

# Proyecto Unidad III



**Nombre:** Andre Alexander Hidrogo Rocha

**Numero de control:** 20490746

**Materia:** Programación Web

**Carrera:** Ing. Sistemas Computacionales

**Fecha:** 19/10/2023

**Lugar:** Mexicali Baja California

# Índice

Introducción.....	2
Desarrollo .....	3
Agregar un login .....	3
Catálogo de productos (CRUD) .....	3
Tienda.....	8
Acerca de.....	9
Agregar la calculadora.....	9
Accesos de usuarios.....	9
Conclusión.....	10

# Introducción

Este reporte tiene como objetivo dar a conocer cómo pude realizar las diferentes características indicadas para el proyecto de la unidad III utilizando HTML, JavaScript y Bootstrap.

Además, permitirá resaltar los puntos más importantes del código del proyecto y como es que este funciona.

# Desarrollo

## Agregar un login

Para realizar el login simplemente utilice inputs y valide las entradas de los mismos para verificar si eran un cliente o un administrador.

```
document.addEventListener('DOMContentLoaded', function() {
  const username = document.getElementById('username');
  const password = document.getElementById('password');
  const loginButton = document.getElementById('loginButton');

  loginButton.addEventListener('click', function() {
    const enteredUsername = username.value;
    const enteredPassword = password.value;

    if (enteredUsername === 'admin' && enteredPassword === '123') {
      localStorage.setItem('userRole', 'admin'); // Almacena el rol en
localStorage
      window.location.href = 'admin.html'; // Redireccionar al panel de
administrador
    } else if (enteredUsername === 'cliente' && enteredPassword === '123')
{
      localStorage.setItem('userRole', 'cliente'); // Almacena el rol
en localStorage
      window.location.href = 'cliente.html'; // Redireccionar al panel
de cliente
    } else {
      alert('Credenciales incorrectas. Por favor, intenta de nuevo.');
```

Menú con las pestanas

## Catálogo de productos (CRUD)

Lo hice utilizando el local storage para obtener los datos de los productos de la tienda y después mostrarlos con el mismo formato que en la tienda.

```

document.addEventListener("DOMContentLoaded", function () {
    const catalogo = JSON.parse(localStorage.getItem("catalogo")) || [];

    const catalogoContainer = document.getElementById("catalogo");
    const botonAgregarProducto =
document.getElementById("botonAgregarProducto");

    catalogo.forEach((producto) => {
        const card = document.createElement("div");
        card.classList.add("col-sm-6", "col-md-4", "col-lg-3", "mb-4");
        card.innerHTML = `
            <div class="card">
                
                <div class="card-body">
                    <div class="col">
                        <input type="text" class="card-input"
id="inputCatalogo3" placeholder="Ingrese nueva Imagen"></input>
                        <h3 class="card-
title">${producto.nombre}  ${producto.id}</h3>
                        <input type="text" class="card-input"
id="inputCatalogo1" placeholder="Ingrese una descripcion"></input>
                        <input type="text" class="card-input"
id="inputCatalogo4" placeholder="Ingrese nuevo ID"></input>
                        <p class="card-text">${producto.precio}</p>
                        <input type="text" class="card-input"
id="inputCatalogo2" placeholder="Ingrese nuevo precio"></input>
                    </div>
                    <div class="col">
                        <button class="btn btn-custom botonModificar"
id="botonModificar">Modificar</button>
                        <button class="btn btn-primary botonEliminar"
id="botonEliminar">Eliminar</button>
                    </div>
                </div>
            </div>
        `;
        catalogoContainer.appendChild(card);
    });

```

Agregue el botón eliminar, el cual toma el id del producto y su index en el array para eliminar el producto.

```

// Agrega un evento de clic al botón eliminar
const botonEliminar = card.querySelector(".botonEliminar");

```

```

        botonEliminar.addEventListener("click", function () {
            const productoAEliminar = catalogo.find(item => item.id ===
producto.id);
            const indexAEliminar = catalogo.indexOf(productoAEliminar);

            // Verifica si se encontró el producto en el array
            if (indexAEliminar !== -1) {
                // Elimina el producto del array
                catalogo.splice(indexAEliminar, 1);

                // Elimina la tarjeta de la interfaz de usuario
                card.remove();

                // Actualiza el localStorage con el catálogo modificado
                localStorage.setItem("catalogo", JSON.stringify(catalogo));
            }
        });
    });
});

```

Agregue el botón modificar el cual verifica si los inputs son visibles y si lo son toma los datos de los mismos y los asigna como el nuevo valor del producto modificado.

```

// Agrega un evento de clic al botón Modificar
const botonModificar = card.querySelector(".botonModificar");
botonModificar.addEventListener("click", function () {
    const inputCatalogo1 = card.querySelector("#inputCatalogo1");
    const inputCatalogo2 = card.querySelector("#inputCatalogo2");
    const inputCatalogo3 = card.querySelector("#inputCatalogo3");
    const inputCatalogo4 = card.querySelector("#inputCatalogo4");
    const visible = inputCatalogo1.style.display === "block";

    if (visible) {

        // Obtiene los nuevos valores de los inputs
        const nuevoID = inputCatalogo1.value;
        const nuevoPrecio = inputCatalogo2.value;
        const nuevaImagen = inputCatalogo3.value;
        const nuevoNombre = inputCatalogo4.value;

        // Verifica que los campos no estén vacíos antes de
actualizar
    }
});

```

```

        if (nuevoID && nuevoPrecio && nuevaImagen && nuevoNombre) {
            // Encuentra el producto en el catálogo actual
            const productoAModificar = catalogo.find(item => item.id
=== producto.id);

            // Actualiza los valores del producto con los nuevos
valores

            productoAModificar.id = nuevoID;
            productoAModificar.precio = nuevoPrecio;
            productoAModificar.imagen = nuevaImagen;
            productoAModificar.nombre = nuevoNombre;

            // Actualiza la información de la tarjeta con los nuevos
valores

            card.querySelector(".card-title").textContent =
nuevoNombre+" "+nuevoID;
            card.querySelector(".card-text").textContent =
`$$${nuevoPrecio}`;
            card.querySelector("#imagenCatalogo").src = nuevaImagen;

            // Actualiza el local storage con el catálogo modificado
localStorage.setItem("catalogo",
JSON.stringify(catalogo));

            inputCatalogo1.style.display = "none";
            inputCatalogo2.style.display = "none";
            inputCatalogo3.style.display = "none";
            inputCatalogo4.style.display = "none";
        } else {
            alert("Por favor, complete todos los campos antes de
modificar el producto.");
        }
    }
    else {
        inputCatalogo1.style.display = "block";
        inputCatalogo2.style.display = "block";
        inputCatalogo3.style.display = "block";
        inputCatalogo4.style.display = "block";
    }
});

```

Por último, esta la función de agregar producto la cual es similar a modificar, ya que toma valores de los inputs para asignarlos al nuevo producto, la diferencia es que agrega estos datos al array de productos y crea una carta para el nuevo producto.

```
botonAgregarProducto.addEventListener("click", function () {
    // Obtiene los valores de los nuevos campos de entrada
    const inputCatalogoImagen =
document.getElementById("inputCatalogoImagen").value;
    const inputCatalogoID =
document.getElementById("inputCatalogoID").value;
    const inputCatalogoPrecio =
document.getElementById("inputCatalogoPrecio").value;
    const inputCatalogoNombre =
document.getElementById("inputCatalogoNom").value;

    // Verifica que los campos no estén vacíos antes de agregar el
producto
    if (inputCatalogoImagen && inputCatalogoID && inputCatalogoPrecio &&
inputCatalogoNombre) {
        // Crea un nuevo producto con los datos ingresados
        const nuevoProducto = {
            id: inputCatalogoID,
            imagen: inputCatalogoImagen,
            precio: inputCatalogoPrecio,
            nombre: inputCatalogoNombre
        };

        // Agrega el nuevo producto al catálogo
        catalogo.push(nuevoProducto);

        // Actualiza el localStorage con el catálogo modificado
        localStorage.setItem("catalogo", JSON.stringify(catalogo));

        // Crea una nueva tarjeta para el nuevo producto y agrega al
catálogoContainer
        const card = document.createElement("div");
        card.classList.add("col-sm-6", "col-md-4", "col-lg-3", "mb-4");
        card.innerHTML = `
            <div class="card">
                
                <div class="card-body">
                    <div class="col">
```



```

        <input type="text" class="card-input"
id="inputCatalogo3" placeholder="Ingrese nueva Imagen"></input>
        <h3 class="card-title">${nuevoProducto.nombre}
${nuevoProducto.id}</h3>
        <input type="text" class="card-input"
id="inputCatalogo1" placeholder="Ingrese una descripcion"></input>
        <input type="text" class="card-input"
id="inputCatalogo4" placeholder="Ingrese nuevo ID"></input>
        <p class="card-text">${nuevoProducto.precio}</p>
        <input type="text" class="card-input"
id="inputCatalogo2" placeholder="Ingrese nuevo precio"></input>
    </div>
    <div class="col">
        <button class="btn btn-custom botonModificar"
id="botonModificar">Modificar</button>
        <button class="btn btn-primary botonEliminar"
id="botonEliminar">Eliminar</button>
    </div>
</div>
`;
catalogoContainer.appendChild(card);

// Limpia los campos de entrada
document.getElementById("inputCatalogoImagen").value = "";
document.getElementById("inputCatalogoID").value = "";
document.getElementById("inputCatalogoPrecio").value = "";
document.getElementById("inputCatalogoNom").value = "";

console.log("Producto agregado correctamente");
} else {
    alert("Por favor, complete todos los campos antes de agregar el
producto.");
}
});

```

## Tienda

Lo único diferente con la tienda a comparación con la tarea 4 es que la hice responsiva utilizando Bootstrap

```
card.classList.add("col-sm-6", "col-md-4", "col-lg-3", "mb-4");
```

## Acerca de

La parte de Acerca de simplemente son párrafos y títulos.

## Agregar la calculadora

Para agregar la calculadora tuve que colocar un botón que ocultara la calculadora al aplastarlo y la mostrara al volverlo a aplastarlo.

```
document.getElementById("botonMostrarCalculadora").addEventListener("click",
function () {
    var calculator = document.getElementById("calculator");

    if (calculator.style.display === "none" || calculator.style.display ===
    "") {
        calculator.style.display = "block";
    } else {
        calculator.style.display = "none";
    }
});
```

## Accesos de usuarios

Para mostrar diferentes menús dependiendo el usuario cree diferentes HTMLS para los usuarios y para que el cliente no se pueda meter a catalogo verifique con una variable la cual se actualiza dependiendo si el cliente o el admin entra a la página.

```
document.addEventListener('DOMContentLoaded', function() {
    const userRole = localStorage.getItem('userRole');

    if (userRole !== 'admin') {
        // Redirige a una página de error o a la página de cliente si el
        usuario no es un administrador.
        window.location.href = 'cliente.html';
    }
});
```

# Conclusión

Para concluir puedo decir que este proyecto tenía bastantes obstáculos y a mi parecer no los aborde de la mejor manera, debido a esto tuve algunas complicaciones, por ejemplo, el como les di acceso a los usuarios, lo hice de esta manera ya que fue la primera que se me ocurrió y al final ya no tuve tiempo para hacerlo de una forma más optima.

Otro punto que me dio problemas fue la creación de nuevos productos, ya que funciona correctamente y el producto se agrega, el problema es que tienes que recargar la pagina para que tenga todas las funcionalidades.

Por último, me hubiera gustado experimentar un poco más con el diseño de la página, pero las ideas que tenía no me parecían muy buenas y el tiempo para tratar de implementar o pensar en mejores no me fue suficiente aun así estoy satisfecho con el producto final.