

DEPARTAMENTO:	CIENCIAS DE LA COMPUTACIÓN	CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN		
ASIGNATURA:	Programación Integrativa de Componentes	NIVEL:	6	FECHA:	17/05/25
DOCENTE:	Ing. Paulo Galarza	TAREA N°:	3	CALIFICACIÓN:	

## Implementar Especificaciones Avanzadas en Web Components

Anderson Arquimedes Campos Alvarado

### RESUMEN

Este proyecto desarrolla un componente web avanzado llamado <espe-modal>, implementado mediante ES Modules para lograr una modularización efectiva. Utiliza Shadow DOM para encapsular estilos y estructura, asegurando que el componente sea aislado y personalizable. Se emplean <template> y slots con contenido de respaldo, lo que permite insertar contenido dinámico desde el HTML principal. Además, el componente integra eventos personalizados para comunicarse con el entorno externo, permitiendo acciones como cerrar el modal y emitir mensajes específicos. El diseño incluye media queries para asegurar la adaptabilidad en dispositivos móviles y diferentes tamaños de pantalla. Este enfoque facilita la reutilización, mantenimiento y escalabilidad del componente dentro de aplicaciones modernas. La implementación demuestra cómo construir componentes robustos, flexibles y responsivos que mejoran la interacción del usuario.

**Palabras clave:** Web Components, Eventos personalizados, Shadow DOM

### 1. INTRODUCCIÓN:

Se creó un componente web modular llamado <espe-modal>, usando Shadow DOM y eventos personalizados para facilitar la comunicación y reutilización en aplicaciones modernas.

### 2. OBJETIVO(S):

#### 2.1 Objetivo general

- Diseñar un componente modular con Shadow DOM y eventos personalizados para mejorar la interacción web.

#### 2.2. Objetivos específicos

- Modularizar el componente con ES Modules.
- Usar Shadow DOM y slots para contenido dinámico.
- Emitir eventos personalizados para comunicación externa.

### 3. MARCO TEÓRICO/ DESARROLLO



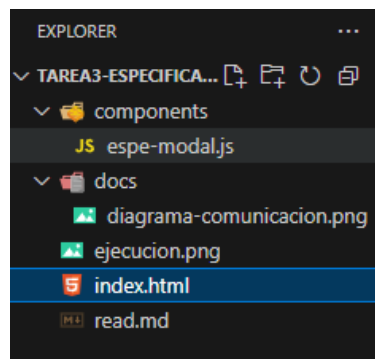
Repositorio git:

[https://github.com/arquimides12/U1\\_Tarea1\\_AndersonCampos/tree/tarea3-especificaciones-avanzadas](https://github.com/arquimides12/U1_Tarea1_AndersonCampos/tree/tarea3-especificaciones-avanzadas)

#### Estructura del proyecto

La estructura del proyecto se organiza en carpetas que separan el componente, las evidencias y el archivo principal

HTML:



#### Paso 1: Modularización con ES Modules

Se utilizó la importación del componente mediante `type="module"`:

```
<!-- Importación del componente -->
<script type="module" src="components/espe-modal.js"></script>
```

#### Paso 2: Uso de `<template>` y Shadow DOM

El componente encapsula su estructura y estilos mediante Shadow DOM:

```
constructor() {  
  super();  
  this.attachShadow({ mode: "open" });  
  this.shadowRoot.appendChild(this.template.content.cloneNode(true));  
}
```

### Paso 3: Slots con contenido de respaldo

Se definieron slots con valores por defecto que permiten personalización desde HTML:

```
<div class="modal">  
  <h2><slot name="titulo">Resultado del estudiante</slot></h2>  
  <p><slot name="contenido">Seleccione una calificación:</slot></p>  
  <button class="btn-verde" id="btn20">20 puntos</button>  
  <button class="btn-azul" id="btn18">18 puntos</button>  
  <button class="btn-rojo" id="btn14">14 puntos</button>  
</div>
```

### Paso 4: Emisión de eventos personalizados

Se despacha el evento modal-cerrado al hacer clic en uno de los botones, con un mensaje específico:

```
enviarNota(mensaje) {  
  this.dispatchEvent(new CustomEvent("modal-cerrado", {  
    bubbles: true,  
    composed: true,  
    detail: { mensaje }  
  }));  
  // this.remove(); // Se mantiene visible  
}
```

### Paso 5: Estilos internos y diseño responsive

El componente incluye estilos encapsulados y un @media para hacerlo adaptable a pantallas pequeñas:

```
@media (max-width: 600px) {  
  .modal {  
    width: 90%;  
    top: 20%;  
    transform: translate(-50%, -20%);  
    padding: 1rem;  
  }  
  
  button {  
    width: 100%;  
  }  
}
```

**Paso 6: Captura del evento en el documento**

El evento personalizado es capturado desde el index.html para mostrar el mensaje al usuario:

```
<!-- Listener para capturar el evento personalizado -->
<script>
  document.addEventListener("modal-cerrado", (e) => {
    console.log("Evento capturado:", e.detail.mensaje);
    alert(e.detail.mensaje); // opcional: mostrar en pantalla también
  });
</script>
```

**4. CONCLUSIONES**

- Al desarrollar el deber la modularización con ES Modules nos ayudó organizar y reutilizar el componente <espe-modal>, contribuyendo a un desarrollo más limpio y mantenible.
- El uso de Shadow DOM y slots permite encapsular estilos y contenido, haciendo que el componente sea altamente personalizable y aislado del resto de la página.
- Para finalizar la implementación de eventos personalizados logró una comunicación eficiente y flexible entre el componente y su entorno, permitiendo manejar acciones como el cierre del modal de manera efectiva.

**5. RECOMENDACIONES**

- Usa modularización en futuros proyectos.
- Explora más técnicas de Shadow DOM y slots.
- Implementa eventos personalizados para comunicación entre componentes.

**6. BIBLIOGRAFÍA:**

Mozilla Developer Network. (2024). Using Shadow DOM. MDN Web Docs. Recuperado de [https://developer.mozilla.org/en-US/docs/Web/Web\\_Components/Using\\_shadow\\_DOM](https://developer.mozilla.org/en-US/docs/Web/Web_Components/Using_shadow_DOM)

Mozilla Developer Network. (2024). JavaScript Modules. MDN Web Docs. Recuperado de <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules>