Java script: SoloWaifus

* Paso 1:
* **Código:**

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

* **Explicación:**

Espera que todo el contenido HTML esté completamente cargado antes de ejecutar el script.

* Paso 2:
* **Código:**

const form = document.getElementById("formulario");

const tablaBody = document.getElementById("tablaBody");

* **Explicación:**

Obtiene referencias al formulario con id formulario y al cuerpo de la tabla donde se mostrarán los datos (tablaBody).

* Paso 3:
* **Código:**

function cargarDatos() {

* **Explicación:**

Declara una función que carga los datos guardados desde localStorage y los muestra en la tabla.

* Paso 4:
* **Código:**

const datos = JSON.parse(localStorage.getItem("reclamos")) || [];

* **Explicación:**

Obtiene y convierte desde JSON los reclamos guardados en localStorage. Si no hay datos, se usa un arreglo vacío.

* Paso 5:
* **Código:**

tablaBody.innerHTML = "";

* **Explicación:**

Limpia el contenido actual de la tabla para actualizarla con nuevos datos

* Paso 6:
* **Código:**

datos.forEach((dato, index) => {

* **Explicación:**

Itera sobre cada objeto de datos almacenados (cada reclamo).

* Paso 7:
* **Código:**

const fila = document.createElement("tr");

* **Explicación:**

Crea una nueva fila en la tabla para ese reclamo.

* Paso 8:
* **Código:**

Object.values(dato).forEach(valor => {

const celda = document.createElement("td");

celda.textContent = valor;

fila.appendChild(celda);

});

* **Explicación:**

Recorre cada valor del objeto dato y lo agrega como una celda a la fila.

* Paso 9:
* **Código:**

const celdaAcciones = document.createElement("td");

* **Explicación:**

Crea una celda adicional para los botones de editar y eliminar.

* Paso 10:
* **Código:**

const btnEditar = document.createElement("button");

btnEditar.textContent = "Editar";

btnEditar.type = "button";

* **Explicación:**

Crea un botón “Editar”.

* Paso 11:
* **Código:**

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

* **Explicación:**

Define lo que pasa al hacer clic en "Editar".

* Paso 12:
* **Código:**

if (btnEditar.textContent === "Editar") {

* **Explicación:**

Si está en modo "Editar", cambia el contenido de las celdas a campos <input> para que el usuario pueda modificarlos.

* Paso 13:
* **Código:**

const input = document.createElement("input");

input.value = fila.children[i].textContent;

* **Explicación:**

Crea inputs con los valores actuales de las celdas para editar.

* Paso 14:
* **Código:**

btnEditar.textContent = "Guardar";

* **Explicación:**

Cambia el texto del botón a "Guardar".

* Paso 15:
* **Código:**

} else {

const datosEditados = JSON.parse(localStorage.getItem("reclamos")) || [];

* **Explicación:**

Si está en modo "Guardar", toma los nuevos valores desde los inputs y Vuelve a cargar los datos desde localStorage.

* Paso 16:
* **Código:**

for (let i = 0; i < fila.children.length - 1; i++) {

const nuevoValor = fila.children[i].firstChild.value.trim();

* **Explicación:**

Obtiene el nuevo valor de cada campo editable.

* Paso 17:
* **Código:**

if (i !== 4 && nuevoValor === "") {

alert("Por favor, no dejes campos obligatorios vacíos.");

return;

}

* **Explicación:**

Valida que los campos (excepto el email) no estén vacíos.

* Paso 18:
* **Código:**

fila.children[i].textContent = nuevoValor;

datosEditados[index][Object.keys(datosEditados[index])[i]] = nuevoValor;

* **Explicación:**

Actualiza tanto el DOM como el objeto original con los nuevos valores.

* Paso 19:
* **Código:**

localStorage.setItem("reclamos", JSON.stringify(datosEditados));

btnEditar.textContent = "Editar";

cargarDatos();

* **Explicación:**

Guarda los cambios en localStorage, cambia el botón a "Editar" y recarga la tabla.

* Paso 20:
* **Código:**

const btnEliminar = document.createElement("button");

btnEliminar.textContent = "Eliminar";

btnEliminar.type = "button";

* **Explicación:**

Crea un botón "Eliminar".

* Paso 21:
* **Código:**

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

* **Explicación:**

Define lo que pasa al hacer clic en "Eliminar".

* Paso 22:
* **Código:**

const datosEliminar = JSON.parse(localStorage.getItem("reclamos")) || [];

datosEliminar.splice(index, 1);

* **Explicación:**

Elimina el reclamo correspondiente por su índice.

* Paso 23:
* **Código:**

localStorage.setItem("reclamos", JSON.stringify(datosEliminar));

cargarDatos();

* **Explicación:**

Guarda la lista actualizada sin el reclamo eliminado y recarga la tabla.

* Paso 24:
* **Código:**

celdaAcciones.appendChild(btnEditar);

celdaAcciones.appendChild(btnEliminar);

fila.appendChild(celdaAcciones);

tablaBody.appendChild(fila);

* **Explicación:**

Agrega los botones a la fila, y luego la fila a la tabla.

* Paso 25:
* **Código:**

form.addEventListener("submit", e => {

e.preventDefault();

* **Explicación:**

Evita que el formulario recargue la página al enviarse.

* Paso 26:
* **Código:**

const nuevoDato = {

nombre: form.firstName.value.trim(),

apellido: form.lastName.value.trim(),

pais: form.address.value.trim(),

ciudad: form.City.value.trim(),

email: form.email.value.trim(),

motivo: form.Motivo.value.trim()

};

* **Explicación:**

Crea un nuevo objeto con los datos ingresados por el usuario.

* Paso 27:
* **Código:**

if (!nuevoDato.nombre || !nuevoDato.apellido || !nuevoDato.pais || !nuevoDato.ciudad || !nuevoDato.motivo) {

alert("Por favor completa todos los campos obligatorios.");

return;

}

* **Explicación:**

Valida que los campos obligatorios no estén vacíos.

* Paso 28:
* **Código:**

if (form["same-address"].checked === false || form["save-info"].checked === false) {

alert("Debes confirmar y autorizar el uso de tus datos.");

return;

}

* **Explicación:**

Verifica que el usuario haya marcado los checkboxes requeridos.

* Paso 29:
* **Código:**

if (nuevoDato.email && !/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(nuevoDato.email)) {

alert("Por favor ingresa un correo electrónico válido.");

return;

}

* **Explicación:**

Valida el formato del email si fue ingresado.

* Paso 30:
* **Código:**

const datos = JSON.parse(localStorage.getItem("reclamos")) || [];

datos.push(nuevoDato);

localStorage.setItem("reclamos", JSON.stringify(datos));

form.reset();

cargarDatos();

* **Explicación:**

Guarda el nuevo reclamo en localStorage, limpia el formulario y actualiza la tabla.

* Paso 31:
* **Código:**

cargarDatos();

* **Explicación:**

Al cargar la página, muestra los datos previamente guardados.

**fila.children[i]:** Estás accediendo al i-ésimo hijo (elemento <td>) de la fila <tr> creada en la tabla. Es decir, .children es como un arreglo que contiene los elementos <td> de esa fila.