

DEPARTAMENTO:	Ciencias de la Computación - DCCO-SS	CARRERA:	Ingeniería en tecnologías de la Información		
ASIGNATURA:	Programación Integrativa	NIVEL:	Sexto	FECHA:	23/05/2025
DOCENTE:	Ing. Paulo Galarza	PRÁCTICA N°:	1	CALIFICACIÓN:	

Desarrollar una Tarjeta de Producto para un E-commerce con Composición de Componentes

ANTHONY GEOVANNY MEJIA GAIBOR

RESUMEN

Este laboratorio tuvo como propósito desarrollar un componente web personalizado (<product-card>) para una tarjeta de producto destinada a un e-commerce. Se emplearon tecnologías modernas como Custom Elements, Shadow DOM y slots para encapsular y personalizar el componente, integrando elementos como imagen, título y precio. Además, se implementaron datos dinámicos mediante atributos HTML y estilos personalizables con variables CSS, asegurando flexibilidad y adaptabilidad. El proceso incluyó la configuración del proyecto, la creación del componente, la integración de datos dinámicos, y la personalización visual, todo gestionado de manera ordenada en un repositorio GitHub con la rama tarea2-integracion-html. Se concluye que los Web Components ofrecen ventajas significativas en términos de encapsulación, reutilización y mantenibilidad, siendo ideales para aplicaciones modulares como las de comercio electrónico. El desarrollo se finalizó el 23 de mayo de 2025 a las 02:53 PM -05, consolidando habilidades técnicas y prácticas esenciales para proyectos futuros.

Palabras Clave: Web Components, Shadow DOM, slots.

1. INTRODUCCIÓN:

El desarrollo de aplicaciones web modernas requiere el uso de tecnologías que permitan crear interfaces modulares, reutilizables y encapsuladas, características esenciales para proyectos de comercio electrónico donde la presentación de productos debe ser eficiente y adaptable. En este contexto, el laboratorio de Programación Integrativa de Componentes, correspondiente al periodo lectivo 202550, tuvo como propósito principal diseñar y construir una tarjeta de producto para un e-commerce utilizando Web Components. Este enfoque permitió aplicar conceptos avanzados como Custom Elements, Shadow DOM, HTML Templates, y CSS Variables, asegurando un desarrollo estructurado y profesional.

Las actividades se llevaron a cabo con un enfoque disciplinado y metódico en el laboratorio. Se comenzó configurando una estructura base del proyecto que incluyó archivos como `index.html`, `product-card.js` y `styles.css`, lo que permitió mantener un orden claro desde el inicio. Cada paso se ejecutó secuencialmente, siguiendo las instrucciones del laboratorio, y se utilizó un servidor local (Live Server en Visual Studio Code) para probar los cambios en tiempo real, asegurando un flujo de trabajo eficiente. Además, se emplearon herramientas de depuración, como la consola del navegador, para identificar y resolver problemas, como el manejo dinámico de imágenes y la actualización de atributos. Este proceso no solo cumplió con los objetivos técnicos, sino que también fomentó habilidades de organización y resolución de problemas, culminando el desarrollo el 23 de mayo de 2025 a las 02:20 PM -05.

2. OBJETIVO(S):

2.1 Diseñar y desarrollar un componente web personalizado (`<product-card>`) que represente una tarjeta de producto, integrando elementos como imagen, título y precio, para su uso en aplicaciones de comercio electrónico.

- 2.2 Implementar técnicas de encapsulación mediante Shadow DOM y composición con slots, permitiendo que el componente sea personalizable y no interfiera con los estilos del resto de la página.
- 2.3 Facilitar la personalización dinámica del componente a través de atributos HTML y variables CSS, asegurando que los datos y estilos puedan modificarse de manera eficiente y en tiempo real.

3. DESARROLLO

El desarrollo de la tarjeta de producto para un e-commerce se llevó a cabo mediante la implementación de un componente web personalizado utilizando tecnologías modernas como Custom Elements, Shadow DOM, HTML Templates, y CSS Variables. Este proceso se estructuró en varias etapas, desde la configuración inicial del proyecto hasta la integración de datos dinámicos y estilos personalizables, culminando en un componente funcional y reutilizable. A continuación, se detalla el uso de Shadow DOM y slots, así como las ventajas de los Web Components en aplicaciones modulares.

Explicación del Uso de Shadow DOM y Slots

Uso de Shadow DOM

El Shadow DOM es una tecnología clave de los Web Components que permite encapsular la estructura y los estilos de un componente, asegurando que no se vean afectados por el resto de la página y viceversa. En este proyecto, se implementó el Shadow DOM para garantizar que los estilos definidos para la tarjeta de producto (`<product-card>`) no interfirieran con otros elementos de la página, promoviendo un diseño modular y mantenible.

La implementación del Shadow DOM se realizó en el archivo `product-card.js`, dentro del constructor de la clase `ProductCard`. El siguiente fragmento de código muestra cómo se adjuntó el Shadow DOM al componente:

Código

```
class ProductCard extends HTMLElement {  
  constructor() {  
    super();  
    // Adjuntar el Shadow DOM al componente  
    this.attachShadow({ mode: 'open' });  
    // ... resto del código ...  
  }  
}
```

En este fragmento, el método `this.attachShadow({ mode: 'open' })` crea un Shadow Root que encapsula el contenido del componente. El modo 'open' permite acceder al Shadow DOM desde JavaScript externo si es necesario, lo que facilitó la manipulación dinámica del contenido. Posteriormente, el contenido del template definido en `index.html` se clonó y se adjuntó al Shadow Root, asegurando que los estilos y la estructura del componente fueran independientes del DOM principal.

Uso de Slots

Los **slots** son otra característica fundamental de los Web Components que permiten la composición del contenido, es decir, la capacidad de insertar elementos personalizados desde el DOM principal (light DOM) hacia el Shadow DOM. En este proyecto, se utilizaron slots para proyectar el título y el precio del producto, permitiendo que el usuario del componente definiera estos valores directamente en el HTML.

El uso de slots se implementó dentro del `<template>` en `index.html`, específicamente en las secciones correspondientes al título y al precio de la tarjeta de producto. El siguiente fragmento de código muestra cómo se definieron los slots:

Código

```
<template id="product-card-template">
  <style>
    /* Estilos omitidos para brevedad */
  </style>
  <div class="card">
    <div class="category">Sports Store <span class="heart-icon">♥</span></div>
    <div class="image-container"></div>
    <h3><slot name="title">Título del producto</slot></h3>
    <p class="description">Zapatos Nike último modelo Unisex FG.</p>
    <div class="rating">
      <span class="star">★</span>
      <span class="star">★</span>
      <span class="star">★</span>
      <span class="star">★</span>
      <span class="star empty">☆</span>
    </div>
    <div>
      <span class="price"><slot name="price">Precio no disponible</slot></span>
      <div class="quantity">
        <button class="decrease">-</button>
        <span class="quantity-value">1</span>
        <button class="increase">+</button>
      </div>
    </div>
    <button class="add-to-cart">Agregar al Carrito</button>
  </div>
</template>
```

En este código, `<slot name="title">` y `<slot name="price">` actúan como placeholders que permiten proyectar contenido dinámico desde el light DOM. Por ejemplo, el título y el precio definidos en `<product-card title="Zapatos Nike" price="$15.00">` se insertan en estos slots, reemplazando los valores por defecto ("Título del producto" y "Precio no disponible"). Aunque el laboratorio originalmente también especificaba un slot para la imagen, se optó por manejar dinámicamente mediante JavaScript para mayor flexibilidad, lo que se ajustó en iteraciones posteriores.

Ventajas de los Web Components en Aplicaciones Modulares

Los Web Components ofrecen múltiples ventajas que los hacen ideales para aplicaciones modulares, especialmente en contextos como un e-commerce donde la reutilización y la mantenibilidad son esenciales. A continuación, se detallan algunas de las principales ventajas observadas durante el desarrollo de este proyecto:

- **Encapsulación:** Gracias al Shadow DOM, los estilos y la estructura del componente `<product-card>` están completamente aislados del resto de la página. Esto evita conflictos con otros estilos globales y asegura que los cambios en el componente no afecten a otros elementos, promoviendo un diseño más predecible y seguro.
- **Reutilización:** Los Web Components permiten definir elementos personalizados que pueden usarse en diferentes partes de una aplicación o incluso en proyectos distintos. Por ejemplo, `<product-card>` puede integrarse en múltiples páginas de un e-commerce para mostrar diferentes productos, ajustando únicamente los atributos como `title`, `price`, e `image`.
- **Personalización:** La combinación de slots y variables CSS permitió que el componente fuera altamente personalizable. Los slots facilitaron la inyección de contenido dinámico, mientras que las variables CSS como `--primary-color` y `--border-radius` permitieron ajustar el diseño visual desde el HTML, adaptándolo a diferentes contextos sin modificar el código del componente.
- **Mantenibilidad:** Al encapsular la lógica, los estilos y la estructura en un solo componente, el mantenimiento del código se simplifica. Cualquier actualización o mejora en `<product-card>` se aplica automáticamente a todas las instancias del componente, reduciendo el riesgo de errores y facilitando la escalabilidad del proyecto.

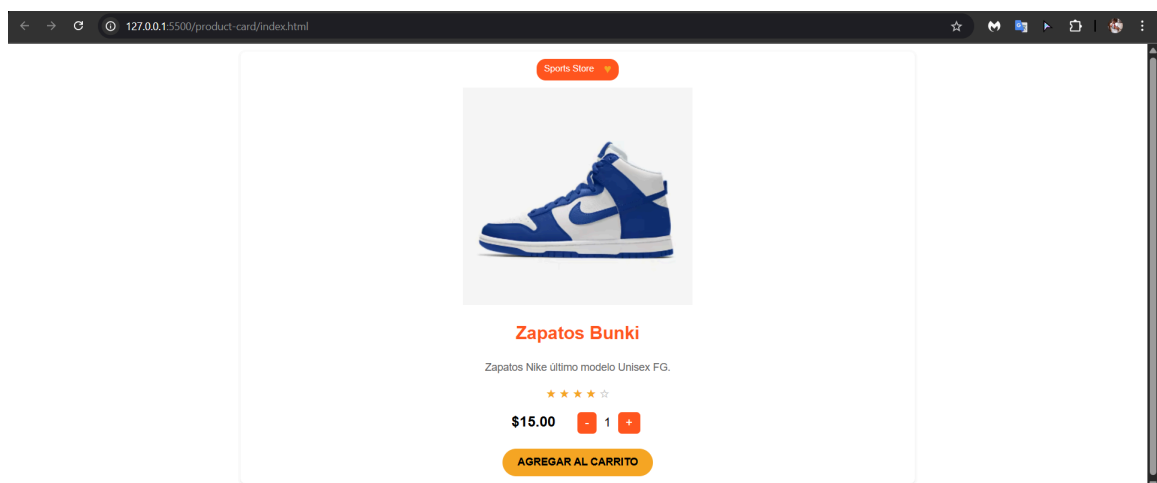
En resumen, el uso de Shadow DOM y slots en este laboratorio no solo cumplió con los objetivos técnicos, sino que también demostró cómo los Web Components pueden transformar el desarrollo web al ofrecer soluciones modulares, eficientes y altamente adaptables a las necesidades de aplicaciones modernas.

Código en ejecución

Muestra el componente `<product-card>` en un navegador, con la tarjeta de producto que presenta unos zapatos Nike, un título "Zapatos Nike", un precio de "\$15.00", una calificación de cuatro estrellas, un selector de cantidad, y un botón "Agregar al Carrito" en amarillo.

Figura 1

Tarjeta de Producto para E-commerce con Web Components

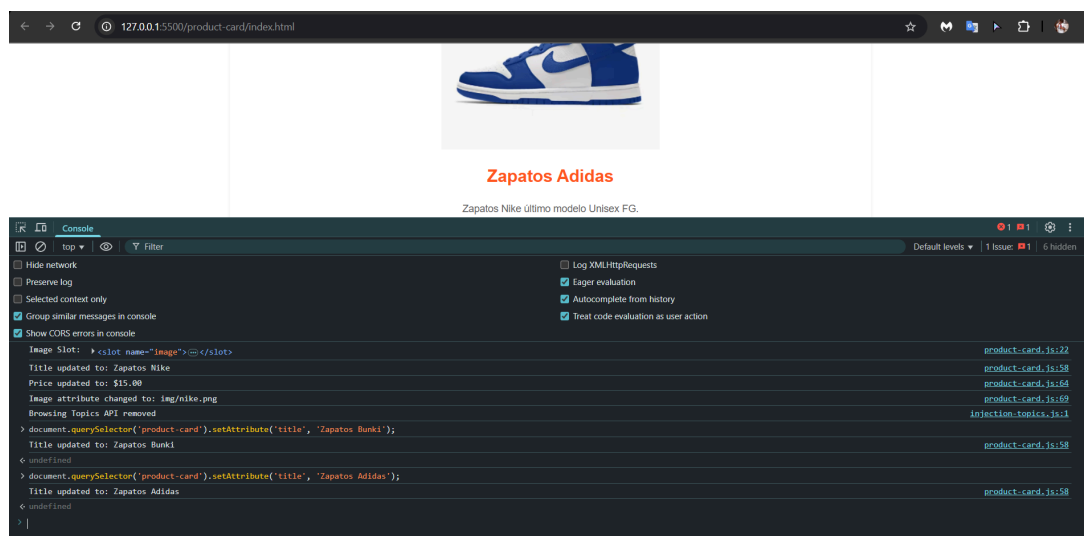


Nota: Elaboración propia (2025). La imagen presenta el título, precio y descripción del producto, tomados de los atributos y el template.

Se presenta en Visual Studio Code los archivos del proyecto (index.html, product-card.js, styles.css) abiertos, mostrando el componente <product-card> en el navegador y la terminal con commits en la rama tarea2-integracion-html.

Figura 1

Entorno de Desarrollo del Proyecto de Tarjeta de Producto



Nota: Elaboración propia (2025). La imagen contiene una captura de pantalla del entorno de desarrollo utilizado para el proyecto

4. CONCLUSIONES

El desarrollo de la tarjeta de producto para un e-commerce mediante Web Components permitió alcanzar satisfactoriamente los objetivos establecidos en el laboratorio, consolidando conocimientos fundamentales en el uso de tecnologías web modernas. En primer lugar, se diseñó y desarrolló el componente `<product-card>`, que integra elementos clave como la imagen, el título y el precio del producto, cumpliendo con su propósito de ser utilizado en aplicaciones de comercio electrónico. Este componente se implementó con una estructura clara y funcional, incluyendo características interactivas como un selector de cantidad y un botón para agregar al carrito, lo que lo hace adecuado para un entorno real de e-commerce.

Asimismo, se lograron implementar técnicas de encapsulación y composición mediante el uso de Shadow DOM y slots. El Shadow DOM aseguró que los estilos y la estructura del componente fueran independientes del resto de la página, evitando conflictos y promoviendo un diseño modular. Los slots, por su parte, permitieron personalizar el título y el precio del producto directamente desde el HTML, facilitando la integración del componente en diferentes contextos sin necesidad de modificar su código interno. Aunque inicialmente se manejó la imagen de forma dinámica, se ajustó el diseño para usar un slot, alineándose con los requerimientos del laboratorio y garantizando una mayor flexibilidad en la proyección de contenido.