



TIENDA

VERDES SOUVENIRS



FullStack II - 001D

Profesor:

Eduardo Baeza



JONATHAN LARRAGUIBEL

EUGENIO PARADA

INTRODUCCION

Verdes Souvenirs es una tienda dedicada a la venta de artículos y recuerdos personalizados, con un estilo único y cercano a nuestros clientes. Nuestro objetivo es ofrecer productos originales que conecten con las personas, ya sea como regalo, colección o recuerdo especial.

Actualmente, la mayoría de nuestras ventas y difusión se realizan de manera presencial y a través de redes sociales, lo que limita nuestro alcance. Por eso, necesitamos una página web, que nos permita:

- Ampliar nuestra presencia digital.
- Llegar a más clientes en distintas comunas y regiones.
- Ofrecer un catálogo organizado y fácil de navegar.
- Facilitar la compra en línea y mejorar la experiencia del cliente.

En resumen, la web será una herramienta clave para modernizar la tienda, aumentar las ventas y consolidar nuestra identidad de marca.



ESTRUCTURA Y ETIQUETADO HTML



En nuestro proyecto Verdes-Souvenir, hemos implementado HTML5 con una estructura semántica sólida.

DOCTYPE HTML5 (<!DOCTYPE html>) en todas nuestras páginas y configuramos el idioma español (lang="es") para una correcta interpretación del contenido.

- <header>
- <nav>
- <main>
- <section>

Utilizamos etiquetas apropiadas como <h1>, <h2> para jerarquizar títulos, <form> para formularios, <label> asociadas correctamente a inputs, y atributos semánticos como ALT en imágenes para mejorar la accesibilidad.

- <h1>
- <h2>
- <label>
- <form>



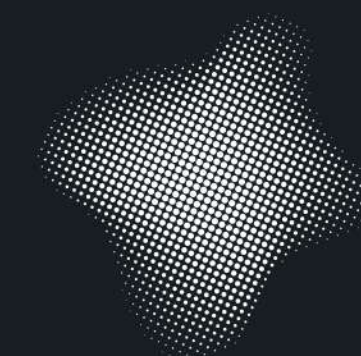


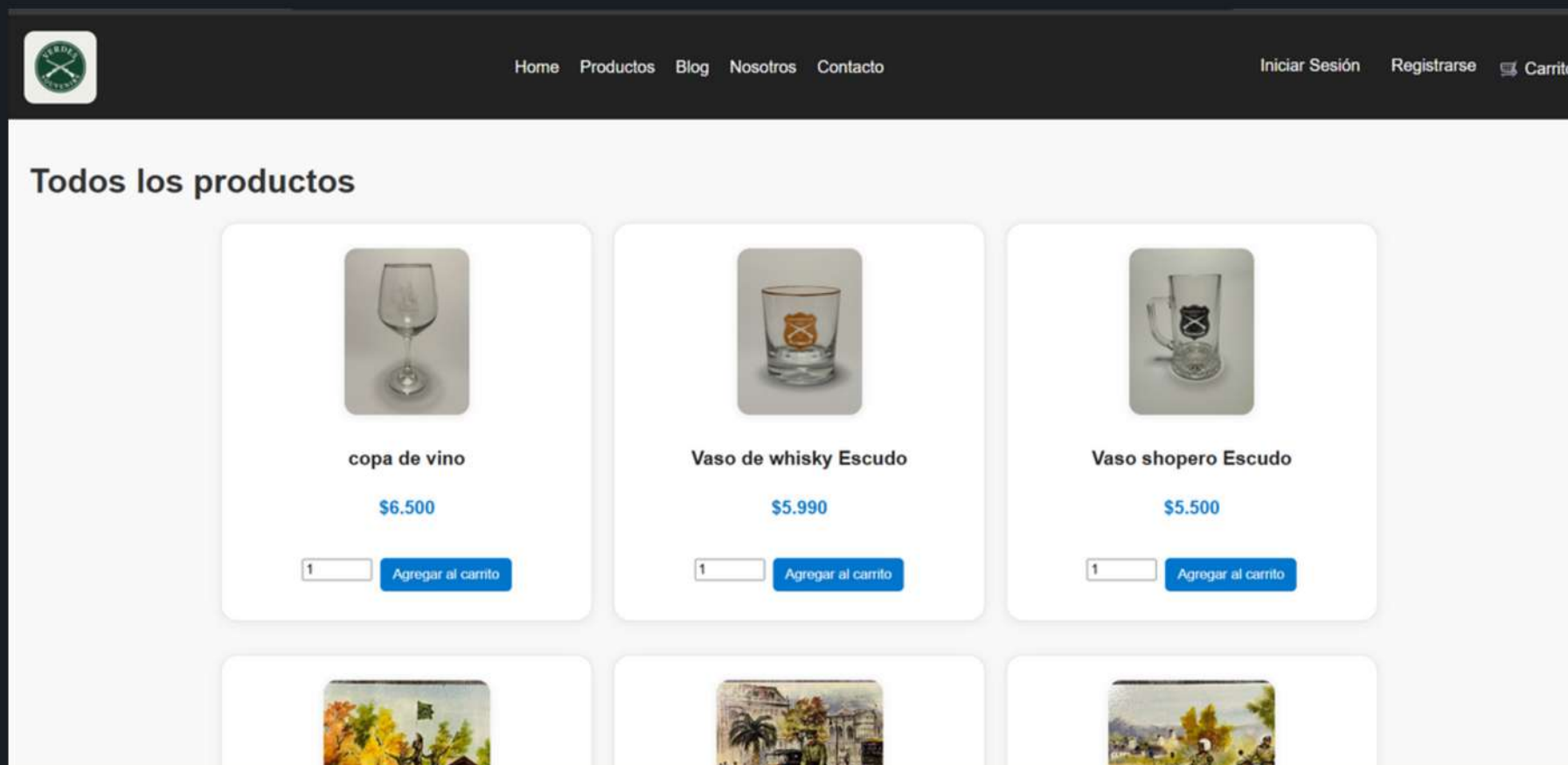
Bienvenido a Verdes-Souvenir

Encuentra copas, vasos y placas conmemorativas de Carabineros. Productos únicos para coleccionistas y simpatizantes de la institución, ideales para regalar o complementar tu colección.



Productos Destacados





```
// Renderizar productos desde localStorage
function renderizarProductosAdmin() {
  const contenedor = document.querySelector('.lista-productos');
  if (!contenedor) return;
  const productos = obtenerProductos();
  if (!productos.length) {
    contenedor.innerHTML = '<div style="text-align:center; color:#888;">No hay productos disponibles.</div>';
    return;
  }
  contenedor.innerHTML = productos.map((prod, idx) => `
    <div class="producto" data-id="${idx}">
      ${prod.imagen ? `<img src='../img/Productos/${prod.imagen}' alt='${prod.nombre}'>` : ''}
      <h2>${prod.nombre}</h2>
      <p>${parseFloat(prod.precio).toLocaleString('es-CL')}</p>
      <div class="agregar-carrito-cantidad" style="margin:10px 0; display:flex; align-items:center; justify-content:center; gap:8px;">
        <input type="number" min="1" value="1" class="input-cantidad" style="width:60px;">
        <button class="btn-agregar-carrito">Agregar al carrito</button>
      </div>
    </div>
  `).join('');
```

REGISTROS, INICIO SESION



Registro de Usuario

RUN*

Ej: 19011022K

Debe ingresar el RUN sin puntos ni guion.

Nombre*

Apellidos*

Correo*

usuario@duoc.cl

Solo se permiten correos @duoc.cl, @profesor.duoc.cl y @gmail.com

Contraseña*

Fecha de nacimiento

dd/mm/aaaa



Región*

Seleccione región



Comuna*



Dirección*

Registrarse

[Home](#) [Productos](#) [Blog](#) [Nosotros](#) [Contacto](#)

Iniciar Sesión

Correo*

usuario@duoc.cl

Solo se permiten correos @duoc.cl, @profesor.duoc.cl y @gmail.com

Contraseña*

Iniciar Sesión

ADMINISTADOR PRODUCTO



Administrador

Dashboard

Usuarios

Productos

Mensajes

Cerrar Sesión

Administración de Productos



Mostrar Productos

Ver y gestionar el listado de productos existentes.



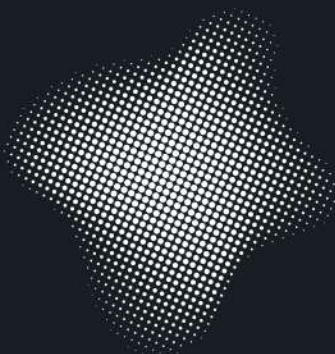
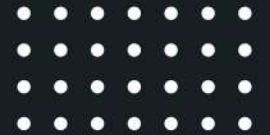
Nuevo Producto

Agregar un nuevo producto al catálogo.



Editar Producto

Modificar información de productos existentes.



ADMINISTADOR USUARIO



Administrador

Gestión de Usuarios

+ Nuevo Usuario

RUN	Nombre	Apellidos	Correo	Dirección	Región	Comuna	Tipo Usuario	Contraseña	Acciones
12345678K	Admin	Prueba	admin@duoc.cl	Dirección admin	Región Metropolitana	Santiago	administrador	admin1234	<button>Editar</button> <button>Eliminar</button>

Dashboard

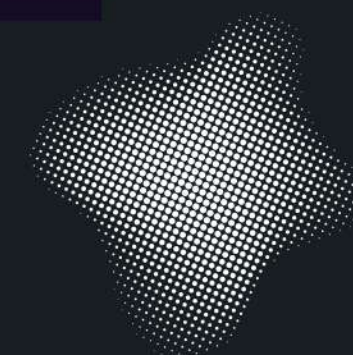
Usuarios

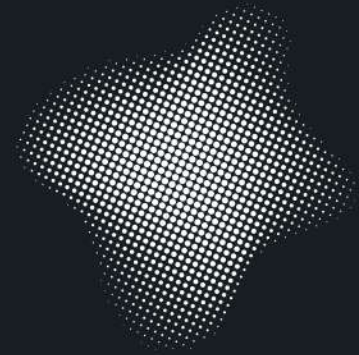
Productos

Mensajes

Cerrar Sesión

```
function mostrarUsuarios() {  
  const tbody = document.getElementById("lista-usuarios");  
  tbody.innerHTML = "";  
  
  let usuarios = JSON.parse(localStorage.getItem("usuarios")) || [];  
  
  usuarios.forEach((u, index) => {  
    const fila = document.createElement("tr");  
  
    fila.innerHTML = `  
      <td>${u.run}</td>  
      <td>${u.nombre}</td>  
      <td>${u.apellidos}</td>  
      <td>${u.correo}</td>  
      <td>${u.direccion}</td>  
      <td>${u.region}</td>  
      <td>${u.comuna}</td>  
      <td>${u.tipoUsuario}</td>  
      <td>${u.contrasena}</td>  
      <td>  
        <button onclick="editarUsuario(${index})">Editar</button>  
        <button onclick="eliminarUsuario(${index})">Eliminar</button>  
      </td>  
    `;  
  
    tbody.appendChild(fila);  
  });  
}
```





USO DE CSS

Hemos implementado un sistema de estilos CSS completamente personalizado y externo, utilizando el archivo “Estilos.css” principal y “EstilosAdmin.css” para la zona administrativa

- Mantenimiento eficiente: Un solo archivo CSS controla el diseño de toda la aplicación
- Carga optimizada: El navegador cachea el archivo CSS, mejorando el rendimiento
- Consistencia visual: Todos los componentes mantienen un diseño uniforme

● `<link rel="stylesheet" href="../css/Estilos.css">`

CARACTERÍSTICAS DEL DISEÑO IMPLEMENTADO:

- Layout flexible: Utilizamos Flexbox para crear una estructura responsiva con display: flex y flex-direction: column
- Sistema de colores coherente: Paleta definida con colores corporativos (#222 para headers, #0077cc para botones)
- Componentes reutilizables: Clases como .navbar, .tabla-bienvenida, .productos-destacados que se aplican consistentemente
- Efectos visuales: Sombras, bordes redondeados y transiciones que mejoran la experiencia de usuario

```
MDN Reference
header {
  background: #222;
  color: white;
  padding: 15px;
}

button {
  margin-top: 10px;
  padding: 8px 12px;
  border: none;
  background: #0077cc;
  color: white;
  cursor: pointer;
  border-radius: 5px;
}

footer {
  background: #222;
  color: white;
  text-align: center;
  padding: 15px;
}

table {
  margin: 30px auto;
  width: 90%;
  border-collapse: collapse;
  margin-bottom: 30px;
  background-color: #fff;
  border-radius: 10px;
  box-shadow: 0px 4px 8px rgba(0,0,0,0.1);
}
```




USO DE JAVASCRIPT

La validación de formularios en JavaScript se implementó de forma robusta, especialmente visible en “funcionesRegistro.js”. Hemos creado un sistema de validación que incluye:

```
<label for="correo">Correo*</label>
<input type="email" id="correo" name="correo" maxlength="100" required placeholder="usuario@duoc.cl" />
<div style="font-size:0.95em; color:■ #555; margin-bottom:10px;">Solo se permiten correos @duoc.cl,

<label for="contrasena">Contraseña*</label>
<input type="password" id="contrasena" name="contrasena" minlength="4" maxlength="10" required style="width:100%;" />

<label for="fechaNacimiento">Fecha de nacimiento</label>
<input type="date" id="fechaNacimiento" name="fechaNacimiento" style="width:100%; margin-bottom:10px;" />

<label for="region">Región*</label>
<select id="region" name="region" required style="width:100%; margin-bottom:10px;"></select>

<label for="comuna">Comuna*</label>
<select id="comuna" name="comuna" required style="width:100%; margin-bottom:10px;"></select>
```

- Etiquetas asociadas: Cada input tiene su <label for="campo">
- Atributos de autocompletado: Utilizamos autocomplete="off" cuando es necesario y campos específicos como type="email"
- Validaciones HTML5 nativas: required, minlength, maxlength, type="date", type="email"
- Placeholders informativos: Como placeholder="Ej: 19011022K" para guiar al usuario

USO DE JAVASCRIPT

```
const regionesYcomunas = [
  {
    region: "Región Metropolitana",
    comunas: ["Santiago", "Maipú", "Puente Alto", "Las Condes", "La Florida"]
  },
  {
    region: "Valparaíso",
    comunas: ["Valparaíso", "Viña del Mar", "Quilpué", "Villa Alemana"]
  },
  {
    region: "Biobío",
    comunas: ["Concepción", "Talcahuano", "Coronel", "Los Ángeles"]
  }
];

// Validación de RUN chileno (sin puntos ni guion)
function validarRun(run) {
  if (!run) return false;
  if (run.length < 7 || run.length > 9) return false;
  return /^[0-9]{7,8}[0-9kK]$/.test(run);
}

// Validación de correo
function validarCorreo(correo) {
  if (!correo) return false;
  if (correo.length > 100) return false;
  return /^(duoc\.cl|profesor\.duoc\.cl|gmail\.com)$/.test(correo);
}
```

```
// Cargar regiones en el select
function cargarRegiones() {
  const selectRegion = document.getElementById('region');
  selectRegion.innerHTML = '<option value="">Selecione región</option>' +
    regionesYcomunas.map(r => `<option value="${r.region}">${r.region}</option>`);
}

// Cargar comunas según región
function cargarComunas() {
  const selectRegion = document.getElementById('region');
  const selectComuna = document.getElementById('comuna');
  const regionSeleccionada = selectRegion.value;
  selectComuna.innerHTML = '<option value="">Selecione comuna</option>';
  if (!regionSeleccionada) return;
  const regionObj = regionesYcomunas.find(r => r.region === regionSeleccionada);
  if (regionObj) {
    selectComuna.innerHTML += regionObj.comunas.map(c => `<option value="${c}">${c}</option>`);
  }
}
```




OPERACIONES EN REPOSITORIO COLABORATIVO

```
Proyecto 4 sem/Vistas/Administracion/dashboard.html

@@ -4,15 +4,15 @@
4      4      <meta charset="UTF-8">
5      5      <meta name="viewport" content="width=device-width,
6      6      initial-scale=1.0">
7      6      <title>Dashboard Administrador – Verdes-Souvenir</t
8      7      itle>
9      7      - <link rel="stylesheet" href="../../css/Estilos.cs
10     8      s">
11     7      + <link rel="stylesheet" href="../../css/EstilosAdmi
12     8      n.css">
13     8      </head>
14     9      - <body>
15     9      + <body class="admin-panel">
16     10     <aside class="sidebar">
17     11     <h2>Administrador</h2>
18     12     <ul>
19     13     <li><a href="dashboard.html">Dashboard</a></li>
20     14     <li><a href="../../Administracion/usuario.html">Usua
21     15     rios</a></li>
22     15     - <li><a href="../../Administracion/productos.html">Pr
23     16     oductos</a></li>
24     15     + <li><a href="../../Administracion/adm-prodcutos.htm
25     16     l">Productos</a></li>
26     16     <li><a href="../../Administracion/ordenes.html">Órde
27     16     nes</a></li>
```

El desarrollo de Verdes-Souvenir se realizó utilizando Git como sistema de control de versiones colaborativo. Las operaciones implementadas incluyen

Current Repository
Verdes-Souvenir

Current Branch
Jonathan

Changes 3

History

Correcciones visuales y rutas en ad...

jhonnycastle · 5 days ago

Merge pull request #5 from jhonnycastle

Euda · 5 days ago

arreglando mensaje

Euda-00 · 5 days ago

Merge pull request #4 from jhonnycastle

Euda · 6 days ago

Merge branch 'main' into Eugenio

Euda · 6 days ago

mensaje listo

Euda-00 · 6 days ago

viernes

Euda-00 · 6 days ago

usuario admin

Euda-00 · 8 days ago

barra lateral de admin

Euda-00 · 9 days ago

e

Euda-00 · 10 days ago

Correcciones visuales y rutas en ac

jhonnycastle · a884cbc · +1477 -2

17 changed files

Proyecto 4 sem/.DS_Store

Proyecto 4 sem/.adm-prodcutos.html

Proyecto 4 sem/Vistas/dashboard.html

Proyecto 4 sem/Vistas/editar-producto.html

Proyecto 4 sem/Vistas/mensaje.html

Proyecto 4 sem/Vistas/mostrar-producto.html

Proyecto 4 sem/Vistas/nuevo-producto.html

Proyecto 4 sem/Vistas/nuevousuario.html

Proyecto 4 sem/Vistas/usuario.html

Proyecto 4 sem/Vistas/carrito.html

Proyecto 4 sem/Vistas/productos.html

Proyecto 4 sem/Vistas/carrito.html

Proyecto 4 sem/Vistas/productos.html

Proyecto 4 sem/css/Estilos.css

Proyecto 4 sem/css/EstilosAdmin.css

Proyecto 4 sem/Vistas/funcionesCarrito.js

Proyecto 4 sem/Vistas/funcionesProductos.js

CONCLUSION

VERDES-SOUVENIR SE CONSTRUYÓ CON BUENAS PRÁCTICAS DE DESARROLLO WEB DE PUNTA A PUNTA: UNA ESTRUCTURA HTML5 SEMÁNTICA Y ACCESIBLE, ESTILOS CSS EXTERNOS Y MODULARES QUE GARANTIZAN CONSISTENCIA Y RENDIMIENTO, JAVASCRIPT QUE VALIDA Y GUÍA AL USUARIO CON LÓGICA PERSONALIZADA.

A NIVEL DE EQUIPO, EL USO DE GIT CON RAMAS, COMMITS DESCRIPTIVOS LO QUE PERMITIÓ UN FLUJO DE TRABAJO ORDENADO, TRAZABLE Y COLABORATIVO. EN CONJUNTO, ESTAS DECISIONES MEJORAN LA ESCALABILIDAD, EL MANTENIMIENTO Y LA EXPERIENCIA DE USUARIO, DEJANDO UNA BASE SÓLIDA PARA SEGUIR CRECIENDO FUNCIONAL Y VISUALMENTE EL PROYECTO.

