

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO PROGRAMACION 1
DOCUMENTO DE ANÁLISIS



Camilo Artola – 306498

M1H REMOTO

Docente: Alberto Zonca

Analista Programador

20-06-2024

Índice

1.	Descripción general del problema a resolver.....	4
1.1.	Tipos de usuario del sistema.....	4
1.2.	Listado de funcionalidades	4
2.	Detalle de Funcionalidades	5
2.1	F1 - Registrarse - Comprador	5
2.1.1	Acceso.....	5
2.1.2	Descripción	5
2.1.3	Interfaz de usuario.....	5
2.1.4	Validaciones.....	5
2.2	F2 - Ingreso al sistema - Comprador.....	6
2.2.1.	Acceso.....	6
2.2.2	Descripción	6
2.2.3.	Interfaz de usuario.....	6
2.2.4.	Validaciones.....	6
2.3	F3 - Compra de productos - Comprador	7
2.3.2	Acceso.....	7
2.3.3	Descripción	7
2.3.4	Interfaz de usuario.....	7
2.3.5	Validaciones.....	7
2.4	F4 - Ver listado de compras - Comprador	7
2.4.2	Acceso.....	7
2.4.3	Descripción	7
2.4.4	Interfaz de usuario.....	8
2.4.5	Validaciones.....	8
2.5	F5 - Ver productos en oferta - Comprador.....	8

2.5.2	Acceso.....	8
2.5.3	Descripción	8
2.5.4	Interfaz de usuario.....	9
2.5.5	Validaciones.....	9
2.6	F6 - Ingreso al sistema - Administrador	9
2.6.1	Acceso.....	9
2.6.2	Descripción	9
6.2.3.	Interfaz de usuario.....	10
6.2.4.	Validaciones.....	10
2.7	F7 - Listado y aprobación de las compras de los usuarios - Administrador ...	10
2.7.2	Acceso.....	10
2.7.3	Descripción	10
2.7.4	Interfaz de usuario.....	11
2.7.5	Validaciones.....	11
2.8	F8 - Crear producto - Administrador	11
2.8.2	Acceso.....	11
2.8.3	Descripción	11
2.8.4	Interfaz de usuario.....	12
2.8.5	Validaciones.....	12
2.9	F9 - Administrar productos - Administrador	12
2.9.2	Acceso.....	12
2.9.3	Descripción	12
2.9.4	Interfaz de usuario.....	13
2.9.5	Validaciones.....	13
2.10	F10 - Ver informe de ganancias - Administrador	13
2.10.2	Acceso.....	13
2.10.3	Descripción	13
2.10.4	Interfaz de usuario.....	14
2.10.5	Validaciones.....	14

1. Descripción general del problema a resolver

Se desea crear una aplicación web para un comercio que se dedica a la venta de artículos deportivos y poder vender productos de forma online a los usuarios (e-commerce).

1.1. Tipos de usuario del sistema

- Administrador
- Comprador

1.2. Listado de funcionalidades

- F1 - Registrarse - Comprador
- F2 - Ingreso al sistema - Comprador/Administrador
- F3 - Compra de productos - Comprador
- F4 - Ver listado de compras - Comprador
 - F4.1 - Pendientes - Comprador
 - F4.2 - Canceladas - Comprador
 - F4.3 - Aprobadas - Comprador
- F5 - Ver productos en oferta - Comprador
- F6 - Listado y aprobación de las compras de los usuarios - Administrador
- F7 - Crear producto - Administrador
- F8 - Administrar productos - Administrador
 - F9.1 - Modificar stock - Administrador
 - F9.2 - Modificar estado - Administrador
 - F9.3 - Asignar o desmarcar producto en oferta - Administrador
- F9 - Ver informe de ganancias - Administrador

2. Detalle de Funcionalidades

A continuación, se presenta el detalle de cada una de las funcionalidades a resolver en el obligatorio. Las funcionalidades están ordenadas por tipo de usuario.

2.1 F1 - Registrarse - Comprador

2.1.1 Acceso

Comprador

2.1.2 Descripción

El comprador completa el registro rellorando todos los datos del formulario y cumpliendo con las validaciones solicitadas. Enseguida se carga el Id auto incremental y el saldo inicial de 3000 como dinero ficticio para hacer compras.

2.1.3 Interfaz de usuario



The screenshot shows a web interface for user registration. At the top, there is a dark blue header with a shopping cart icon. Below the header, the title "Registro" is displayed. The form consists of several input fields with labels above them: "NOMBRE" (Ingrese nombre...), "APELLIDO" (Ingrese apellido...), "NOMBRE DE USUARIO" (Ingrese nombre de usuario...), "CONTRASEÑA" (Ingrese contraseña...), "NÚMERO DE TARJETA DE CRÉDITO" (Ingrese numero de tarjeta...), and "CVC (CODIGO DE VERIFICACIÓN)" (Ingrese codigo de verificación...). At the bottom of the form is a dark blue button labeled "REGISTRARSE".

2.1.4 Validaciones

- Todos los campos son obligatorios
- Validar que nombre de usuario(case insensitive) sea único
- Validar contraseña con mínimo y caracteres solicitados
- Validar numero de tarjeta de crédito con formato indicado
- Validar código de verificación con formato indicado

- En caso de no cumplir con algo, informar de los errores mediante la interfaz HTML y no realizar el registro.

2.2 F2 - Ingreso al sistema - Comprador

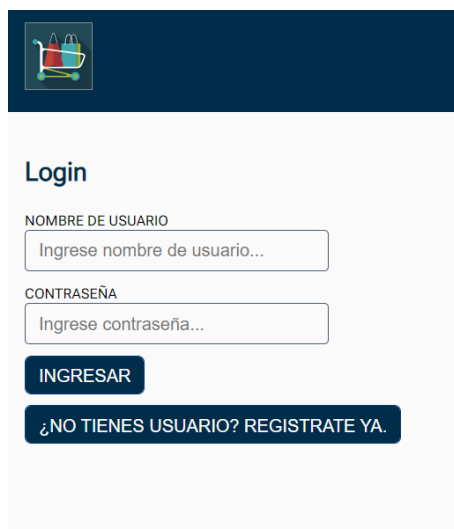
2.2.1. Acceso

Comprador

2.2.2 Descripción

Al registrarse, el comprador debería acceder al sistema realizando el login ingresando nombre de usuario y contraseña.

2.2.3. Interfaz de usuario



The screenshot shows a login form titled "Login" with a shopping cart icon in the top left. It contains two input fields: "NOMBRE DE USUARIO" with the placeholder "Ingrese nombre de usuario..." and "CONTRASEÑA" with the placeholder "Ingrese contraseña...". Below these fields are two buttons: a dark blue "INGRESAR" button and a dark blue button with white text that says "¿NO TIENES USUARIO? REGÍSTRATE YA.".

2.2.4. Validaciones

- Todos los campos obligatorios
- Validar nombre de usuario
- Validar contraseña
- En caso de no cumplir o algún dato incorrecto, informar error en interfaz HTML.

2.3 F3 - Compra de productos - Comprador


2.3.2 Acceso

Comprador

2.3.3 Descripción




El comprador puede ver los productos e indicando cuantas unidades de un producto desea, solicitar la compra, presionando el botón.

2.3.4 Interfaz de usuario



COMPRA DE PRODUCTOSLISTADO DE COMPRASPRODUCTOS EN OFERTA CERRAR SESIÓN

Compra de Productos

Nombre	Descripción	Precio por unidad. (\$U)		Imagen	Saldo: \$3000	
Pelota de Basquetbol	Pelota de Basquetbol marca Nivia, incluye inflador de regalo.	300			<input type="text" value="1"/>	COMPRAR
Botella deportiva	Botella deportiva ideal para salir a correr o ir al gimnasio.	120	EN OFERTA!		<input type="text" value="1"/>	COMPRAR
Pelota de Fútbol Americano	Pelota de Fútbol Americano marca Champion Sports tamaño oficial.	260	EN OFERTA!		<input type="text" value="1"/>	COMPRAR

2.3.5 Validaciones

- Los productos serán visibles(disponibles) si tienen **stock mayor a 0** y estado **“activo”**.
- Al presionar el botón comprar, la compra pasa a estado **“pendiente”**.

2.4 F4 - Ver listado de compras - Comprador

2.4.2 Acceso

Comprador

2.4.3 Descripción

El comprador podrá ver una lista de los productos y respectiva información que ha seleccionado para efectuar la compra y al final un párrafo con el monto total de las compras realizadas (si es que se aprobaron) así como el saldo disponible. También podrá filtrar las compras por estados: pendientes, canceladas y aprobadas.

2.4.4 Interfaz de usuario

COMPRA DE PRODUCTOS LISTADO DE COMPRAS PRODUCTOS EN OFERTA CERRAR SESIÓN				
Listado de Compras				
FILTRAR COMPRAS				
Todas				
Nombre	Cantidad de Unid.	Monto Total	Estado	
Bolso deportivo	4	1500	pendiente	CANCELAR
Raqueta de Tenis	2	700	pendiente	CANCELAR
Pelota de futbol	1	260	aprobada	
Pelota de volley	3	800	cancelada	CANCELAR
Gorra de natacion	5	750	cancelada	CANCELAR

2.4.5 Validaciones

- En cada compra deberá aparecer un botón que permita cancelar la compra, **solo si ésta**, tiene **estado “pendiente”**.
- Al presionar botón, dicha compra pasara a **estado “cancelada”**.
- Al final mostrar el monto de las compras realizadas, es decir solo las que están en estado “aprobada”.
- Mostrar lista de estados correspondiente según el comprador filtre.

2.5 F5 - Ver productos en oferta - Comprador





2.5.2 Acceso

Comprador

2.5.3 Descripción

El comprador podrá ver los productos que están en oferta y comprar tal cual como en la pantalla principal.

2.5.4 Interfaz de usuario

 COMPRA DE PRODUCTOS LISTADO DE COMPRAS PRODUCTOS EN OFERTA CERRAR SESIÓN						
Productos en Oferta!						
Nombre	Descripción	Precio por unidad. (\$U)		Imagen	Saldo: \$3000	
Botella deportiva	Botella deportiva ideal para salir a correr o ir al gimnasio.	120	EN OFERTA!		<input type="text" value="1"/>	COMPRAR
Pelota de Fútbol Americano	Pelota de Fútbol Americano marca Champion Sports tamaño oficial.	260	EN OFERTA!		<input type="text" value="1"/>	COMPRAR
Raqueta de Tenis	Raqueta de Tenis marca Head ultimo modelo compuesta de aluminio.	400	EN OFERTA!		<input type="text" value="1"/>	COMPRAR

2.5.5 Validaciones

- Al presionar, **mostrar únicamente** productos marcados en oferta.

2.6 F6 - Ingreso al sistema - Administrador

2.6.1 Acceso

Administrador

2.6.2 Descripción

El Administrador debería acceder al sistema realizando el login ingresando nombre de usuario y contraseña ya antes precargados.

6.2.3. Interfaz de usuario

6.2.4. Validaciones

- Todos los campos obligatorios
- Validar nombre de usuario
- Validar contraseña

En caso de no cumplir o algún dato incorrecto, informar error en interfaz HTML.

2.7 F7 - Listado y aprobación de las compras de los usuarios - Administrador


2.7.2 Acceso

Administrador

2.7.3 Descripción

El administrador podrá ver una sección con las compras solicitadas por los usuarios, el cual podrá seleccionar tres listas distintas, de compras pendientes, canceladas y aprobadas. En la lista de compras pendientes el administrador podrá, presionando un botón, aprobar la compra o no.

2.7.4 Interfaz de usuario



LISTADO Y APROBACIÓN DE COMPRAS CREAR PRODUCTOS ADMINISTRAR PRODUCTOS VER INFORME DE GANANCIAS CERRAR SESIÓN

Listado y aprobación de compras

COMPRAS
Pendientes

Nombre	Cantidad de Unid.	Monto Total	Estado	
Bolso deportivo	4	1500	pendiente	APROBAR
Raqueta de Tenis	2	700	pendiente	APROBAR

2.7.5 Validaciones

- Aprobar compra solo si **el usuario tiene saldo suficiente**, el **stock es suficiente** y el **estado es “activo”**.
- Al aprobar la compra el estado pasa a “aprobada”, se **descuenta el stock del producto** y el **saldo del comprador** se actualiza.
- Si al aprobar la compra, **el stock queda en 0**, el estado de la compra **pasa a “pausado”**.
- En caso de que la compra **no pueda ser efectuada** por las condiciones anteriores, esta se pasará a estado **“cancelada”** **NO** alterando stock ni saldo.

2.8 F8 - Crear producto - Administrador


2.8.2 Acceso

Administrador

2.8.3 Descripción

El administrador podrá crear nuevos productos para que los usuarios puedan comprarlos.

2.8.4 Interfaz de usuario



Crear Producto

NOMBRE
Ingrese nombre...

PRECIO
Ingrese precio...

DESCRIPCIÓN
Ingrese una descripcion del producto...

IMAGEN
Seleccionar archivo Ningún archivo seleccionado

STOCK
Max 10

CREAR PRODUCTO

2.8.5 Validaciones

- Todos los campos son obligatorios
- Validar que **stock y precio** sea **mayor que 0**.
- Al crear producto, éste tendrá estado “activo” y se le asignara el correspondiente ID.

2.9 F9 - Administrar productos - Administrador





2.9.2 Acceso

Administrador

2.9.3 Descripción

El administrador deberá tener una vista que permita administrar todos los productos. creados, pudiendo modificar el stock, modificar el estado, asignar el producto de oferta o desmarcarlo si lo está.

2.9.4 Interfaz de usuario

 LISTADO Y APROBACIÓN DE COMPRAS CREAR PRODUCTOS ADMINISTRAR PRODUCTOS VER INFORME DE GANANCIAS CERRAR SESIÓN						
Administrar Productos						
Nombre	Precio por unid. (\$U)		Imagen	Stock actual	Nuevo stock	
Pelota de Basquetbol	300	NO		5	<div>+</div> <div>-</div>	Activo
Botella deportiva	120	EN OFERTA		5	<div>+</div> <div>-</div>	Activo
Pelota de Fútbol Americano	260	EN OFERTA		5	<div>+</div> <div>-</div>	Activo

2.9.5 Validaciones

- Tener en cuenta que, si se cambia el stock a 0, el estado cambiara a “pausado”.
- Mostrar o no en cada caso cada producto según algún cambio de estado o Si esta en oferta o no.

2.10 F10 - Ver informe de ganancias - Administrador

2.10.2 Acceso

Administrador

2.10.3 Descripción

El administrador deberá ver un párrafo con la cantidad total de unidades vendidas de cada producto y la ganancia total por todas las compras realizadas por los usuarios.

2.10.4 Interfaz de usuario

[LISTADO Y APROBACIÓN DE COMPRAS](#)[CREAR PRODUCTOS](#)[ADMINISTRAR PRODUCTOS](#)[VER INFORME DE GANANCIAS](#)[CERRAR SESIÓN](#)

Ver informe de ganancias

Producto: Unidades vendidas: Producto: Unidades vendidas: Producto: Unidades vendidas: Producto: Unidades vendidas: Producto: Unidades vendidas: Producto: Unidades vendidas: Producto: Unidades vendidas: La ganancia total es: 3900

2.10.5 Validaciones

No hay validaciones.

CODIGO COMPLETO HTML

```
<!-- /*Camilo Artola 306498*/ -->
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/estilos.css">
  <title>obligatorioP1</title>
</head>

<body>
  <div id="contenedor">
    <header>
      
      <nav>
        <ul id="navPrincipalAdmin" class="seccion">
          <li>
            <a id="menuAdminListadoCompras">Listado y aprobación de
compras</a>
          </li>
          <li>
            <a id="menuAdminCrearProducto">Crear productos</a>
          </li>
          <li>
            <a id="menuAdminAdministrarProd">Administrar productos</a>
          </li>
          <li>

```

```

        <a id="menuAdminVerInforme">Ver informe de ganancias</a>
    </li>
    <li>
        <a id="menuCerrarSesiónAdmin">Cerrar Sesión</a>
    </li>
</ul>

<ul id="navPrincipalComp" class="seccion">
    <li>
        <a id="menuCompraProductos">Compra de productos</a>
    </li>
    <li>
        <a id="menuListadoCompras">Listado de compras</a>
    </li>
    <li>
        <a id="menuProductosEnOferta">Productos en oferta</a>
    </li>
    <li>
        <a id="menuCerrarSesión">Cerrar Sesión</a>
    </li>
</ul>
</nav>
</header>

<main>
    <section id="seccionLogin" class="seccion">

        <h1>Login</h1>
        <form id="FormLogin">
            <label for="txtNomUsuarioAdmin">Nombre de Usuario</label>
            <input type="text" id="txtNomUsuarioAdmin" placeholder="Ingrese nombre
de usuario..." required />
            <label for="txtContraseñaAdmin">Contraseña</label>
            <input type="text" id="txtContraseñaAdmin" placeholder="Ingrese
contraseña..." required />
            <input type="button" value="Ingresar" id="btnIngresar" />
            <input type="button" value="¿No tienes usuario? Registrate ya."
id="btnRegistrarseYa" />
        </form>
    </section>

    <section id="seccionRegistro" class="seccion">
        <h1>Registro</h1>
        <form id="FormRegistro">
            <label for="txtNombreComp">Nombre</label>
            <input type="text" id="txtNombreComp" placeholder="Ingrese nombre..."
required />
            <label for="txtApellidoComp">Apellido</label>
            <input type="text" id="txtApellidoComp" placeholder="Ingrese
apellido..." required />
            <label for="txtNomUsuarioComp">Nombre de Usuario</label>

```

```

        <input type="text" id="txtNomUsuarioComp" placeholder="Ingrese nombre
de usuario..." required />
        <label for="txtContraseñaComp">Contraseña</label>
        <input type="text" id="txtContraseñaComp" pattern="(?!.*\d)(?!.*[a-
z])(?!.*[A-Z]).{5,}"
            title="La contraseña debe tener al menos 5 caracteres, incluyendo
al menos una mayúscula, una minúscula y un número"
            placeholder="Ingrese contraseña..." required />
        <label for="txtNroTarjeta">Número de tarjeta de crédito</label>
        <input type="text" id="txtNroTarjeta" pattern="\d{4}-\d{4}-\d{4}-
\d{4}" title="WWWW-XXXX-YYYY-ZZZZ"
            placeholder="Ingrese numero de tarjeta..." required />
        <label for="txtCodigoSeg">CVC (Codigo de verificación)</label>
        <input type="text" id="txtCodigoSeg" pattern="\d{3}" title="Numero de
tres digitos"
            placeholder="Ingrese codigo de verificación..." required />
        <input type="button" value="Registrarse" id="btnRegistro" />
    </form>
</section>
<section id="seccionCompraProductos" class="seccion">
    <h1>Compra de Productos</h1>
    <table>
        <thead>
            <tr>
                <th>Nombre</th>
                <th>Descripcion</th>
                <th>Precio por unid. ($U)</th>
                <th></th>
                <th>Imagen</th>
                <th>Saldo: $<span id="saldo1"></span></th>
                <th></th>
            </tr>
        </thead>
        <tbody id="tblProductos">
        </tbody>
    </table>
</section>

<section id="seccionListadoCompras" class="seccion">
    <h1>Listado de Compras</h1>
    <label for="slcFiltrar">Filtrar compras</label>
    <select id="slcFiltrar">
        <option value="todas">Todas</option>
        <option value="pendiente">Pendientes</option>
        <option value="aprobada">Aprobadas</option>
        <option value="cancelada">Canceladas</option>
    </select>
    <table>
        <thead>
            <tr>
                <th>Nombre</th>
                <th>Cantidad de Unid.</th>

```



```

        <th>Monto Total</th>
        <th>Estado</th>
        <th></th>
    </tr>
</thead>
<tbody id="tblListadoCompras">
    <tr>
        <td></td>
        <td></td>
        <td></td>
        <td></td>
    </tr>
</tbody>
</table>
<p id="pResultado"></p>
</section>

<section id="seccionProductosEnOferta" class="seccion">
    <h1>Productos en Oferta!</h1>
    <table>
        <thead>
            <tr>
                <th>Nombre</th>
                <th>Descripcion</th>
                <th>Precio por unid. ($U)</th>
                <th></th>
                <th>Imagen</th>
                <th>Saldo: $<span id="saldo2"></span></th>
                <th></th>
            </tr>
        </thead>
        <tbody id="tblProdEnOferta">
        </tbody>
    </table>
</section>

<section id="seccionCrearProducto" class="seccion">
    <h1>Crear Producto</h1>
    <form id="formProducto">
        <label for="txtNombreProd">Nombre</label>
        <input type="text" id="txtNombreProd" placeholder="Ingrese nombre..."
required />
        <label for="txtPrecio">Precio</label>
        <input type="number" min="1" id="txtPrecio" placeholder="Ingrese
precio..." required />
        <label for="txtDescProd">Descripción</label>
        <textarea name="descripcion" rows="3" cols="40" id="txtDescProd"
placeholder="Ingrese una descripcion del producto..."
required></textarea>
        <label for="ImagenProd">Imagen</label>
        <input type="file" id="ImagenFile" required>
        <label for="txtStock">Stock</label>
    </form>
</section>

```

```

        <input type="number" min="1" max="10" id="txtStock" placeholder="Max
10" required />
        <input type="button" value="Crear Producto" id="btnCrearProd">
    </form>
</section>

<section id="seccionAdministrarProd" class="seccion">
    <h1>Administrar Productos</h1>
    <table>
        <thead>
            <tr>
                <th>Nombre</th>
                <th>Precio por unid. ($U)</th>
                <th></th>
                <th>Imagen</th>
                <th>Stock actual</th>
                <th>Nuevo stock</th>
                <th></th>
            </tr>
        </thead>
        <tbody id="tblAdministrarProd">
        </tbody>
    </table>
</section>

<section id="seccionListadoyAprobacionAdmin" class="seccion">
    <h1>Listado y aprobación de compras</h1>
    <label for="slcFiltrarAdmin">Compras</label>
    <select id="slcFiltrarAdmin">
        <option value="pendiente">Pendientes</option>
        <option value="aprobada">Aprobadas</option>
        <option value="cancelada">Canceladas</option>
    </select>
    <table>
        <thead>
            <tr>
                <th>Nombre</th>
                <th>Cantidad de Unid.</th>
                <th>Monto Total</th>
                <th>Estado</th>
                <th></th>
            </tr>
        </thead>
        <tbody id="tblListadoComprasAdmin">
        </tbody>
    </table>
</section>

<section id="seccionVerInforme" class="seccion">
    <p id="pParrafo"></p>
</section>
</main>

```

```

</div>
<script src="js/obligatorio.js"></script>
</body>

</html>

```

CODIGO COMPLETO JS

```

/*Camilo Artola 306498*/

class Sistema {
  constructor() {
    this.ListaAdmins = [
      new Administrador("camo", "camo2024"),
      new Administrador("kevin", "kevin1234"),
      new Administrador("pedro", "pedrito124"),
      new Administrador("manu", "manu1234"),
      new Administrador("tina", "tina1234"),
    ]
    this.ListaCompradores = [
      new Comprador("Camilo", "Artola", "camoartola", "Camo2025",
"4213000403490902", "123"),
      new Comprador("Kevin", "Gomez", "kevingomez", "Kevin1234", "4213000403490902",
"123"),
      new Comprador("Pedro", "Gonzalez", "pedrogon21", "Pedro1234",
"4213000403490902", "123"),
      new Comprador("Manuel", "Gutierrez", "manu2002", "Manu1234",
"4213000403490902", "123"),
      new Comprador("Martina", "Tapia", "tina2024", "Marti124", "4213000403490902",
"123"),
    ];
    this.ListaProductos = [
      new Producto("Pelota de Basquetbol", "Pelota de Basquetbol marca Nivia,
incluye inflador de regalo.", 300, "PelotaBasquet.jpg", "activo", 5, false),
      new Producto("Botella deportiva", "Botella deportiva ideal para salir a correr
o ir al gimnasio.", 120, "Botella.jpg", "activo", 5, true),
      new Producto("Pelota de Fútbol Americano", "Pelota de Fútbol Americano marca
Champion Sports tamaño oficial.", 260, "PelotaFutbolAmericano.jpg", "activo", 5, true),
      new Producto("Bolso deportivo", "Bolso deportivo grande y duradero para llevar
lo que quieras.", 350, "BolsoDeportivo.jpeg", "activo", 5, false),
      new Producto("Raqueta de Tenis", "Raqueta de Tenis marca Head ultimo modelo
compuesta de aluminio.", 400, "raquetaTenis.jpg", "activo", 5, true),
      new Producto("Medias Antideslizantes", "Par de Medias Antideslizantes para
Futbol de muy buena calidad.", 220, "MediasAntideslizantes.jpg", "activo", 5, false),

```

```

        new Producto("Licra deportiva", "Licra deportiva de manga larga ideal para
epocas de frio y buen calentamiento.", 300, "Licra.jpg", "activo", 5, true),
        new Producto("Kit de Mancuernas", "Kit de mancuernas de 20kg desarmable de
acero inoxidable.", 400, "kitMancuernas.jpg", "activo", 5, false),
        new Producto("Kit de Raquetas de Ping Pong", "Kit de par de Raquetas de Ping
Pong con tres pelotas incluidas.", 420, "raquetaPingpong.jpg", "activo", 5, true),
        new Producto("Cuerda para saltar", "Cuerda para saltar marca Domyos con
revestimiento de espuma para mayor comodidad.", 300, "CuerdaParaSaltar.jpeg", "activo", 5,
false)
    ];
    this.ListaCompras = [
        new Compra("Bolso deportivo", 4, 1500, "pendiente"),
        new Compra("Raqueta de Tenis", 2, 700, "pendiente"),
        new Compra("Pelota de futbol", 1, 260, "aprobada"),
        new Compra("Pelota de volley", 3, 800, "cancelada"),
        new Compra("Gorra de natacion", 5, 750, "cancelada"),
    ];
    this.usuarioLogueado = null;
    this.ListaProdEnOferta = [];
}

aprobarCompra(saldo, total, stock, CantUnid, estado) {
    let laCompra = this.obtenerElemento(this.ListaCompras, "total", total)
    let elProducto = this.obtenerElemento(this.ListaProductos, "stock", stock)
    if (((saldo - total) >= 0)
        && ((stock - CantUnid) >= 0)
        && (estado === "activo")) {
        laCompra.estado = "aprobada"
        elProducto.stock = stock - CantUnid;
        this.usuarioLogueado.saldo = saldo - total;
        if (stock == 0) {
            elProducto.estado = "pausado"
        }
    }
    else {
        laCompra.estado = "cancelada"
    }
}

cambiarEstado(idCompra) {
    let laCompra = this.obtenerElemento(this.ListaCompras, "id", idCompra)
    laCompra.estado = "cancelada"
}

montoTotalCompras() {
    let total = 0;
    for (let i = 0; i < this.ListaCompras.length; i++) {
        let unaCompra = this.ListaCompras[i];
    }
}

```

```

        if (unaCompra.estado === "aprobada") {
            total += unaCompra.total
        }
    }
    gananciaTotal += total;
    return total;
}

agregarCompradorEnSist(unComprador) {
    this.ListaCompradores.push(unComprador);
}

agregarCompraEnSist(unaCompra) {
    this.ListaCompras.push(unaCompra);
}

actualizarListaSistEnOfertaEnSist() {
    this.ListaProdEnOferta = [];
    for (let i = 0; i < this.ListaProductos.length; i++) {
        let unProd = this.ListaProductos[i];
        if (unProd.enOferta) {
            this.ListaProdEnOferta.push(unProd);
        }
    }
}

actualizarListaSistuctoEnSist(unProducto) {
    this.ListaProductos.push(unProducto);
}

obtenerProductosSist() {
    return this.ListaProductos
}

obtenerElemento(arrayElem, prop, valor) {
    let obj = null
    for (let i = 0; i < arrayElem.length; i++) {
        let unElemento = arrayElem[i];
        if (unElemento[prop] === valor) {
            obj = unElemento
            break;
        }
    }
    return obj
}

cambiarEnOferta(idProd) {
    let elProducto = this.obtenerElemento(this.ListaProductos, "id", idProd)
    if (elProducto.enOferta) {
        elProducto.enOferta = false
    }
}

```

```

        else {
            elProducto.enOferta = true
        }
    }

    aumentarStock(idProd) {
        let elProducto = this.obtenerElemento(this.ListaProductos, "id", idProd)
        if (elProducto.stock == 0) {
            elProducto.stock++
            elProducto.estado = "activo"
        }
        else {
            elProducto.stock++
        }
    }

    disminuirStock(idProd) {
        let elProducto = this.obtenerElemento(this.ListaProductos, "id", idProd)
        if (elProducto.stock > 0) {
            if (elProducto.stock == 1) {
                elProducto.estado = "pausado"
                elProducto.stock--
            }
            else {
                elProducto.stock--
            }
        }
    }

    cambiarActivoPausado(idProd) {
        let elProducto = this.obtenerElemento(this.ListaProductos, "id", idProd)
        if (elProducto.estado === "activo") {
            elProducto.estado = "pausado"
        }
        else if (elProducto.stock == 0) {
            elProducto.stock++
            elProducto.estado = "activo"
        } else {
            elProducto.estado = "activo"
        }
    }
}

let gananciaTotal = 0;

let contadorAdmin = 1

class Administrador {
    constructor(usuario, pass) {
        this.id = contadorAdmin
    }
}

```

```

        this.usuario = usuario
        this.pass = pass
        contadorAdmin++
    }
}

let contadorComprador = 1

class Comprador {
    constructor(nombre, apellido, usuario, pass, NroTarjeta, codigoCVC) {
        this.id = contadorComprador
        this.nombre = nombre
        this.apellido = apellido
        this.usuario = usuario
        this.pass = pass
        this.nroTarjeta = NroTarjeta
        this.codigoCVC = codigoCVC
        this.saldo = 3000
        contadorComprador++
    }
}

let contadorProducto = 1

class Producto {
    constructor(nomProd, descProd, precio, UrlImagen, estado, stock, enOferta) {
        this.nombre = nomProd
        this.descripcion = descProd
        this.precio = precio
        this.imagen = UrlImagen
        this.stock = stock
        this.estado = estado
        this.enOferta = enOferta
        this.id = "PROD_ID_" + contadorProducto
        contadorProducto++
    }
}

let contCompra = 1

class Compra {
    constructor(nomProd, cantUnid, montoTotal, estado) {
        this.nombre = nomProd
        this.unidades = cantUnid
        this.total = montoTotal
        this.estado = estado
        this.id = contCompra
        contCompra++
    }
}

```

```

let sistema = new Sistema();

inicio()

function inicio() {

    OcultarSecciones();
    actualizarListaSistEnOferta();
    document.querySelector("#seccionLogin").style.display = "block";
    document.querySelector("#btnIngresar").addEventListener("click", hacerLogin);
    document.querySelector("#btnRegistrarseYa").addEventListener("click",
mostrarRegistro);
    document.querySelector("#btnRegistro").addEventListener("click",
hacerRegistroComprador);
    document.querySelector("#menuCompraProductos").addEventListener("click",
menuCompraProductos);
    document.querySelector("#menuListadoCompras").addEventListener("click",
menuListadoCompras);
    document.querySelector("#menuProductosEnOferta").addEventListener("click",
menuProductosEnOferta);
    document.querySelector("#menuAdminCrearProducto").addEventListener("click",
menuCrearProducto);
    document.querySelector("#menuAdminAdministrarProd").addEventListener("click",
menuAdministrarProducto);
    document.querySelector("#menuAdminListadoCompras").addEventListener("click",
menuAdminAprobarCompra);
    document.querySelector("#btnCrearProd").addEventListener("click", crearProducto);
    document.querySelector("#menuCerrarSesión").addEventListener("click", cerrarSesion);
    document.querySelector("#menuCerrarSesiónAdmin").addEventListener("click",
cerrarSesion);
    document.querySelector("#menuAdminVerInforme").addEventListener("click", VerInforme);
}

function VerInforme() {

}

function actualizarListaSistEnOferta() {
    sistema.actualizarListaSistEnOfertaEnSist();
}

function cargarTablaAdministrarProd() {
    document.querySelector("#tblAdministrarProd").innerHTML = "";
    let tblAdministrarProd = "";
    for (let i = 0; i < sistema.ListaProductos.length; i++) {
        let unProd = sistema.ListaProductos[i];
        tblAdministrarProd = `
            <tr>
            <td>${unProd.nombre}</td>
            <td>${unProd.precio}</td>`

        if (unProd.enOferta) {

```



```

        tblAdministrarProd += `<td>
            <select class="slcEnOferta" id="slc${unProd.id}">
                <option value selected="EnOferta">EN OFERTA</option>
                <option value="NoEnOferta">NO</option>
            </select>
        </td>`
    } else {
        tblAdministrarProd += `<td>
            <select class="slcEnOferta" id="slc${unProd.id}">
                <option value="EnOferta">EN OFERTA</option>
                <option value selected="NoEnOferta">NO</option>
            </select>
        </td>`
    }
    tblAdministrarProd += `
        <td></td>
        <td>${unProd.stock}</td>
        <td><input type="button" value="+" class="btnAumentar"
id="aumentar${unProd.id}"/><input type="button" value="-" class="btnDisminuir"
id="disminuir${unProd.id}"/>
        </td>`

    if (unProd.estado === "activo") {
        tblAdministrarProd += `
            <td>
                <select class="slcActivarPausar" id="slcActivarPausar${unProd.id}">
                    <option value selected="Activo">Activo</option>
                    <option value="Pausado">Pausado</option>
                </select>
            </td>
        </tr>`
    }
    else {
        tblAdministrarProd += `
            <td>
                <select class="slcActivarPausar" id="slcActivarPausar${unProd.id}">
                    <option value="Activo">Activo</option>
                    <option value selected="Pausado">Pausado</option>
                </select>
            </td>
        </tr>`
    }
}

document.querySelector("#tblAdministrarProd").innerHTML += tblAdministrarProd
}

let listaSlcOferta = document.querySelectorAll(".slcEnOferta")

for (let i = 0; i < listaSlcOferta.length; i++) {
    const unSlc = listaSlcOferta[i];
    unSlc.addEventListener("change", cambiarEnOferta)
}

```

```

let listaBtnAumentar = document.querySelectorAll(".btnAumentar")

for (let i = 0; i < listaBtnAumentar.length; i++) {
    const unBtn = listaBtnAumentar[i];
    unBtn.addEventListener("click", aumentarStock)
}

let listabBtnDisminuir = document.querySelectorAll(".btnDisminuir")

for (let i = 0; i < listabBtnDisminuir.length; i++) {
    const unBtn = listabBtnDisminuir[i];
    unBtn.addEventListener("click", disminuirStock)
}

let listaSlcActivarPausar = document.querySelectorAll(".slcActivarPausar")

for (let i = 0; i < listaSlcActivarPausar.length; i++) {
    const unSlc = listaSlcActivarPausar[i];
    unSlc.addEventListener("change", cambiarActivoPausado)
}
}

function aumentarStock() {
    let idBtn = this.id;
    let idProd = idBtn.substring(8);
    sistema.aumentarStock(idProd);
    actualizarTablasProductos();
}

function disminuirStock() {
    let idBtn = this.id;
    let idProd = idBtn.substring(9);
    sistema.disminuirStock(idProd);
    actualizarTablasProductos();
}

function cambiarEnOferta() {
    let idSlc = this.id;
    let idProd = idSlc.substring(3);
    sistema.cambiarEnOferta(idProd);
    actualizarTablasProductos();
    actualizarListaSistEnOferta();
}

function cambiarActivoPausado() {
    let idSlc = this.id;
    let idProd = idSlc.substring(16);
    sistema.cambiarActivoPausado(idProd);
    actualizarTablasProductos();
}

```

```

function actualizarTablasProductos() {
    cargarTablaAdministrarProd();
    armarTablaProductos();
    armarTablaProductosEnOferta();
}

function crearProducto() {
    let formProd = document.querySelector("#formProducto")
    if (formProd.reportValidity()) {
        let nombreProd = document.querySelector("#txtNombreProd").value;
        let precio = Number(document.querySelector("#txtPrecio").value);
        let descProd = document.querySelector("#txtDescProd").value;
        let ImagenProd = document.querySelector("#ImagenFile").value;
        let nombreImagen = ImagenProd.substring(ImagenProd.lastIndexOf("\\") + 1);
        let stock = Number(document.querySelector("#txtStock").value);
        let estado = "activo";
        let enOferta = false;
        let ProdAux = new Producto(nombreProd, descProd, precio, nombreImagen, estado,
stock, enOferta);
        sistema.actualizarListaSistuctoEnSist(ProdAux)
        formProd.reset();
    }
}

function cerrarSesion() {
    inicio();
}

function OcultarSecciones() {
    let secciones = document.querySelectorAll(".seccion");
    for (let i = 0; i < secciones.length; i++) {
        const unaSeccion = secciones[i];
        unaSeccion.style.display = "none"
    }
}

function menuCompraProductos() {
    OcultarSecciones();
    document.querySelector("#navPrincipalComp").style.display = "block";
    document.querySelector("#seccionCompraProductos").style.display = "block";
    armarTablaProductos();
}

function menuListadoCompras() {
    OcultarSecciones();
    document.querySelector("#navPrincipalComp").style.display = "block";
    document.querySelector("#seccionListadoCompras").style.display = "block";
    ListarCompras();
}

function menuProductosEnOferta() {

```

```

OcultarSecciones();
document.querySelector("#navPrincipalComp").style.display = "block";
document.querySelector("#seccionProductosEnOferta").style.display = "block";
mostrarSaldo();
}

function menuCrearProducto() {
    OcultarSecciones();
    document.querySelector("#navPrincipalAdmin").style.display = "block";
    document.querySelector("#seccionCrearProducto").style.display = "block";
}

function menuAdministrarProducto() {
    OcultarSecciones();
    document.querySelector("#navPrincipalAdmin").style.display = "block";
    document.querySelector("#seccionAdministrarProd").style.display = "block";
    cargarTablaAdministrarProd();
    ListarCompras();
    ListarComprasAdmin();
}

function menuAdminAprobarCompra() {
    OcultarSecciones();
    document.querySelector("#navPrincipalAdmin").style.display = "block";
    document.querySelector("#seccionListadoyAprobacionAdmin").style.display = "block";
}

function hacerLogin() {
    let formLogin = document.querySelector("#FormLogin")
    if (formLogin.reportValidity()) {
        let usuario = document.querySelector("#txtNomUsuarioAdmin").value.toLowerCase();
        let pass = document.querySelector("#txtContraseñaAdmin").value;
        let adminBuscado = sistema.obtenerElemento(sistema.ListaAdmins, "usuario",
usuario)

        if (adminBuscado != null) {

            if (adminBuscado.pass === pass) {
                OcultarSecciones();
                document.querySelector("#navPrincipalAdmin").style.display = "block";
                document.querySelector("#seccionListadoyAprobacionAdmin").style.display =
"block";

                ListarComprasAdmin();
                ListarCompras();
                actualizarTablasProductos();
                formLogin.reset();
            }
            else {
                alert("La contraseña es incorrecta")
            }
        }
        else {

```

```

        let CompBuscado = sistema.obtenerElemento(sistema.ListaCompradores, "usuario",
usuario)

        if (CompBuscado != null) {

            if (CompBuscado.pass === pass) {
                OcultarSecciones();
                document.querySelector("#navPrincipalComp").style.display = "block";
                document.querySelector("#seccionCompraProductos").style.display =
"block";

                sistema.usuarioLogueado = CompBuscado;
                actualizarTablasProductos();
                ListarCompras();
                ListarComprasAdmin();
                mostrarSaldo();
                formLogin.reset();
            }
            else {
                alert("La contraseña es incorrecta")
            }
        }
        else {
            alert("El usuario que acaba de ingresar no existe")
        }
    }
}

function mostrarSaldo() {
    let saldo1 = document.querySelector("#saldo1");
    let saldo2 = document.querySelector("#saldo2");
    saldo1.innerHTML = sistema.usuarioLogueado.saldo;
    saldo2.innerHTML = sistema.usuarioLogueado.saldo;
}

function armarTablaProductos() {
    document.querySelector("#tblProductos").innerHTML = "";
    let tblProductos = "";
    for (let i = 0; i < sistema.ListaProductos.length; i++) {
        let unProd = sistema.ListaProductos[i];

        if (unProd.estado === "activo" && unProd.stock > 0) {
            tblProductos = `
                <tr>
                <td>${unProd.nombre}</td>
                <td>${unProd.descripcion}</td>
                <td>${unProd.precio}</td>
            `
        }
    }
}

```

```

        if (unProd.enOferta === true) {
            tblProductos += `<td> EN OFERTA!</td>`;
        } else {
            tblProductos += `<td></td>`
        }
        tblProductos +=

            `<td></td>`
            `<td><input type="number" min="1" max="8" value="1" step="1" `
class="CantUnidades" id="Unid${unProd.id}"/></td>`
            `<td><input type="button" value="Comprar" class="btnComprar" `
id="btnComprar${unProd.id}"/></td>`
            `</tr> `

        document.querySelector("#tblProductos").innerHTML += tblProductos
    }
}

let listaBtns = document.querySelectorAll(".btnComprar")

for (let i = 0; i < listaBtns.length; i++) {
    const unBtn = listaBtns[i];
    unBtn.addEventListener("click", solicitarCompra)
}

}

function solicitarCompra() {

    let idBoton = this.id
    let idProducto = (idBoton.substring(10));
    let idCantUnid = "Unid" + idProducto;
    let CantUnid = Number(document.querySelector("#" + idCantUnid).value);
    let Prod = sistema.obtenerElemento(sistema.ListaProductos, "id", idProducto);
    let nombreProd = Prod.nombre;
    let estado = "pendiente";
    let montoTotal = CantUnid * Prod.precio;
    let CompraAux = new Compra(nombreProd, CantUnid, montoTotal, estado);
    sistema.agregarCompraEnSist(CompraAux);
    alert("Solicitud de compra realizada");
    ListarCompras();
    ListarComprasAdmin();
}

document.querySelector("#slcFiltrar").addEventListener("change", ListarCompras)
function ListarCompras() {
    document.querySelector("#tblListadoCompras").innerHTML = "";
    document.querySelector("#pResultado").innerHTML = "";
    let tblListadoCompras = "";
    let pResultado = "";
    let estado = document.querySelector("#slcFiltrar").value;
    for (let i = 0; i < sistema.ListaCompras.length; i++) {
        let unaCompra = sistema.ListaCompras[i];
    }
}

```

```

        if (unaCompra.estado === estado || estado === "todas") {
            tblListadoCompras = `
<tr>
<td>${unaCompra.nombre}</td>
<td>${unaCompra.unidades}</td>
<td>${unaCompra.total}</td>
<td>${unaCompra.estado}</td>
`

            if (unaCompra.estado === "pendiente") {
                tblListadoCompras += `
<td><input type="button" value="Cancelar" class="btnCancelar"
id="btnCancelar${unaCompra.id}" /></td>
</tr>
`

            } else if (unaCompra.estado === "cancelada") {
                tblListadoCompras += `
<td><input type="button" value="Cancelar" class="btnCancelar"
id="btnCancelar${unaCompra.id}" disabled /></td>
</tr>
`

            } else {
                tblListadoCompras += `
<td></td></tr>`
            }
        }

        document.querySelector("#tblListadoCompras").innerHTML += tblListadoCompras;
        tblListadoCompras = ""
    }

    let totalCompras = sistema.montoTotalCompras();
    let saldoDisponible = sistema.usuarioLogueado.saldo;
    pResultado = "Monto total: " + totalCompras + "<br><br> Saldo disponible: " +
    saldoDisponible;
    document.querySelector("#pResultado").innerHTML += pResultado;

    let listaBtnsCancelar = document.querySelectorAll(".btnCancelar")

    for (let i = 0; i < listaBtnsCancelar.length; i++) {
        const unBtn = listaBtnsCancelar[i];
        unBtn.addEventListener("click", cancelarCompra)
    }
}

function cancelarCompra() {
    let idBoton = this.id
    let idCompra = Number(idBoton.substring(11))
    console.log(idCompra)
    sistema.cambiarEstado(idCompra)
    ListarCompras();
    ListarComprasAdmin();
}

```

```

}

document.querySelector("#slcFiltrarAdmin").addEventListener("change", ListarComprasAdmin)
function ListarComprasAdmin() {
    document.querySelector("#tblListadoComprasAdmin").innerHTML = "";
    let tblListadoComprasAdmin = "";
    let estado = document.querySelector("#slcFiltrarAdmin").value;
    for (let i = 0; i < sistema.ListaCompras.length; i++) {
        let unaCompra = sistema.ListaCompras[i];
        if (unaCompra.estado === estado) {
            tblListadoComprasAdmin = `
<tr>
<td>${unaCompra.nombre}</td>
<td>${unaCompra.unidades}</td>
<td>${unaCompra.total}</td>
<td>${unaCompra.estado}</td>
`

            if (unaCompra.estado === "pendiente") {
                tblListadoComprasAdmin += `
<td><input type="button" value="Aprobar" class="btnAprobar"
id="btnAprobar${unaCompra.id}" /></td>
</tr>
`
            } else if (unaCompra.estado === "cancelada") {
                tblListadoComprasAdmin += `
<td><input type="button" value="Aprobar" class="btnAprobar"
id="btnAprobar${unaCompra.id}" disabled /></td>
</tr>
`
            } else {
                tblListadoComprasAdmin += `</tr>`
            }
        }
    }
    document.querySelector("#tblListadoComprasAdmin").innerHTML +=
tblListadoComprasAdmin;
    tblListadoComprasAdmin = "";
}

let listaBtnsAprobar = document.querySelectorAll(".btnAprobar")

for (let i = 0; i < listaBtnsAprobar.length; i++) {
    const unBtn = listaBtnsAprobar[i];
    unBtn.addEventListener("click", aprobarCompra)
}

function aprobarCompra() {
    let idBoton = this.id;
    let idCompra = Number(idBoton.substring(10));
    let laCompra = sistema.obtenerElemento(sistema.ListaCompras, "id", idCompra);

```



```

    let total = laCompra.total;
    let CantUnid = laCompra.unidades;
    let nombreCompra = laCompra.nombre;
    let elProducto = sistema.obtenerElemento(sistema.ListaProductos, "nombre",
nombreCompra);
    let stockProducto = elProducto.stock;
    let estadoProd = elProducto.estado;
    let saldo = sistema.usuarioLogueado.saldo;
    sistema.aprobarCompra(saldo, total, stockProducto, CantUnid, estadoProd);
    ListarCompras();
    ListarComprasAdmin();
}

function armarTablaProductosEnOferta() {
    document.querySelector("#tblProdEnOferta").innerHTML = "";
    let tblProdEnOferta = "";
    for (let i = 0; i < sistema.ListaProdEnOferta.length; i++) {
        let unProdEnOferta = sistema.ListaProdEnOferta[i];

        if (unProdEnOferta.estado === "activo" && unProdEnOferta.stock > 0) {
            tblProdEnOferta = `
                <tr>
                <td>${unProdEnOferta.nombre}</td>
                <td>${unProdEnOferta.descripcion}</td>
                <td>${unProdEnOferta.precio}</td>
                <td>EN OFERTA!</td>
                <td></td>
                <td><input type="number" min="1" max="8" value="1" step="1"
class="CantUnidades" id="Unid${unProdEnOferta.id}"/></td>
                <td><input type="button" value="Comprar" class="btnComprar"
id="btnComprar${unProdEnOferta.id}"/></td>
                </tr>`

            document.querySelector("#tblProdEnOferta").innerHTML += tblProdEnOferta
        }

        let listaBtns2 = document.querySelectorAll(".btnComprar")

        for (let i = 0; i < listaBtns2.length; i++) {
            const unBtn = listaBtns2[i];
            unBtn.addEventListener("click", solicitarCompra)
        }
    }
}

function mostrarRegistro() {
    OcultarSecciones();
    document.querySelector("#seccionRegistro").style.display = "block";
}

```

```

function hacerRegistroComprador() {
    let formRegistro = document.querySelector("#FormRegistro")
    if (formRegistro.reportValidity()) {
        let nombre = document.querySelector("#txtNombreComp").value;
        let apellido = document.querySelector("#txtApellidoComp").value;
        let usuario = document.querySelector("#txtNomUsuarioComp").value.toLowerCase();
        let pass = document.querySelector("#txtContraseñaComp").value;
        let NroTarjeta =
eliminarGuionesTarjeta(document.querySelector("#txtNroTarjeta").value);
        let codigoCVC = document.querySelector("#txtCodigoSeg").value;

        let existeUsuarioComp = sistema.obtenerElemento(sistema.ListaCompradores,
"usuario", usuario)
        let existeUsuarioAdmin = sistema.obtenerElemento(sistema.ListaAdmins, "usuario",
usuario)

        if (existeUsuarioAdmin == null && existeUsuarioComp == null) {

            let TarjetaValida = validarTarjeta(NroTarjeta);
            if (TarjetaValida) {
                OcultarSecciones();
                document.querySelector("#seccionLogin").style.display = "block";
                let CompradorAux = new Comprador(nombre, apellido, usuario, pass,
NroTarjeta, codigoCVC);
                sistema.agregarCompradorEnSist(CompradorAux);
                formRegistro.reset();
            } else {
                alert("El numero de tarjeta introducido no es válido")
            }
        }
        else {
            alert("El nombre de usuario introducido ya está en uso")
        }
    }
}

function eliminarGuionesTarjeta(TarjetaString) {
    let NrosTarjeta = TarjetaString.replace(/-/g, '');
    return NrosTarjeta;
}

function validarTarjeta(numeroTarjeta) {
    let tarjetaValida = false
    let digitoAVerificar = numeroTarjeta.charAt(numeroTarjeta.length - 1)
    let acumulador = 0

    for (let i = 0; i < numeroTarjeta.length - 1; i++) {
        if (i % 2 === 0) {
            let duplicado = Number(numeroTarjeta.charAt(i)) * 2
            if (duplicado >= 10) {
                let duplicadoString = String(duplicado)

```

```

        let resultado = Number(duplicadoString.charAt(0)) +
Number(duplicadoString.charAt(1))
        acumulador += resultado
    } else {

        acumulador += duplicado
    }

    } else {
        acumulador += Number(numeroTarjeta.charAt(i))
    }
}
let multiplicadoPor9 = acumulador * 9
let multiplicadoPor9String = String(multiplicadoPor9)
let digitoVerificador = multiplicadoPor9String.charAt(multiplicadoPor9String.length -
1)
if (digitoAVerificar === digitoVerificador) {
    tarjetaValida = true
}
return tarjetaValida
}

```