

DEPARTAMENTO:	Ciencias de la Computación	CARRERA:	Ingeniería en Tecnologías de la Información		
ASIGNATURA:	PROGRAMACION INTEGRATIVA DE COMPONENTES	NIVEL:	Sexto	FECHA:	17/05/2025
DOCENTE:	Ing. Paulo Galarza	PRÁCTICA N°:	1	CALIFICACIÓN:	

FICHA TECNICA DEL PROYECTO

Toapanta Paez Darwin Andres

1. Título del Proyecto

Integrar Elementos HTML en un Web Component.

2. Descripción General

Este proyecto tiene como finalidad demostrar la implementación de Web Components en una aplicación web, utilizando tecnologías modernas como HTML5, CSS y JavaScript. Se desarrolló un componente personalizado que encapsula una tarjeta de usuario con imagen, nombre y descripción. Adicionalmente, se integró un formulario externo que permite modificar dinámicamente el contenido del componente.

3. Objetivos

- Crear un Componente con HTML Integrado.
- Integrar con el Componente de la Tarea 1.
- Subir Evidencias a GitHub.

4. Tecnologías Utilizadas

- HTML5
- CSS
- JavaScript
- Web Components (Custom Elements, Shadow DOM y Slots).

5. Requisitos del Sistema

Hardware y software necesarios para ejecutar el proyecto.

Sistemas operativos compatibles.

- Windows
- macOS
- Linux

Navegador recomendado.

- Google Chrome
- Brave

6. Funcionalidades del Proyecto

- Cómo se integran elementos HTML estándar en un Web Component.

Los Web Components nos permiten encapsular y reutilizar los elementos personalizados en otras partes del código. Para integrar elementos HTML estándar como `<form>`, `<input>`, `<button>`, entre otros, se utiliza el Shadow DOM, que actúa como un DOM encapsulado del componente.

```
render() {  
  this.shadowRoot.innerHTML = `  
    <form id="formulario">  
      <label for="nuevoContenido">Ingresa un nuevo nombre:</label>  
      <input type="text" id="nuevoContenido" name="nuevoContenido" />  
      <slot name="boton-enviar"></slot>  
    </form>  
  `;  
}
```

- **Mecanismos de comunicación entre componentes (ej: eventos personalizados).**

Como podemos ver en el ejercicio, El componente `<formulario-cambio>` accede a `<tarjeta-usuario>` mediante `document.querySelector()` y modifica un slot directamente:

```
// aquí cambiamos slot contenido donde esta mi nombre del componente <tarjeta-usuario>  
const tarjeta = document.querySelector('tarjeta-usuario');  
if (tarjeta) {  
  const slotContenido = tarjeta.querySelector('[slot="contenido"]');  
  if (slotContenido) {  
    slotContenido.textContent = nuevoTexto;  
  }  
}
```

- **Ventajas de encapsular HTML complejo dentro de un componente.**

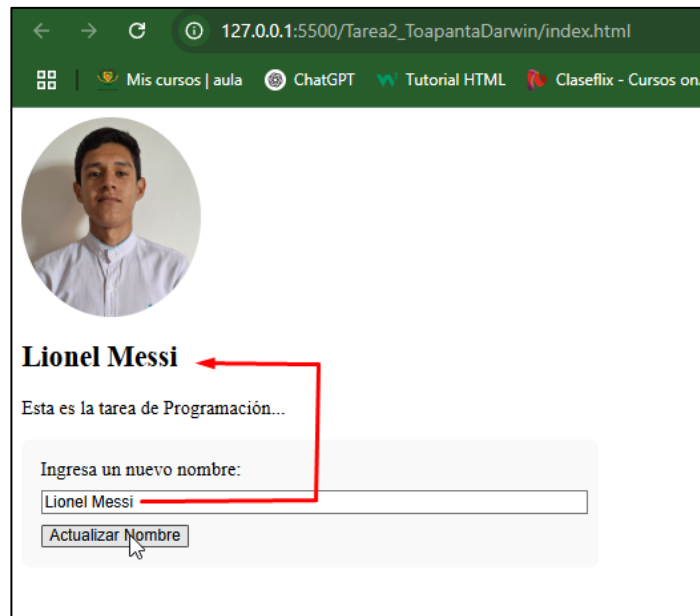
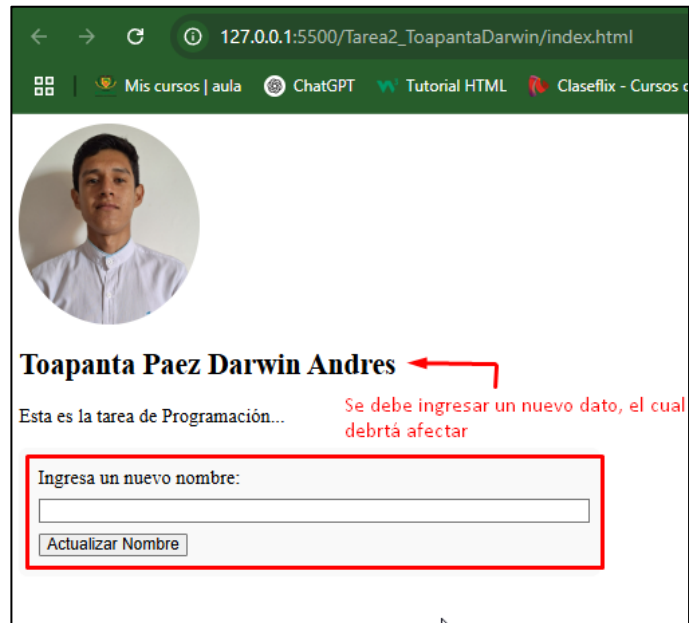
- Una de las principales ventajas es que se puede reutilizar la misma estructura (como una tarjeta de usuario o formulario) en múltiples partes de una aplicación sin duplicar código.
- Además, que los estilos dentro del Shadow DOM no se ven afectados por estilos externos ni afectan al resto del sitio, evitando conflictos de CSS.

7. Estructura del Proyecto

El proyecto sigue la siguiente estructura:

- Index.html (Página principal que utiliza el componente).
- Script.js (Aquí defino el componente usado).
- Perfil.jpg. (Una imagen para usar dentro del ejemplo, la cual es opcional).

8. Capturas de Pantalla



9. Datos del Autor o Equipo

- **Nombre:** Darwin Andrés Toapanta Paez
- **Carrera:** ITIN
- **Materia:** Programación Integrativa
- **Docente:** Ing. Paulo Galarza
- **Fecha:** 17/05/2025

10. Estado del Proyecto

Finalizado.