**Aplicación de Tecnologías Web**

Tema n.° 3

CSS Avanzado

**Índice**

**Pág.**

[Introducción 3](#_Toc161254968)

[1.3. Formularios 4](#_Toc161254969)

[1.3.1. Elementos Básicos de un Formulario 4](#_Toc161254970)

[1.3.2. Elementos Específicos del Formulario 5](#_Toc161254971)

[1.3.3. Validación y Envío 6](#_Toc161254972)

[1.3.4. Helpers 9](#_Toc161254973)

[1.3.5. Componentes 12](#_Toc161254974)

[1.3.6. Templates 16](#_Toc161254975)

[Recursos complementarios 23](#_Toc161254976)

[Referencias 24](#_Toc161254977)

[Autoevaluación 25](#_Toc161254978)

# Introducción

El desarrollo web ha experimentado una rápida evolución en las últimas décadas, impulsado en gran medida por la demanda de interfaces más interactivas y atractivas. En este contexto, CSS (Cascading Style Sheets) ha emergido como una herramienta fundamental para el diseño y la presentación de contenido en la web. Sin embargo, el dominio de CSS avanzado va más allá de la mera estilización de elementos HTML, abarcando aspectos cruciales como formularios, helpers, componentes y templates.

Los formularios son elementos esenciales en cualquier aplicación web, permitiendo la interacción entre el usuario y el sistema. En el ámbito de CSS, se exploran técnicas para personalizar la apariencia y funcionalidad de formularios, optimizando la experiencia del usuario y la accesibilidad.

Los helpers constituyen una serie de clases y utilidades predefinidas que facilitan el desarrollo y mantenimiento del código CSS. Estas herramientas abstraen tareas comunes, como el manejo de espaciados, tipografía y alineación, simplificando así el proceso de diseño y garantizando una mayor coherencia visual.

Los componentes y templates son elementos modulares que permiten estructurar y reutilizar el código CSS de manera eficiente. Desde menús desplegables hasta tarjetas de producto, estos componentes encapsulan funcionalidades específicas y facilitan su integración en diversos contextos, promoviendo la cohesión y consistencia del diseño.

En esta sección se abordan las técnicas y mejores prácticas asociadas con CSS avanzado, con un enfoque particular en formularios, helpers, componentes y templates. Mediante ejemplos prácticos, se busca proporcionar una comprensión sólida y actualizada de este aspecto crucial del desarrollo web moderno.

# Formularios

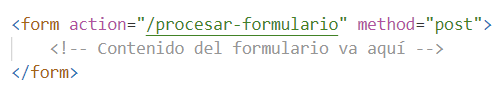
Los formularios son elementos cruciales en el desarrollo web, permitiendo la interacción del usuario con la página. Su diseño efectivo y atractivo es esencial para mejorar la experiencia del usuario. En esta sección, exploraremos los elementos y propiedades de los formularios en CSS y cómo utilizarlos para crear formularios atractivos y funcionales.

## *Elementos Básicos de un Formulario*

**Etiquetas <form>.** es el contenedor principal de un formulario. Envuelve todos los elementos del formulario.

**Figura 1**

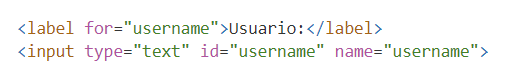
Uso de la etiqueta form.



**Etiquetas <input>.** La etiqueta <input> se utiliza para crear campos de entrada como cajas de texto, contraseñas, casillas de verificación, entre otros.

**Figura 2**

Uso de la etiqueta input

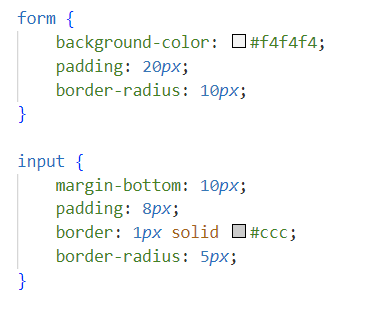


***Propiedades y Estilos en CSS aplicables a los formularios.***

**Estilización Básica:**Aplicar estilos básicos mejora la apariencia del formulario. Puedes cambiar colores, márgenes y bordes según el diseño deseado.

**Figura 3**

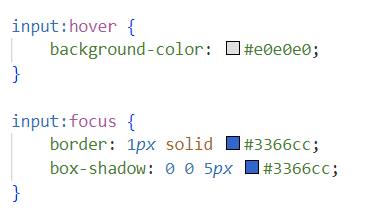
Estilización Básica de un formulario mediante selectores de CSS

****

**Pseudoclases para Interactividad:** hover y focus permiten estilos interactivos cuando el usuario interactúa con el formulario.

**Figura 4**

Uso de pseudoclases para Interactividad

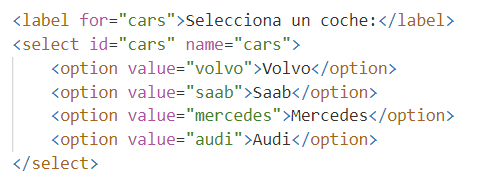


## *Elementos Específicos del Formulario*

**Etiqueta <select>:**se utiliza para crear menús desplegables.

**Figura 5**

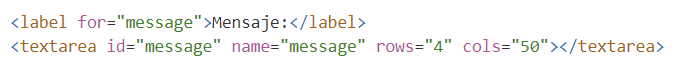
Uso de la etiqueta <select>



**Etiqueta <textarea>:** permite a los usuarios ingresar texto en áreas más grandes.

**Figura 6**

Uso de la etiqueta <textarea> en html

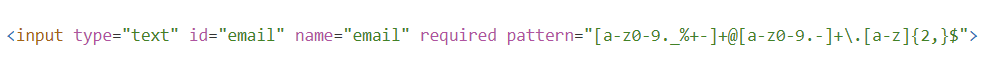


## *Validación y Envío*

**Validación del lado del cliente:**Utiliza atributos como required y pattern para realizar validaciones básicas del lado del cliente.

**Figura 7**

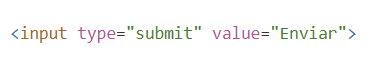
Validación del lado del cliente



**Botón de Envío:**Agrega un botón de envío para enviar el formulario al servidor.

**Figura 8**

Estructura del boton enviar



A continuación, se presenta un ejemplo integrado de los elementos utilizados anteriormente.

**Tabla 1**

Código del archivo index.html

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <link rel="stylesheet" href="estilos.css">      <title>Formulario de Contacto</title>  </head>  <body>      <div class="container">          <form action="#" method="post" class="contact-form">              <h2>Contacto</h2>              <label for="name">Nombre:</label>              <input type="text" id="name" name="name" required>              <label for="email">Correo Electrónico:</label>              <input type="email" id="email" name="email" required>              <label for="message">Mensaje:</label>              <textarea id="message" name="message" rows="4" required></textarea>              <input type="submit" value="Enviar Mensaje">          </form>      </div>  </body>  </html> |

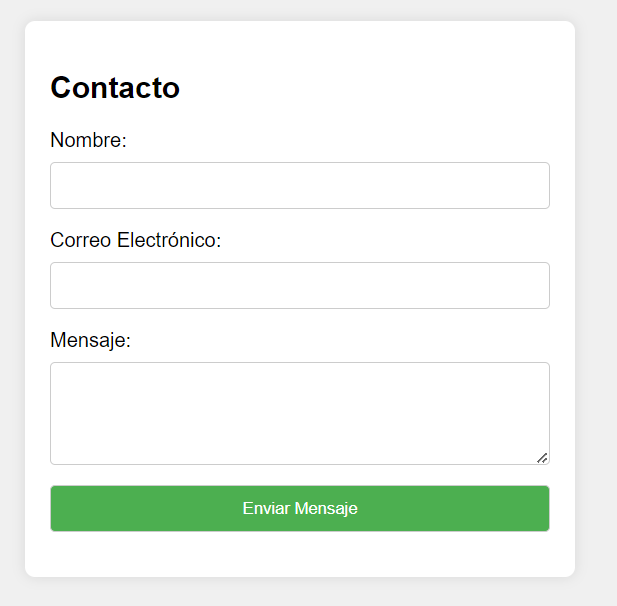
**Tabla 2**

Código del archivo estilos.css

|  |
| --- |
| body {      font-family: 'Arial', sans-serif;      margin: 0;      padding: 0;      display: flex;      justify-content: center;      align-items: center;      height: 100*vh*;      background-color: #f0f0f0;  }  .container {      width: 400*px*;      background-color: #fff;      padding: 20*px*;      border-radius: 8*px*;      box-shadow: 0 0 10*px* rgba(0, 0, 0, 0.1);  }  .contact-form {      display: flex;      flex-direction: column;  }  label {      margin-bottom: 8*px*;  }  input,  textarea {      padding: 10*px*;      margin-bottom: 16*px*;      border: 1*px* solid #ccc;      border-radius: 4*px*;      box-sizing: border-box;  }  input[type="submit"] {      background-color: #4caf50;      color: #fff;      cursor: pointer;  }  input[type="submit"]:hover {      background-color: #45a049;  } |

**Figura 9**

Vista del archivo index.html desde el navegador web.



## *Helpers*

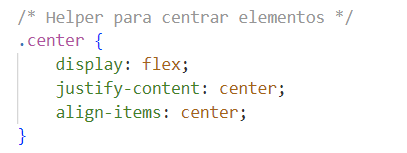
Los "helpers" en CSS son clases o estilos reutilizables que simplifican el desarrollo y mantenimiento del código. Estos estilos se diseñan para proporcionar funcionalidades específicas que pueden aplicarse a múltiples elementos. En esta sección, exploraremos los elementos, propiedades y ejemplos de "helpers" en CSS.

***Elementos Básicos de un Helper***

**Clases Reutilizables**:Los helpers generalmente se implementan mediante clases que se pueden aplicar a varios elementos en el código HTML.

**Figura 10**

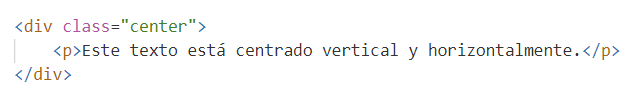
Helper para centrar elementos.



**Elementos HTML:**Los "helpers" pueden aplicarse a una variedad de elementos HTML, desde contenedores hasta elementos de texto.

**Figura 11**

Helper para centrar elementos html.

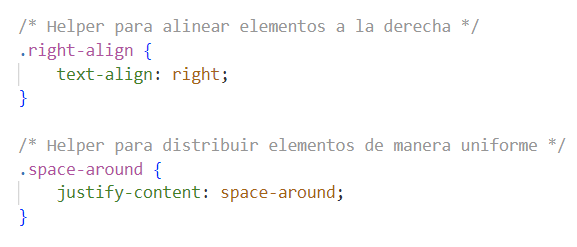


***Propiedades y Estilos en CSS***

**Estilos de Posicionamiento:**Los "helpers" a menudo incluyen estilos de posicionamiento para alinear y distribuir elementos de manera específica.

**Figura 12**

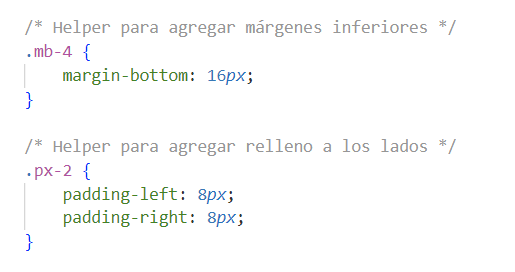
Helper que ayudan mejoran estilos de Posicionamiento



**Estilos de Margen y Padding:**Los "helpers" pueden definir estilos de márgenes y relleno para facilitar el diseño y la separación de elementos.

**Figura 13**

Estilos de Margen y Padding

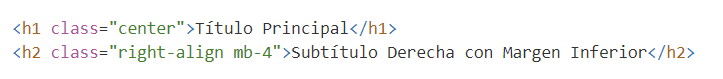


***Utilización de Helpers en Situaciones Prácticas***

**Creación de Encabezados Estilizados:**Aplicar "helpers" para estilizar encabezados de manera coherente en todo el sitio.

**Figura 14**

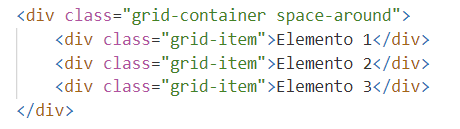
Creación de Encabezados Estilizados



**Diseño de Cuadrículas Flexibles:** Utilizar "helpers" para diseñar cuadrículas flexibles y responsivas.

**Figura 15**

Diseño de cuadrículas flexibles



## Componentes

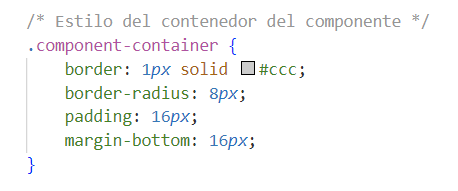
Los componentes en CSS son bloques autónomos que encapsulan un conjunto específico de funcionalidades y estilos. Estos ayudan a modularizar el código y mejorar la reutilización. En esta sección, exploraremos los elementos, propiedades y ejemplos de componentes en CSS.

***Elementos Básicos de un Componente***

**Contenedores de Componentes.** Los componentes suelen estar contenidos en divs u otros elementos HTML para agrupar y estilizar de manera coherente.

**Figura 16**

Aplicar estilos a los contenedores de Componentes



**Estructura del Componente.** La estructura interna del componente puede variar según su propósito, pero generalmente incluirá elementos como encabezados, imágenes y texto.

**Figura 17**

Estructura de un componente

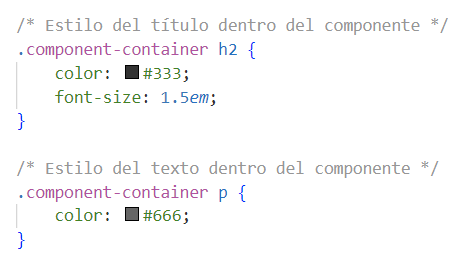


***Propiedades y Estilos en CSS***

**Estilos de Texto y Tipografía:** Aplicar estilos de texto específicos para asegurar coherencia en los componentes.

**Figura 18**

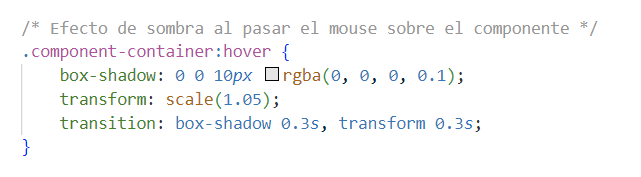
Estilos de Texto y Tipografía en CSS



**Efectos de hover y transiciones:**Agregar interactividad a los componentes mediante estilos de hover y transiciones.

**Figura 19**

Efectos de hover y transiciones



***Utilización de Componentes en Situaciones Prácticas***

**Creación de tarjetas de producto:**Utilizar componentes para diseñar tarjetas de productos que se pueden reutilizar en una tienda en línea.

**Figura 20**

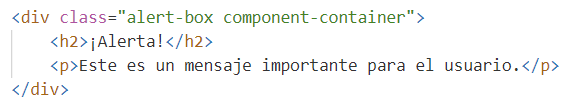
Creación de tarjetas de producto mediante CSS



**Diseño de cajas de alerta:** Crear componentes para cajas de alerta que se pueden incorporar fácilmente en diversas páginas.

**Figura 21**

Diseño de cajas de alerta



Ejemplo completo:

**Tabla 3**

Ejemplo completo de CSS y HTML

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <link rel="stylesheet" href="estilos.css">      <title>Tienda en Línea</title>  </head>  <body>      <div class="product-card component-container">          <h2>Zapatillas Deportivas</h2>          <img src="zapatillas.jpg" alt="Imagen de las Zapatillas">          <p class="price">$99.99</p>          <button class="add-to-cart">Agregar al Carrito</button>      </div>  </body>  </html> |

**Tabla 4**

Contenido del fichero estilos.css

|  |
| --- |
| body {      font-family: 'Arial', sans-serif;      margin: 0;      padding: 0;      display: flex;      justify-content: center;      align-items: center;      height: 100*vh*;      background-color: #f0f0f0;  }  .component-container {      border: 1*px* solid #ccc;      border-radius: 8*px*;      padding: 16*px*;      margin-bottom: 16*px*;      text-align: center;      transition: box-shadow 0.3*s*, transform 0.3*s*;  }  .product-card img {      max-width: 100*%*;      height: auto;      margin-bottom: 12*px*;  }  .price {      color: #333;      font-size: 1.2*em*;      font-weight: bold;  }  .add-to-cart {      background-color: #4caf50;      color: #fff;      padding: 8*px* 16*px*;      border: none;      border-radius: 4*px*;      cursor: pointer;      transition: background-color 0.3*s*;  }  .add-to-cart:hover {      background-color: #45a049;  }  .component-container:hover {      box-shadow: 0 0 10*px* rgba(0, 0, 0, 0.1);      transform: scale(1.05);  } |

**Figura 22**

Ejecución del código html y CSS desde el navegador.



## Templates

Los templates en CSS ofrecen una estructura predefinida para facilitar la creación de páginas web consistentes y bien diseñadas. En esta sección, exploraremos los elementos, propiedades y ejemplos de templates en CSS.

***Estructura HTML Básica***

Los templates suelen comenzar con una estructura HTML básica que incluye elementos esenciales como head, body, y enlaces a hojas de estilo (CSS) y scripts (JavaScript).

**Figura 23**

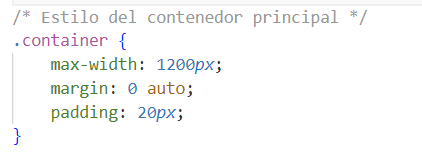
Estructura básica de una página html.



**Contenedor Principal:**El contenedor principal del template agrupa el contenido de la página y puede incluir encabezados, menús y secciones.

**Figura 24**

Contenedor principal en CSS

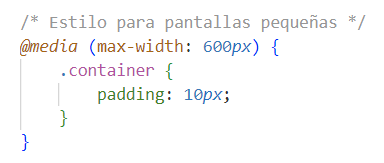


***Propiedades y Estilos en CSS***

**Diseño Responsivo:**Aplicar estilos que hagan que el template sea receptivo a diferentes tamaños de pantalla utilizando reglas de medios (@media).

**Figura 25**

Propiedades de CSS para el diseño responsive.



***Tipografía y Colores:*** Definir estilos de tipografía y colores que mantengan la coherencia en toda la página.

**Figura 26**

Tipografía y colores para el desarrollo de templates.

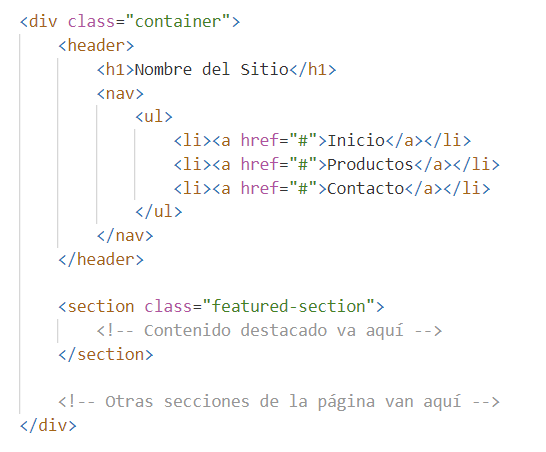


***Ejemplo detallado sobre la utilización de templates en situaciones prácticas***

**Página de Inicio:** Crear un template para la página de inicio que incluya un encabezado, una sección de destacados y enlaces a otras secciones.

**Figura 27**

Sección container dentro del documento HTML



**Página de Producto:**Utilizar el mismo template para crear una página de producto, adaptando el contenido según sea necesario.

**Figura 28**

Cabecera y contenido del container de productos.



**Ejemplo completo**

**Tabla 5**

Código fuente de la página index.html

|  |
| --- |
| <!DOCTYPE html>  <html lang="es">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <link rel="stylesheet" href="styles.css">      <title>Sitio Web Ejemplo</title>  </head>  <body>      <div class="container">          <header>              <h1>Nombre del Sitio</h1>              <nav>                  <ul>                      <li><a href="#">Inicio</a></li>                      <li><a href="#">Productos</a></li>                      <li><a href="#">Contacto</a></li>                  </ul>              </nav>          </header>          <section class="featured-section">              <h2>Destacados</h2>              <div class="featured-item">                  <img src="destacado1.jpg" alt="Producto Destacado 1">                  <p>Descripción corta del producto destacado.</p>              </div>              <div class="featured-item">                  <img src="destacado2.jpg" alt="Producto Destacado 2">                  <p>Descripción corta del producto destacado.</p>              </div>          </section>          <section class="main-content">              <h2>Contenido Principal</h2>              <p>Texto explicativo sobre el contenido principal del sitio.</p>          </section>          <footer>              <p>&copy; 2023 Sitio Web Ejemplo. Todos los derechos reservados.</p>          </footer>      </div>  </body>  </html> |

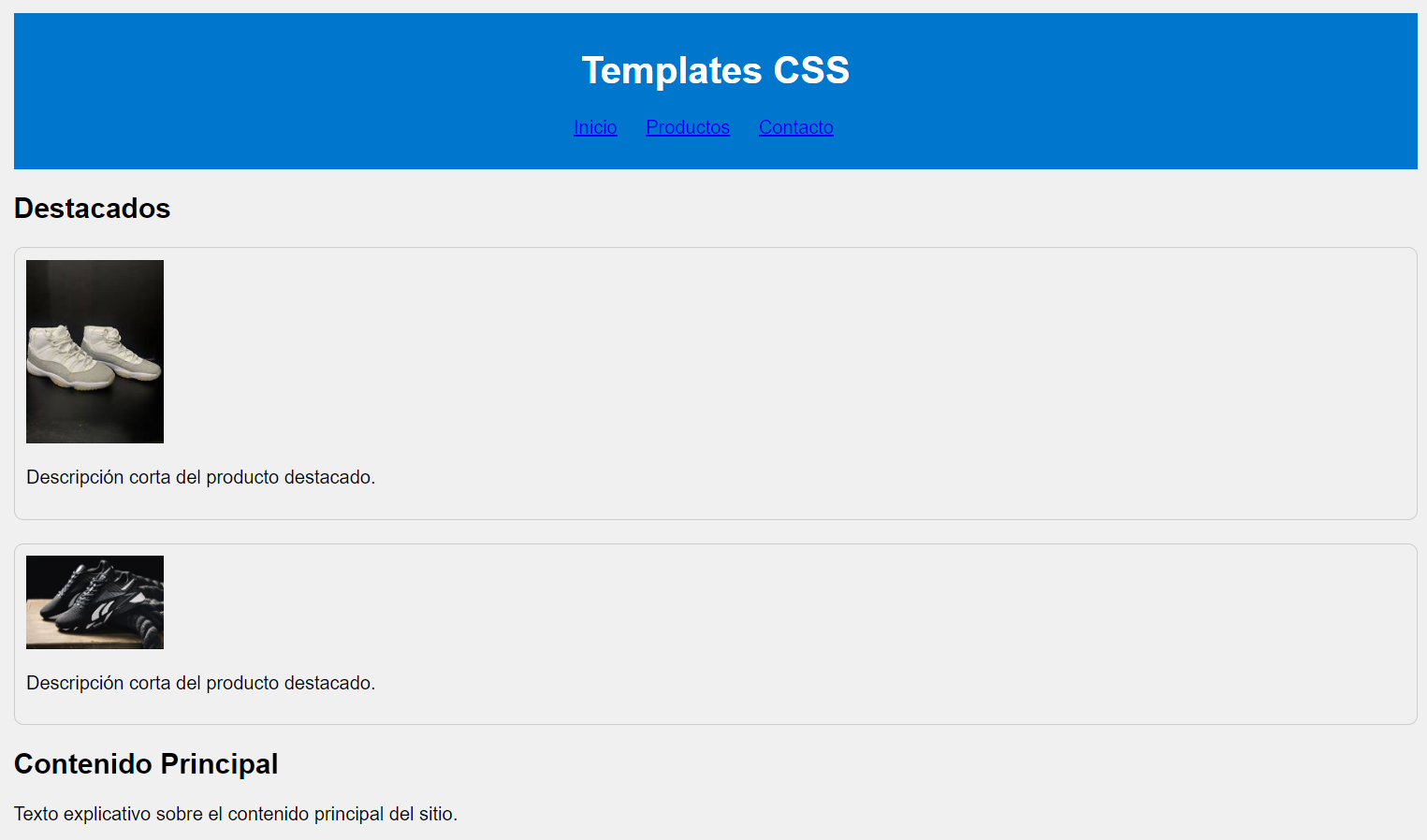
**Tabla 6**

Código fuente del archivo styles.css

|  |
| --- |
| body {      font-family: 'Arial', sans-serif;      margin: 0;      padding: 0;      display: flex;      justify-content: center;      align-items: center;      min-height: 100*vh*;      background-color: #f0f0f0;  }  .container {      max-width: 1200*px*;      margin: 0 auto;      padding: 20*px*;  }  header {      text-align: center;      background-color: #0077cc;      color: #fff;      padding: 10*px*;  }  nav ul {      list-style: none;      padding: 0;  }  nav ul li {      display: inline;      margin-right: 20*px*;  }  .featured-section {      margin-top: 20*px*;  }  .featured-item {      border: 1*px* solid #ccc;      border-radius: 8*px*;      padding: 10*px*;      margin-bottom: 20*px*;  }  .main-content {      margin-top: 20*px*;  }  footer {      margin-top: 20*px*;      text-align: center;      color: #555;  } |

**Figura 29**

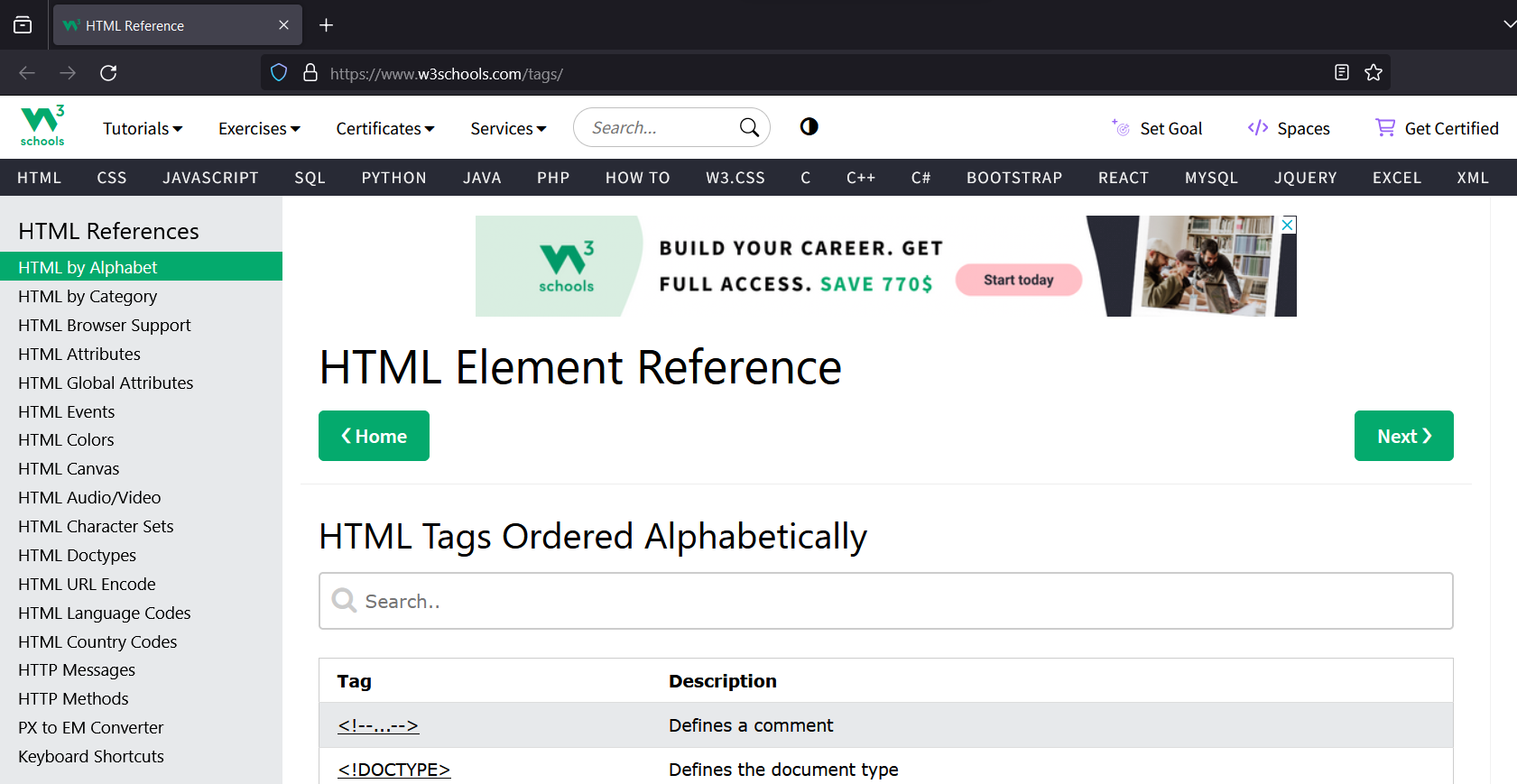
Resultado del template desde el navegador.



# Recursos complementarios

* Manual de referencia de los elementos HTML

<https://www.w3schools.com/tags/>



# Referencias

Brown, A. (2020). *Desarrollo web moderno: estrategias y herramientas*. Editorial Digital Solutions.

García, M.; Rodríguez, A. & Pérez, S. (2021). Tecnologías Back-End: un enfoque integral. *Revista de Desarrollo Tecnológico*, 15(2), 45-63. doi:10.1234/rdt.2021.1234567890

Johnson, R. (2018). *Desarrollo de interfaces de usuario: principios y prácticas*. España: Editorial TechPress.

Johnson, R. (2019). Desarrollo de software empresarial: principios y prácticas. Editorial TechPress.

Smith, J. (2018). *Diseño de interfaces de usuario: principios y prácticas*. Editorial TechBooks.

# Autoevaluación

1. **¿Cuál de las siguientes etiquetas se utiliza para crear un campo de entrada de texto en un formulario HTML?**

<input type="text">

<text-input>

<textfield>

<boxinput>

1. **¿Cuál de las siguientes propiedades de CSS se utiliza para especificar la cantidad de columnas en un grid?**

grid-gap

grid-template-columns

grid-row

grid-column

1. **¿Cómo se podría aplicar un estilo de fuente personalizada en CSS?**

font-family: custom-font;

font-style: custom;

font-source: url('custom-font.woff');

@font-face { font-family: custom-font; src: url('custom-font.woff'); }

1. **¿Qué se entiende por "componente" en el contexto de desarrollo web?**

Un lenguaje de programación.

Un conjunto de estilos reutilizables y funcionalidades encapsuladas.

Una etiqueta HTML específica.

Un tipo de base de datos.

1. **¿Cuál es la función de las reglas de medios (@media) en CSS para la responsividad?**

Definir imágenes de alta resolución.

Cambiar el estilo de la página según el dispositivo o tamaño de pantalla.

Crear animaciones interactivas.

Especificar fuentes personalizadas.

1. **¿Qué caracteriza a un "helper" en CSS?**

Es una clase de programación en JavaScript.

Proporciona funciones adicionales para el servidor.

Es un estilo reutilizable que simplifica el desarrollo.

Solo se utiliza en el diseño de bases de datos.

1. **¿Cuál es el propósito principal de utilizar templates en desarrollo web?**

Definir estilos para imprimir documentos.

Proporcionar una estructura predefinida para crear páginas web consistentes.

Crear animaciones interactivas.

Generar archivos de hojas de estilo.

1. **¿Cómo se especifica que un campo de entrada en un formulario es obligatorio?**

<input required>

<input type="required">

<input mandatory>

<input optional>

1. **¿Por qué es beneficioso utilizar componentes en el desarrollo web?**

Incrementa la complejidad del código.

Facilita la reutilización de estilos y funcionalidades.

Limita la modularidad del código.

Aumenta la carga del servidor.

1. **¿Qué ventajas ofrece el uso de templates en el desarrollo web?**

Mayor complejidad en la estructura del código.

Menor coherencia visual entre páginas.

Facilita la creación de páginas consistentes y bien diseñadas.

Incrementa la carga de trabajo del desarrollador.