let dosis = medicamentoDiv.querySelector(".dosis").value;

        if (nombre && uds && dosis) {

            nuevosMedicamentos.push({ nombre, uds: Number(uds), dosis });

        }

    }

    let paciente = pacientes.find(p => p.id === pacienteId);

    if (paciente) {

        paciente.medicamentos = nuevosMedicamentos;

        guardarPacientes();

        alert("Medicamentos guardados con éxito");

        // Enviar datos a Node-RED a través de WebSocket

        socket.send(JSON.stringify({

            action: "guardarMedicamentos",

            pacienteId: pacienteId,

            medicamentos: nuevosMedicamentos

        }));

        cargarMedicamentos(pacienteId);

    }

}

pacientes = [

    { id: 1, nombre: "Gabriel Brown Ahumada", medicamentos: [] },

    { id: 2, nombre: "Gregorio Lopez Salvador", medicamentos: [] },

    { id: 3, nombre: "Pablo Navalon Diaz", medicamentos: [] }

];

guardarPacientes();

// Actualizar el desplegable de medicamentos al recibir datos del WebSocket

socket.onmessage = function (event) {

    try {

        let data = JSON.parse(event.data);

        if (data.action === "medicamentos") {

            console.log("Medicamentos recibidos:", data.medicamentos);

            // No necesitas actualizar el desplegable aquí, ya que los medicamentos se agregan directamente como campos de entrada

        }

    } catch (error) {

        console.error("Error al analizar la respuesta del WebSocket:", error);

    }

};

**Las funciones que realiza este código son las siguientes:**

-Busca pacientes desde una lista guardada localmente.

-Carga medicamentos asignados a ese paciente.

-Edita o agrega medicamentos.

-Guarda cambios enviando los datos por WebSocket a un servidor (tipo Node-RED).

-Guarda pacientes localmente en el navegador con localStorage.