

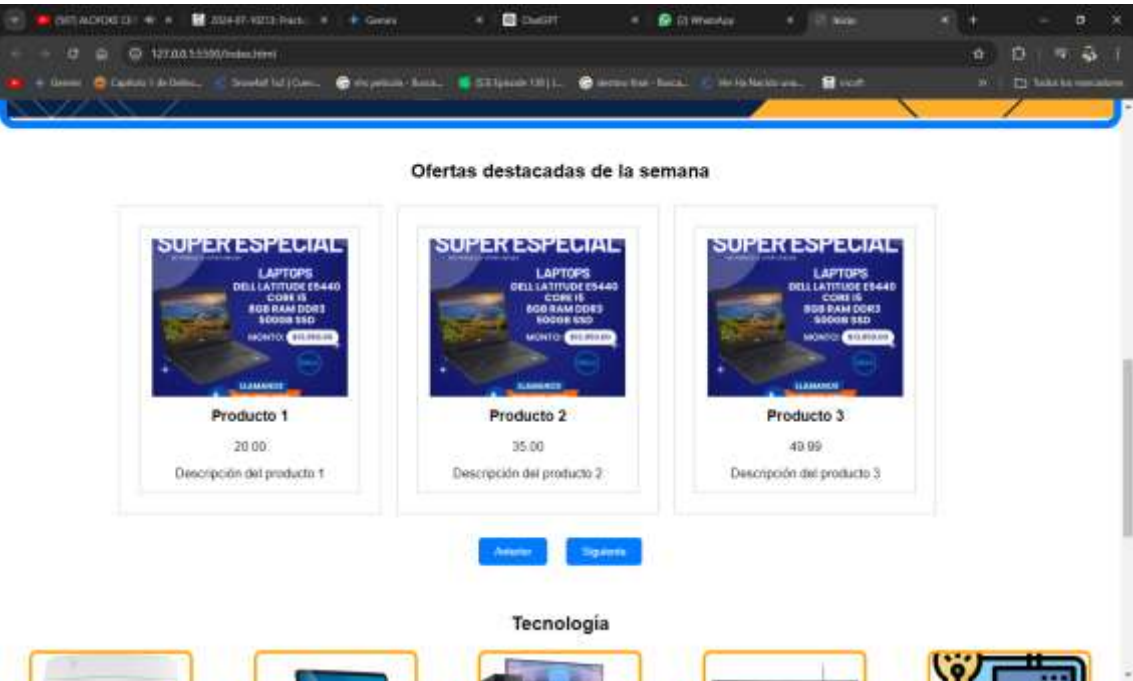
PROYECTO FINAL PAGINA WEB

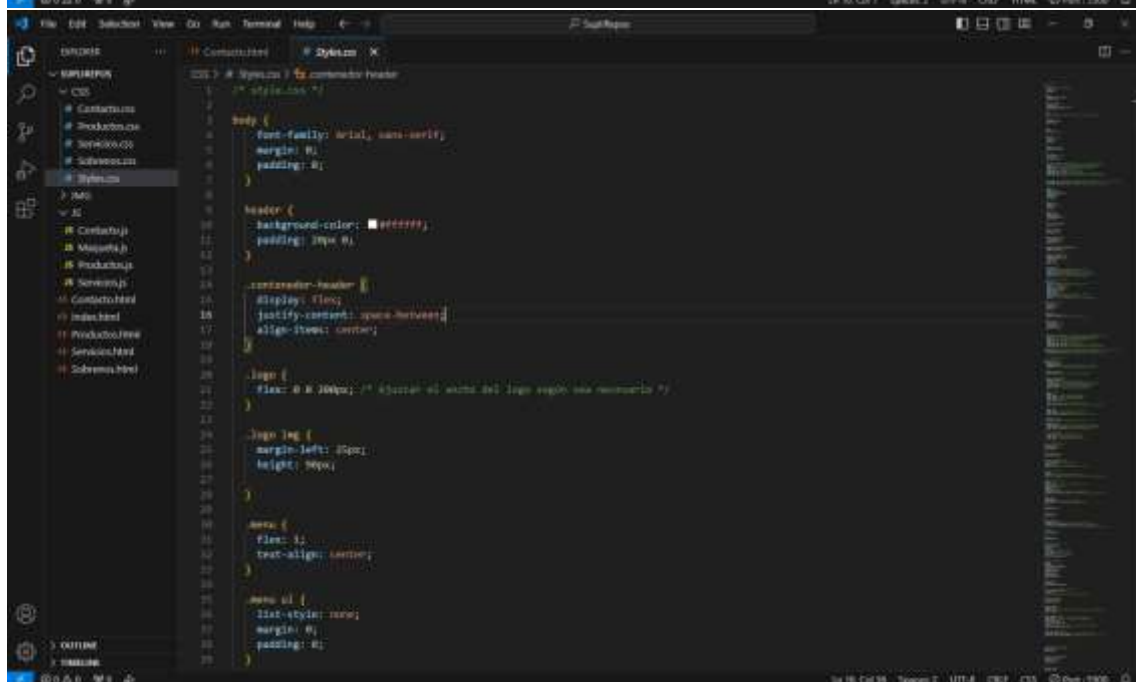
Francis Miguel Jimenez Polanco

Introducción

La página que se ha realizado y de la cual se muestra la documentación, es una página realizada en forma de plantilla sobre la empresa SUPLITECMO SRL., una empresa multiservicios la cual se especializa en informática, he realizado esta página en forma de plantilla ya que se me hace más fácil el hecho de realizar la página de tal forma para luego realizar modificaciones de contenido y mantener el diseño original, sin embargo, esta página se encuentra en fase beta.

Página Index





```
File Edit Selection View Go Run Terminal Help SvelteKit
-- Styles.css --
CSS > Styles.css > @container header
46
47 menu li a {
48   text-decoration: none;
49   color: #000000;
50   font-weight: bold;
51 }
52
53 menu li:hover:after { /* pseudo-element "after" para la linea naranja */
54   content: ""; /* contenido vacío para que la linea se haga texto */
55   display: block; /* convertir el pseudo-elemento en un bloque */
56   width: 100%; /* ancho completo del elemento del menú */
57   height: 1px; /* altura de la linea */
58   background-color: #ff8c00; /* color naranja */
59   margin-top: 1px; /* margen superior para separar la linea del texto */
60   transition: width 0.3s ease; /* animación suave para el cambio de ancho */
61 }
62
63 menu li:hover a { /* Añadir estilo al texto dentro del elemento del menú al pasar el cursor */
64   font-size: 1.2em; /* aumentar el tamaño del texto en un 20% */
65   transition: font-size 0.3s ease; /* animación suave para el cambio de tamaño */
66 }
67
68 .submenu {
69   display: none;
70   position: absolute;
71   background-color: #f0f0f0;
72   padding: 5px;
73   left: -10px;
74   top: 10px;
75 }
76
77 .submenu li {
78   list-style-type: none;
79   margin: 0;
80   padding: 0;
81 }
82
83 .submenu li a {
84   display: block;
```

```
File Edit Selection View Go Run Terminal Help SvelteKit
-- Styles.css --
CSS > Styles.css > @container header
118
119 .nav-container {
120   width: 100%; /* Make the container take full width */
121 }
122
123 /* Main navigation styles */
124 .main-nav {
125   position: absolute; /* Position relative to parent */
126   top: 0; /* Top position */
127   right: 0; /* Right position */
128   background-color: #f0f0f0;
129   border: 1px solid #ccc;
130   padding: 5px 0;
131   display: flex;
132   justify-content: flex-end; /* Align items to the right */
133   width: 100%; /* Adjust width as needed */
134   z-index: 1; /* Ensure navigation is on top of other elements */
135 }
136
137 /* Navigation links */
138 .main-nav ul {
139   list-style-type: none;
140   margin: 0;
141   padding: 0;
142 }
143
144 .main-nav li {
145   display: inline-block;
146   margin-left: 10px; /* Add spacing between links */
147 }
148
149 .main-nav li a {
150   text-decoration: none;
151   color: #000000;
152   font-weight: bold;
153   padding: 5px 10px;
```

```

File Edit Selection View Git Run Terminal Help
SustRapp

CSS # Stylesheet
124 // Stylesheet for carousel header
125
126 // Main container
127
128 // Main navigation styles
129
130 main-nav {
131   position: absolute; /* Position relative to parent */
132   top: 0; /* Top position */
133   right: 0; /* Right position */
134   background-color: #fff;
135   border: 1px solid #ccc;
136   padding: 10px 0;
137   display: flex;
138   justify-content: flex-end; /* Align items to the right */
139   width: 100%; /* Adjust width as needed */
140   z-index: 1; /* Ensure navigation is on top of other elements */
141 }
142
143 // Navigation links
144 main-nav ul {
145   list-style-type: none;
146   margin: 0;
147   padding: 0;
148 }
149
150 main-nav li {
151   display: inline-block;
152   margin-left: 10px; /* Add spacing between links */
153 }
154
155 main-nav li a {
156   text-decoration: none;
157   color: #333;
158   font-weight: bold;
159   padding: 10px 10px;
160 }

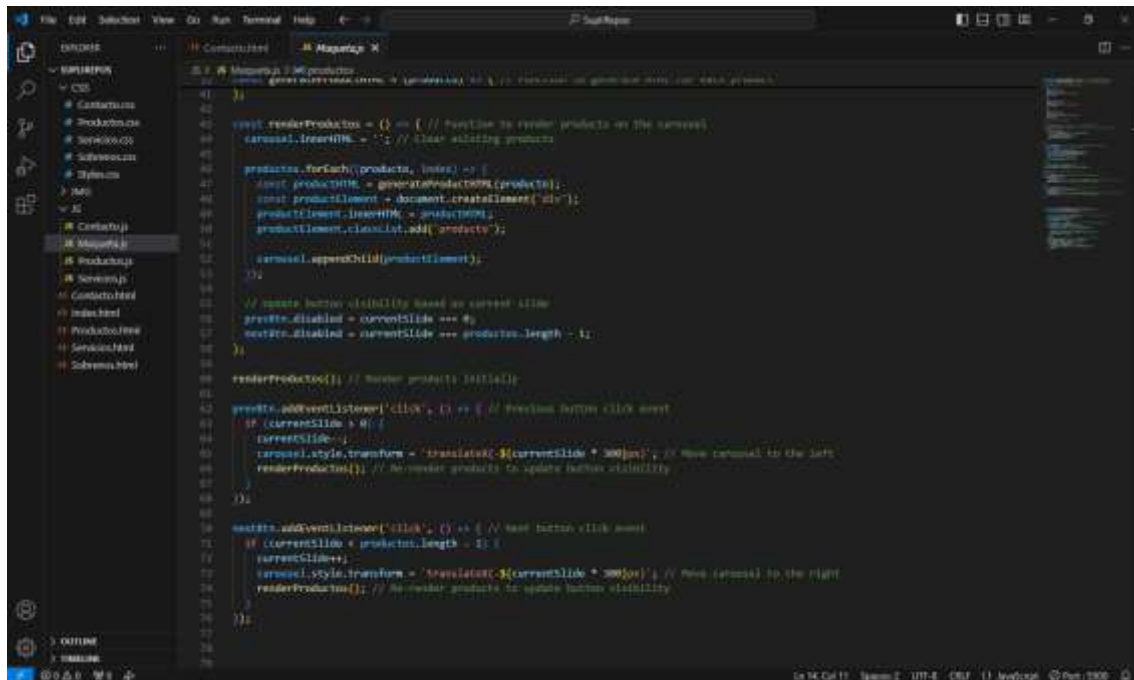
```

```

File Edit Selection View Git Run Terminal Help
SustRapp

JS # Mapape.js
1 // Mapape.js
2
3 const carousel = document.querySelector('.carousel'); // Get the carousel element
4 const prevBtn = document.querySelector('.prev-btn'); // Get the previous button element
5 const nextBtn = document.querySelector('.next-btn'); // Get the next button element
6
7 const products = [ // Array of product objects
8   {
9     id: 1,
10    img: 'img/Producto1.jpg',
11    titulo: 'Producto 1',
12    descripcion: 'Descripción del producto 1',
13    precio: 20.00
14  },
15   {
16     id: 2,
17    img: 'img/Producto2.jpg',
18    titulo: 'Producto 2',
19    descripcion: 'Descripción del producto 2',
20    precio: 30.00
21  },
22   {
23     id: 3,
24    img: 'img/Producto3.jpg',
25    titulo: 'Producto 3',
26    descripcion: 'Descripción del producto 3',
27    precio: 40.00
28  }
29 ];
30
31 // Add more product objects here
32
33 let currentSlide = 0; // Track the current slide (index starts at 0)
34
35 const generateProductHTML = (products) => { // Function to generate HTML for each product
36   return
37     `<div class="producto">
38       
39       <h3>${products[titulo]}</h3>
40       <p>${products[descripcion]}</p>
41       <p>${products[precio]}</p>
42     </div>`;
43 }

```

El HTML del código define la estructura básica de la página web, organizando su contenido en diferentes secciones. Estas incluyen un encabezado con elementos como una barra de navegación, un área de inicio de sesión, un carrito de compras y una barra de búsqueda. Además, se despliegan productos destacados, un banner publicitario, un carrusel con ofertas especiales, y una galería de marcas reconocidas. Cada una de estas secciones está organizada utilizando etiquetas HTML estándar como ``, ``, `

`, `

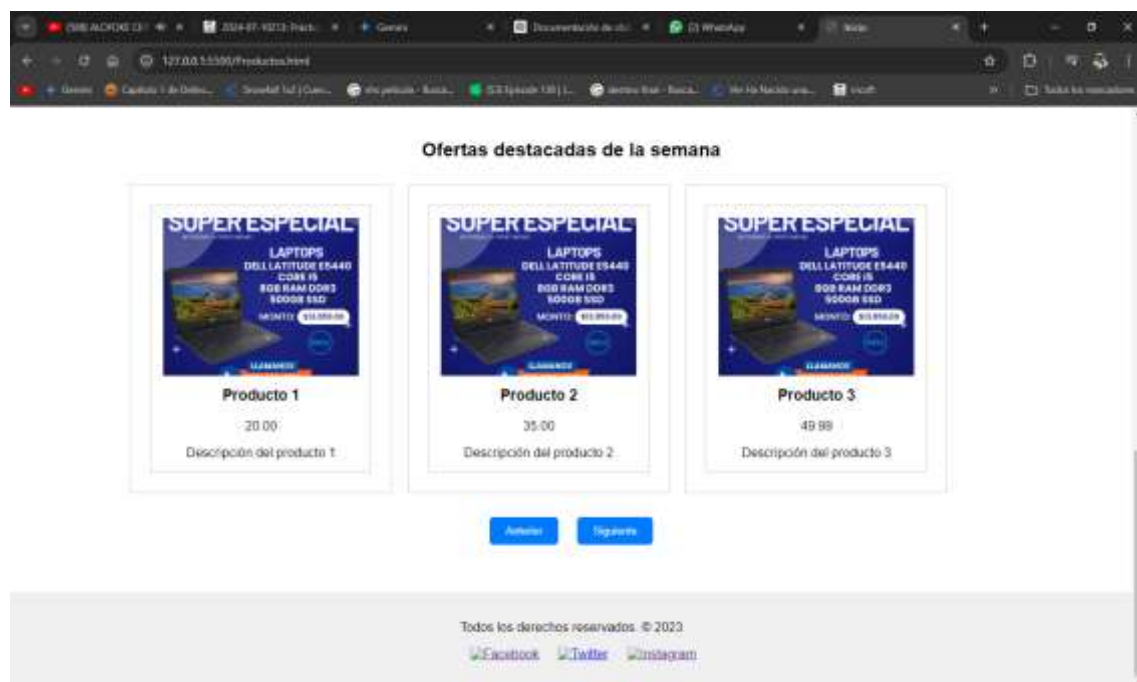
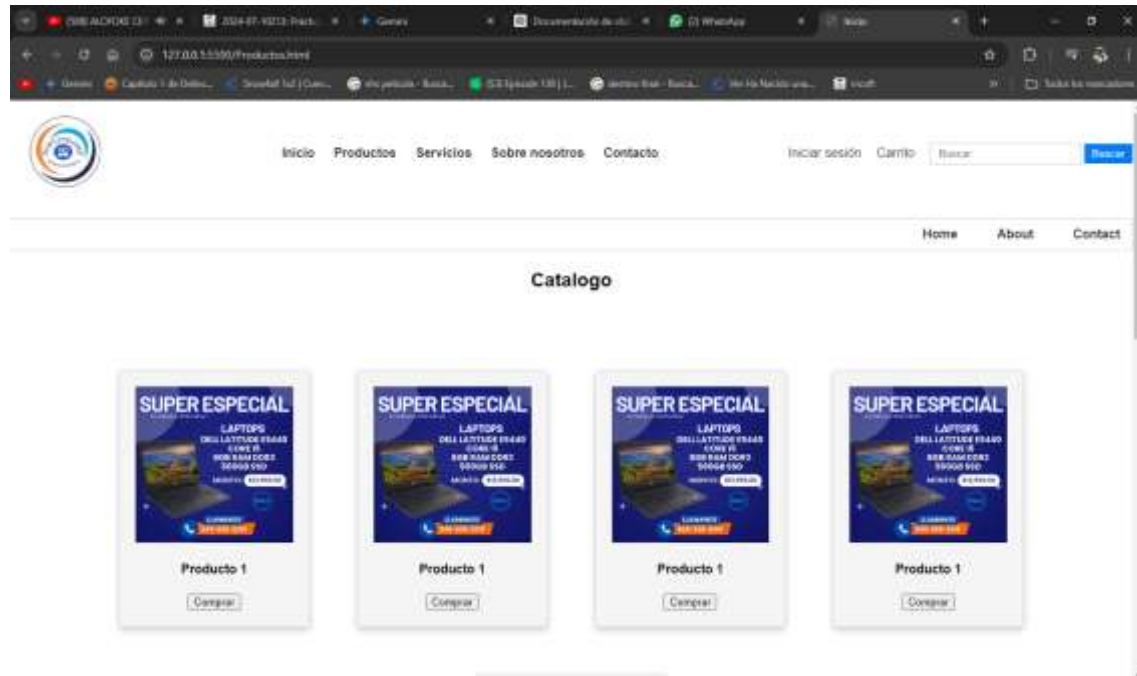
`, y ``, proporcionando una estructura clara y semántica para la página.

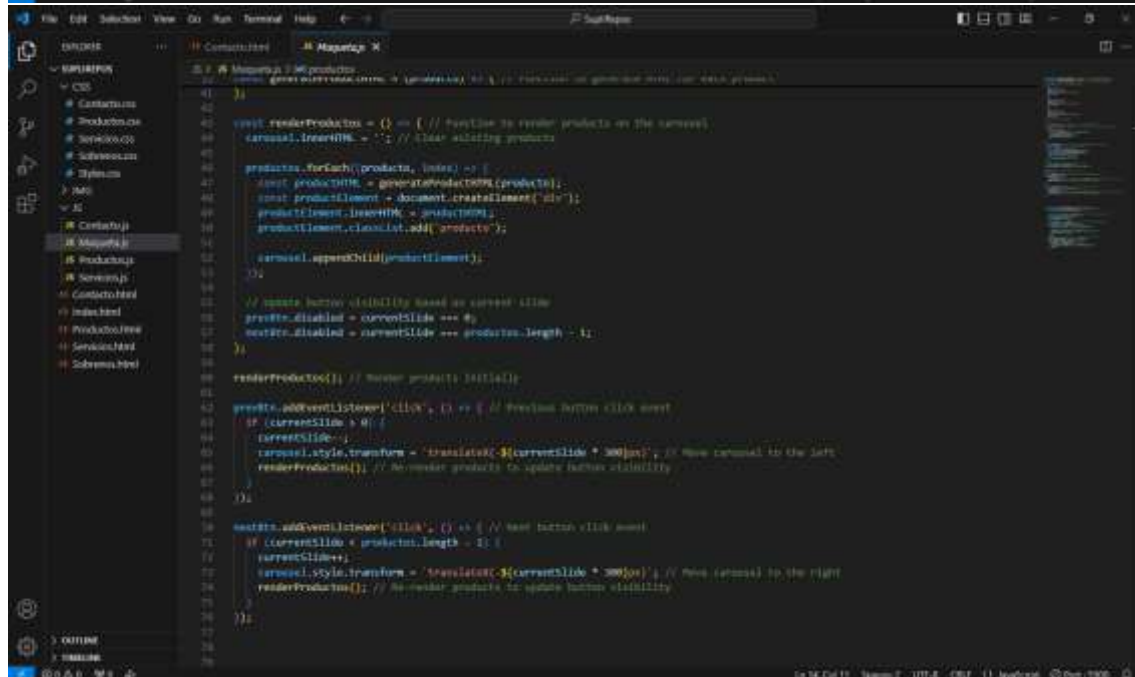
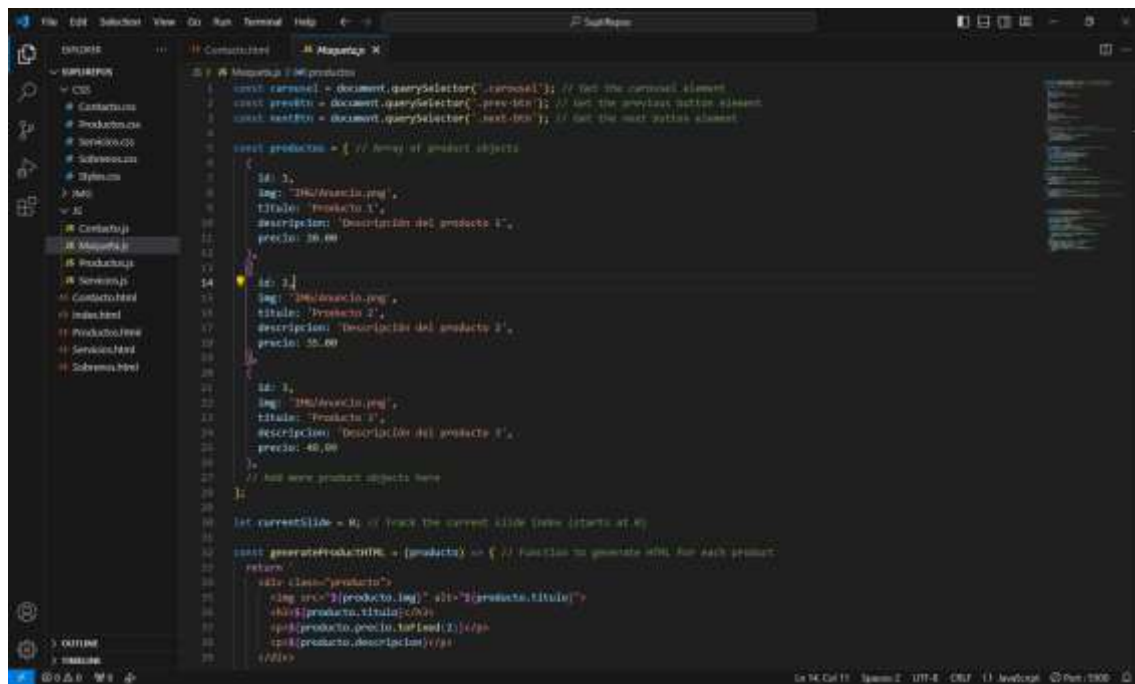
El CSS se encarga de la apariencia visual de la página, utilizando flexbox para organizar los elementos de manera eficiente y asegurando que el diseño sea responsivo y se adapte a diferentes dispositivos. Se aplican estilos específicos para cada sección, como colores de fondo, márgenes, tamaños de fuente, y disposición de los elementos. Además, se incluyen efectos visuales, como el agrandamiento de productos y la adición de sombras al pasar el cursor, lo que mejora la estética de la página y la experiencia del usuario.

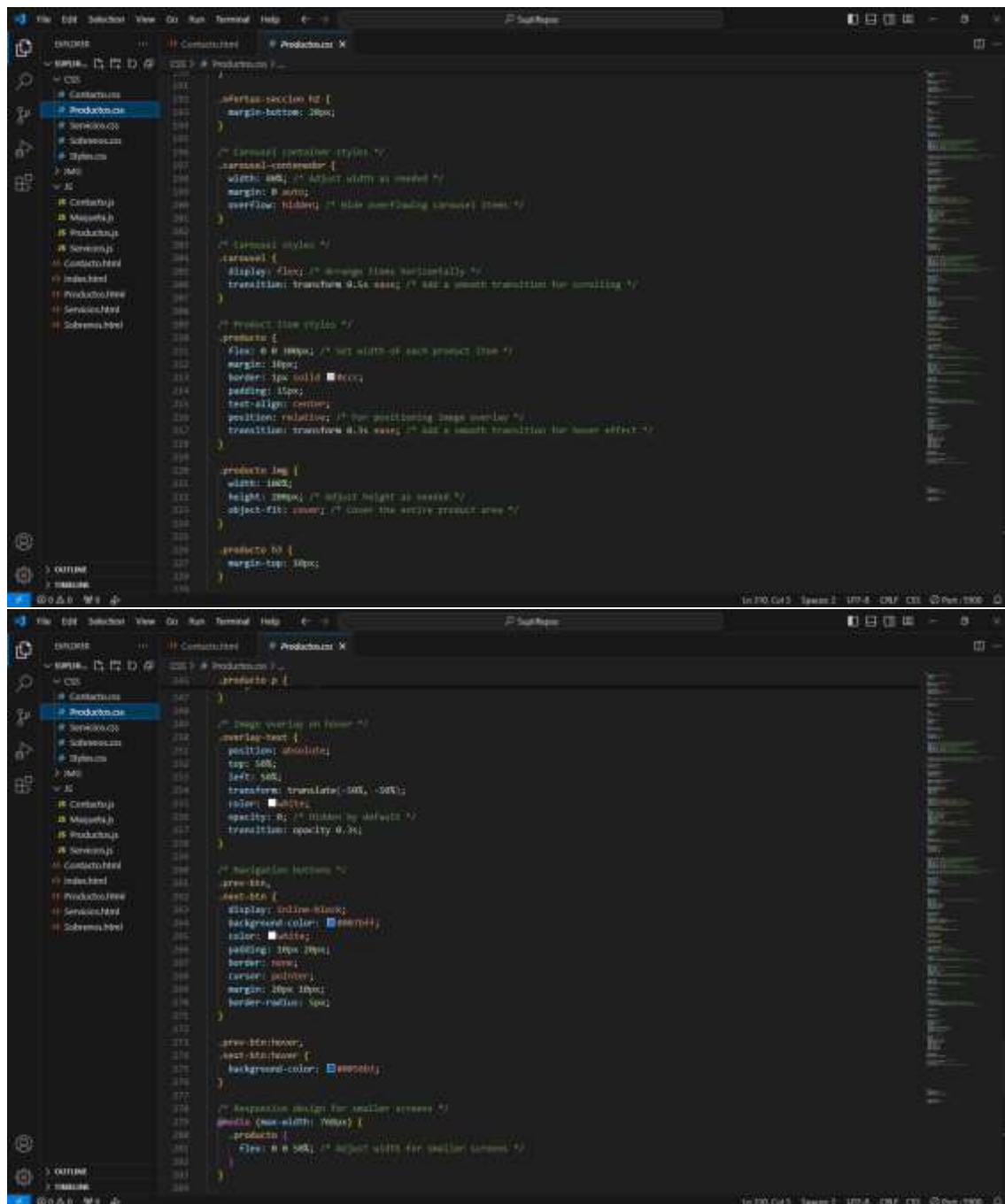
Por último, el JavaScript añade interactividad a la página, controlando la navegación en el carrusel de productos y la rotación automática de imágenes en el deslizador. También maneja eventos como clics y desplazamientos, permitiendo que el usuario interactúe de manera dinámica con el contenido. Este uso de JavaScript convierte a la página en una experiencia más atractiva y fluida, mejorando su funcionalidad y manteniendo a los usuarios comprometidos.

Página productos

PD: Se mostrara el body HTML y css ya que mantiene la estructura principal del menú.





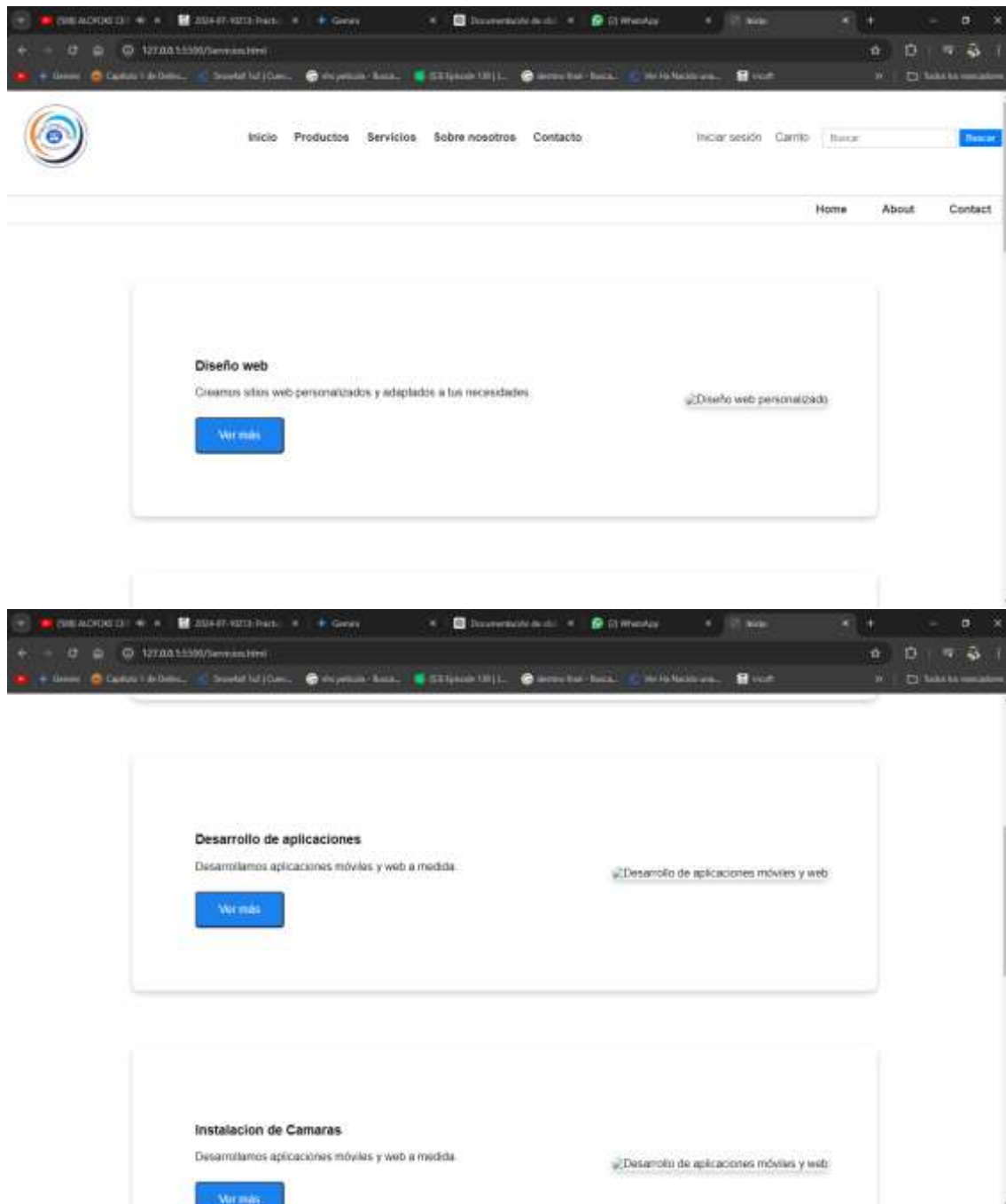


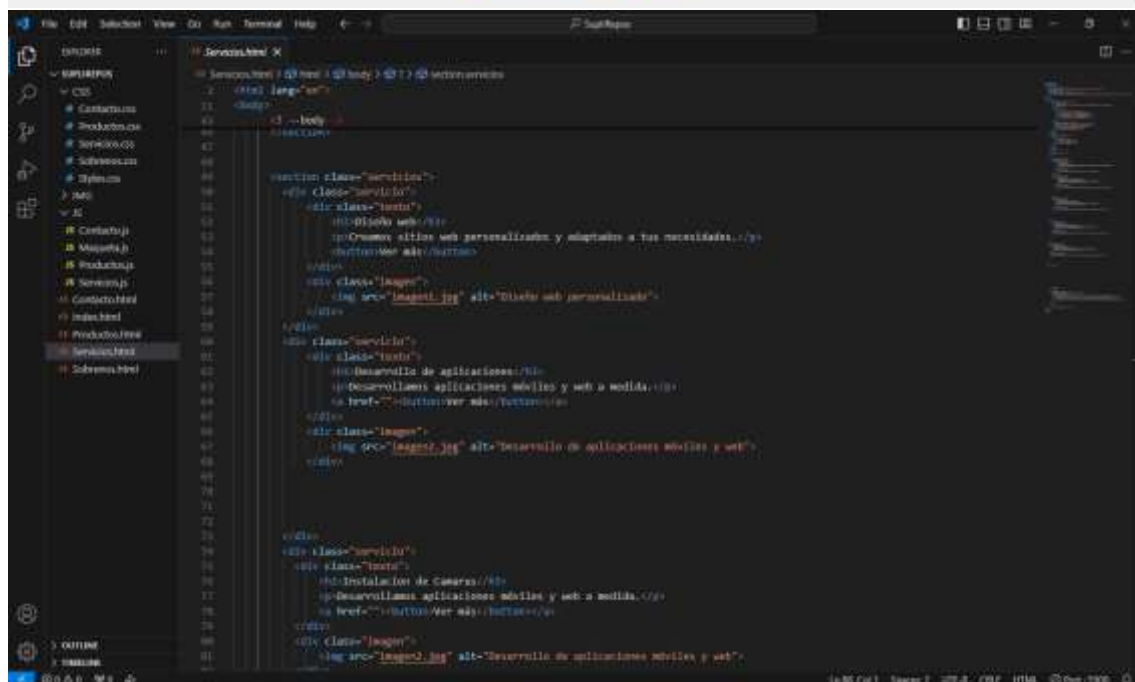
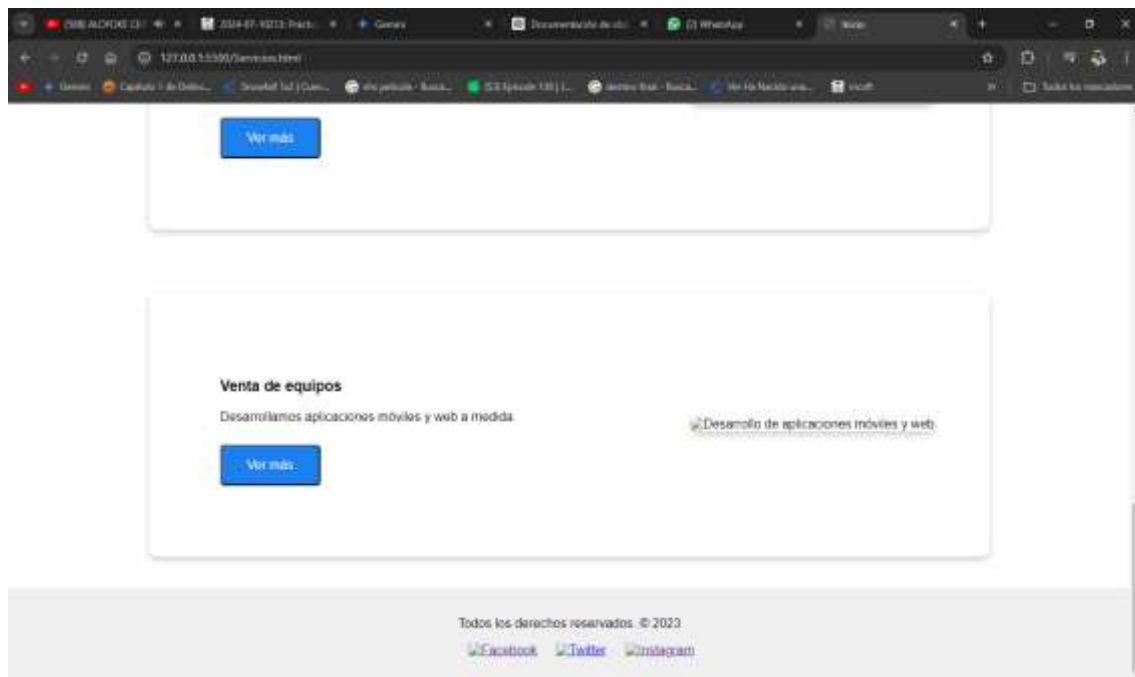
En el HTML, la estructura de la página de productos está organizada en varias secciones clave: el encabezado, el catálogo de productos, una sección de ofertas destacadas y el pie de página. El encabezado incluye el logo de la empresa, un menú de navegación y elementos adicionales como enlaces para iniciar sesión, acceder al carrito de compras y realizar búsquedas. A continuación, el catálogo de productos se presenta en un contenedor flexible con tarjetas de productos, cada una con una imagen, un título y un botón de compra. Finalmente, la sección de ofertas destaca un carrusel para mostrar productos en oferta y el pie de página proporciona información de derechos reservados y enlaces a redes sociales.

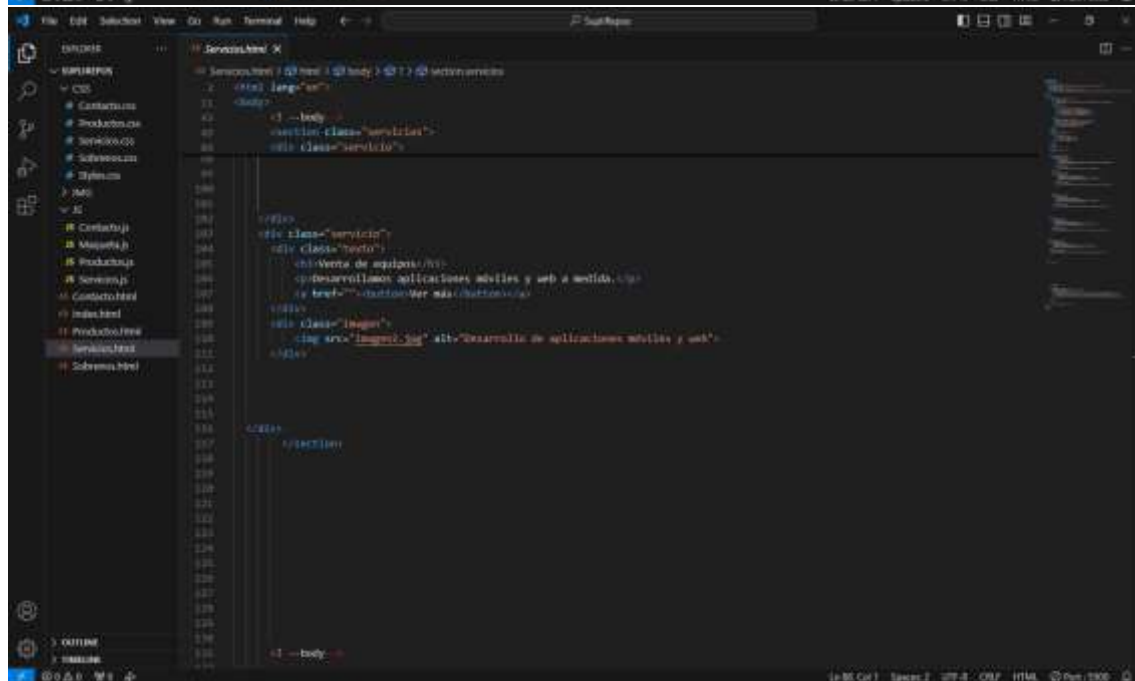
El CSS define el estilo visual de la página. Utiliza flexbox para la disposición de los elementos y establece un diseño responsivo que se adapta a diferentes tamaños de pantalla. Los estilos para el encabezado y el menú de navegación se enfocan en la alineación y la interacción, con efectos como el cambio de tamaño y la adición de una línea de color naranja en el menú. Las tarjetas de producto están diseñadas con un fondo claro, sombra y un efecto de escala en el hover para mejorar la apariencia. La sección de ofertas y el carrusel tienen estilos que permiten una presentación atractiva y funcional de los productos destacados.

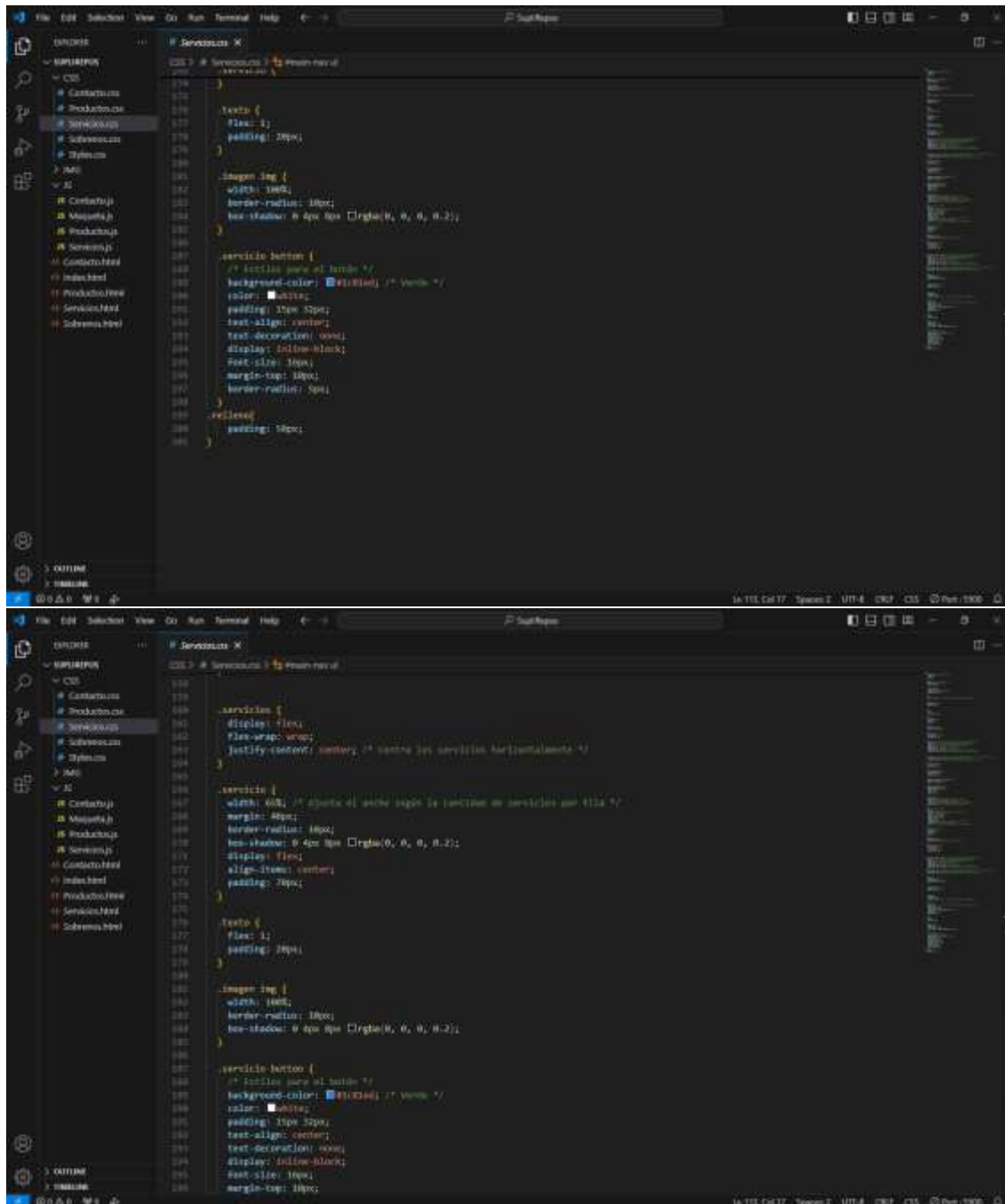
El JavaScript gestiona la interactividad en la página, especialmente el carrusel de productos. Se define un arreglo de objetos de productos con detalles como imagen, título, descripción y precio. La función `generateProductHTML` crea el HTML para cada producto y `renderProductos` se encarga de renderizar estos productos en el carrusel. Los botones de navegación permiten al usuario desplazarse entre los productos en el carrusel, con lógica para habilitar o deshabilitar los botones según la posición actual del carrusel. Este script asegura que la interfaz sea dinámica y que los productos se presenten de manera fluida.

Página servicios







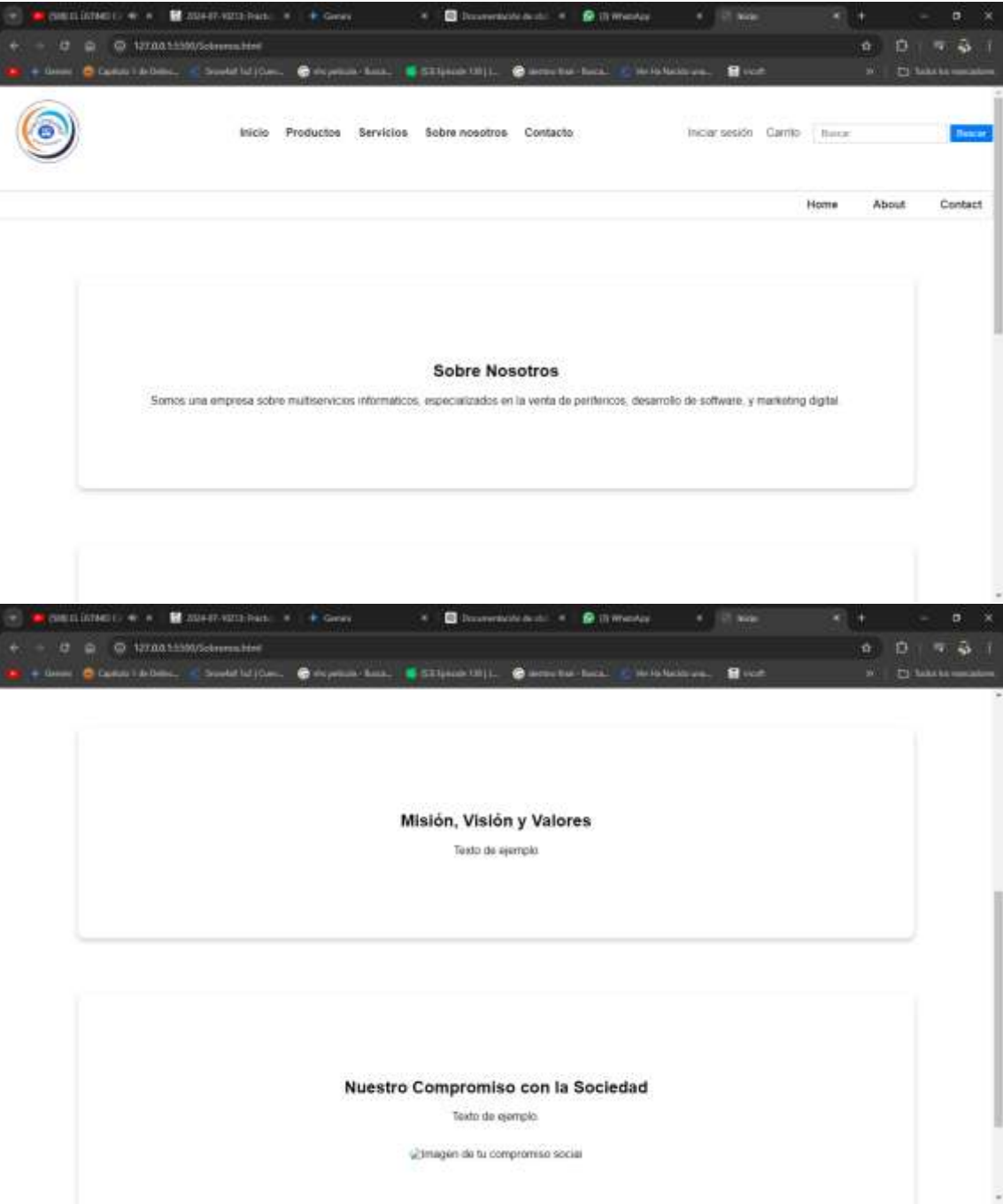


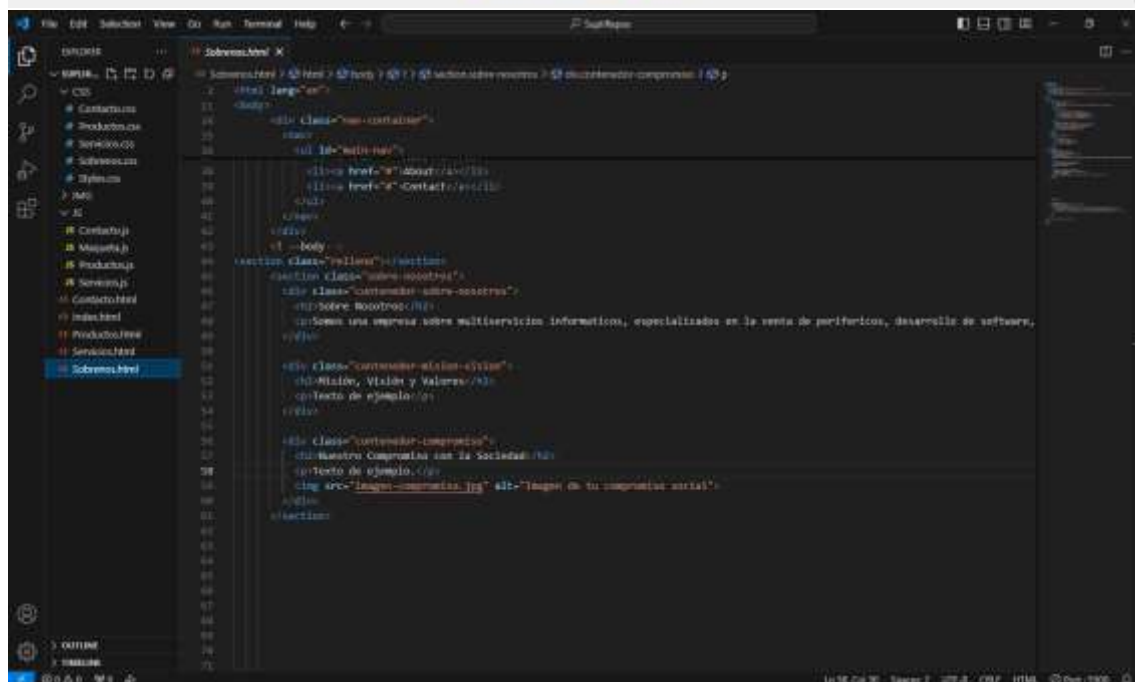
Fuera del menú, el contenido del <body> se divide en varias secciones. La primera sección destacada es el <section> con la clase `relleno`, que actúa como un espacio de relleno con un padding de 50px para proporcionar separación visual en la página. A continuación, se presenta la sección `servicios`, que utiliza un diseño de flexbox para mostrar una serie de bloques de servicios. Cada bloque de servicio (`.servicio`) está diseñado para ser una tarjeta con un ancho del 65% de su contenedor, margen de 40px y un diseño flexible que alinea su contenido centralmente. Los bloques tienen un fondo blanco, bordes redondeados, y una sombra para darles un efecto de elevación.

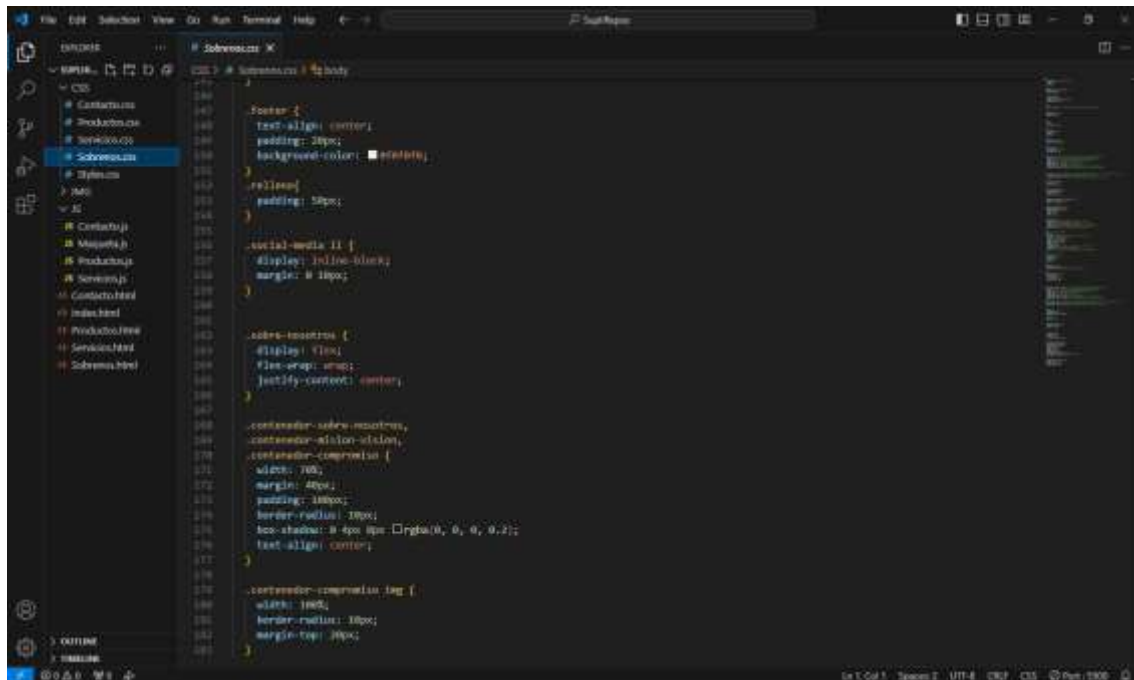
Cada bloque de servicio se divide en dos partes: `.texto` y `.imagen`. La clase `.texto` contiene la descripción del servicio y un botón para obtener más información, mientras que la clase `.imagen` muestra una imagen relacionada con el servicio. Los botones están estilizados con un fondo azul y texto blanco, con un padding de 15px en vertical y 32px en horizontal, además de

un borde redondeado. La imagen dentro de `.imagen` está configurada para ocupar el 100% del ancho del contenedor con bordes redondeados y una sombra similar a la del contenedor principal.

Página Sobre nosotros







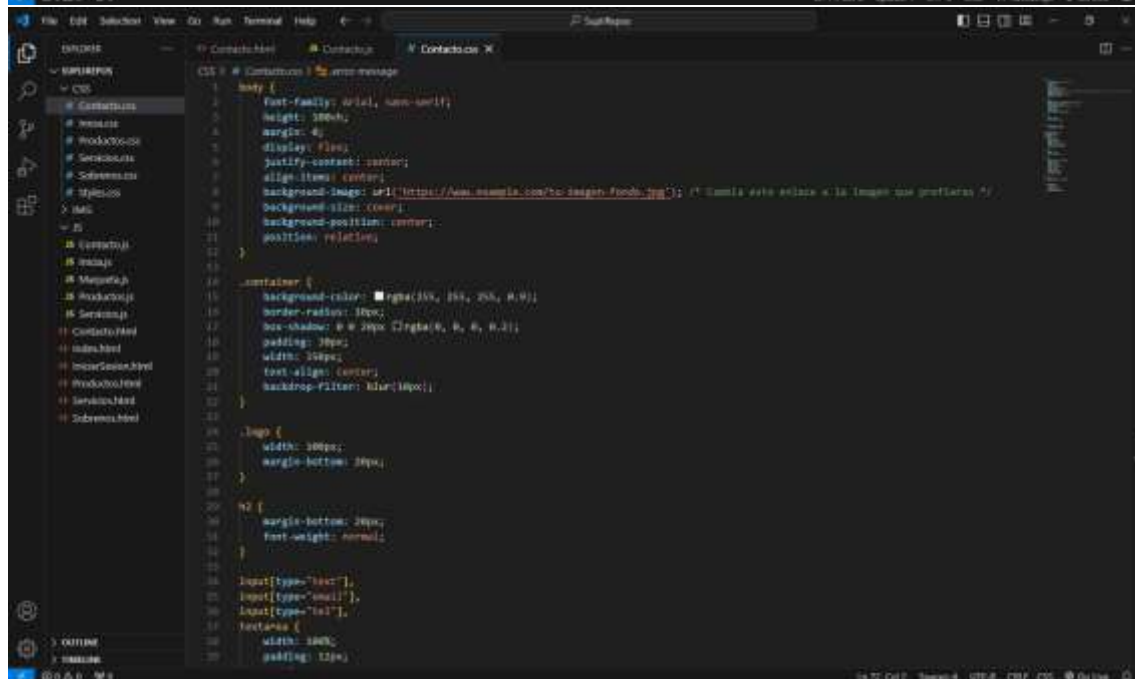
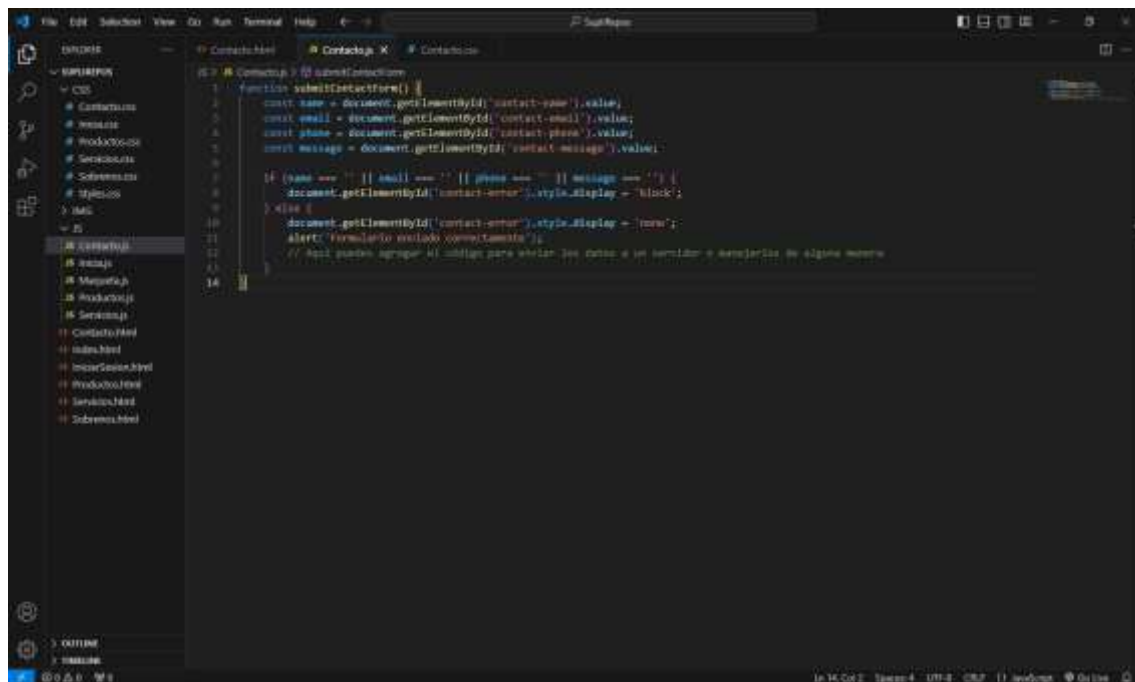
La página "Sobre Nosotros" está organizada en varias secciones clave para presentar información esencial sobre la empresa. La estructura HTML incluye un bloque principal dentro de `<section class="sobre-nosotros">`, que contiene tres contenedores: `.contenedor-sobre-nosotros`, `.contenedor-mision-vision`, y `.contenedor-compromiso`. Cada contenedor ofrece una sección específica con encabezados y párrafos para explicar la empresa, su misión y visión, y su compromiso social, respectivamente. Además, el contenedor de compromiso incluye una imagen para ilustrar visualmente el compromiso de la empresa.

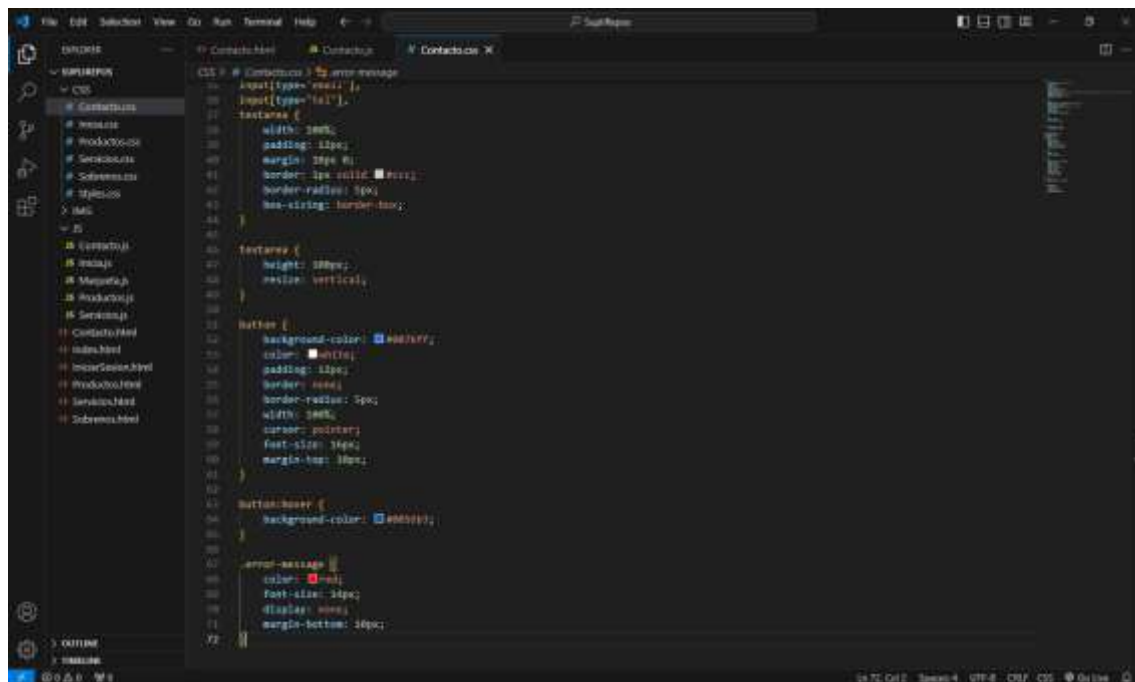
En términos de diseño, el CSS aplica estilos coherentes para mantener una presentación limpia y profesional. Los contenedores tienen un ancho del 70% y están centrados con márgenes de 40px y padding de 100px. La sombra de caja y los bordes redondeados mejoran la estética, mientras que la imagen en el bloque de compromiso social se ajusta al 100% del ancho del contenedor con bordes redondeados. Estos estilos aseguran que el contenido sea visualmente atractivo y legible en diferentes dispositivos, proporcionando una experiencia de usuario efectiva.

Página Contacto



```
File Edit Selection View Go Run Terminal Help P Sublime
Contacto.html X Contacto Contacto
Contacto.html < @html > @body < @html
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Inicio</title>
7 <link rel="stylesheet" href="CSS/Contacto.css">
8 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" integrity="sha384-11mZ9q1ES29+JqcbWNbmivk0W7e28+iR3KzR3r9FLeFNB" crossorigin="anonymous">
9 </head>
10 <body>
11
12 <div class="container">
13 <div src="img/logo.jpg" alt="Logo de la empresa" class="img"> <!-- Cambia este enlace al logo de tu empresa -->
14 </div>
15 <div class="error-message" id="contact-error">Por favor, completa todos los campos.</div>
16 <input type="text" id="contact-name" placeholder="Nombre" required>
17 <input type="email" id="contact-email" placeholder="Correo electrónico" required>
18 <input type="tel" id="contact-phone" placeholder="Teléfono" required>
19 <div class="id" id="contact-message" placeholder="Mensaje" required></div>
20 <button onclick="submitContactForm()">Enviar</button>
21 </div>
22
23 <script src="JS/Contacto.js"></script>
24
25 </body>
26 </html>
```

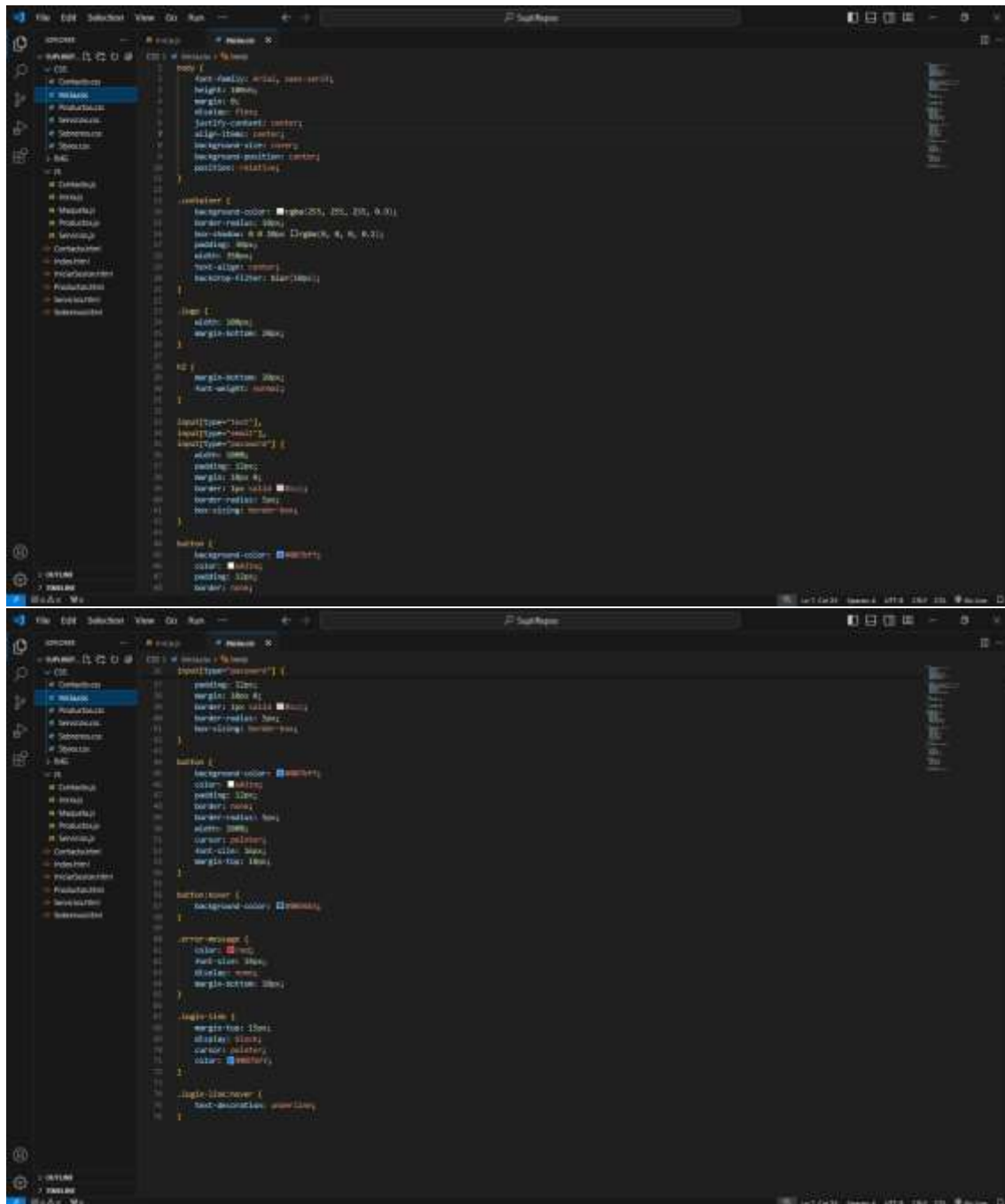




El script JavaScript vinculado al formulario valida que todos los campos estén completos antes de permitir el envío. Si algún campo no está rellenado, la función `submitContactForm()` muestra un mensaje de error al usuario utilizando la propiedad `display` para hacer visible el mensaje oculto. En caso de que todos los campos estén correctamente completados, el mensaje de error se oculta y se muestra una alerta indicando que el formulario se ha enviado correctamente. Esta funcionalidad básica se puede extender para enviar los datos a un servidor.

En cuanto a los estilos CSS, la página presenta un diseño moderno y limpio, con una imagen de fondo que cubre toda la vista utilizando `background-size: cover`. El contenedor del formulario tiene un fondo semitransparente y bordes redondeados, además de una sombra que le da un aspecto elevado sobre el fondo. Los campos de entrada y el botón de envío están estilizados para que ocupen el ancho completo del contenedor y sean visualmente atractivos y fáciles de usar. La estructura y los estilos CSS aseguran que la página sea tanto estética como funcional.

[illegible]



Esta página de inicio de sesión está diseñada para ofrecer una experiencia de usuario fluida y segura, combinando una interfaz intuitiva con una lógica de validación robusta. La estructura HTML incluye dos formularios principales: uno para el registro de nuevos usuarios y otro para el inicio de sesión, que se muestran de forma alternada según la interacción del usuario. Los formularios están acompañados de mensajes de error que alertan al usuario en caso de datos incompletos o incorrectos, mejorando así la precisión y la seguridad en el proceso de autenticación.

El diseño visual de la página se centra en la simplicidad y la accesibilidad, con un fondo difuminado que mantiene la atención en los elementos clave del formulario. Los campos de entrada están claramente delineados y son suficientemente amplios para facilitar la

interacción en dispositivos móviles y de escritorio. El botón de envío está resaltado con un color azul vibrante, que cambia de tono al pasar el cursor, indicando su interactividad. Además, los mensajes de error se muestran en un rojo llamativo para captar la atención del usuario de inmediato.

La lógica de la página, implementada en JavaScript, incluye funciones para alternar entre los formularios de registro e inicio de sesión, así como para gestionar los intentos de inicio de sesión fallidos. Después de tres intentos incorrectos, se cierra la página automáticamente, lo que añade una capa adicional de seguridad. Esta lógica está diseñada para ser simple pero efectiva, permitiendo futuras expansiones o integraciones con sistemas de autenticación más complejos.

Conclusión

Esta pagina se encuentra en fase BETA, por lo cual aun faltan muchos detalles por pulir, sin embargo se mantiene la plantilla de la pagina y se demuestra el propósito con el cual fue realizada esta página, con un diseño sutil pero efectivo y llamativo para el usuario, recalando el hecho de que se están realizando varios servicios informáticos para el usuario.