



**UNIVERSIDAD DE LAS FUERZAS ARMADAS-ESPE
SEDE SANTO DOMINGO DE LOS TSÁCHILAS**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN - DCCO-SS
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**

PERIODO : Octubre 2024 – Marzo 2024

ASIGNATURA : Programación Integrativa de Componente

TEMA : Tarea 1 U1

ESTUDIANTE : González Orellana Adriana Pamela

NIVEL-PARALELO - NRC: 6to “A” -1428

DOCENTE : Ing. Luis Alberto Castillo Salinas

FECHA DE ENTREGA : 05/noviembre/2024

Tarea 1: Web Components

SANTO DOMINGO – ECUADOR

2024

1. Introducción	3
2. Desarrollo	4
2.1. Componentes web	4
2.2. Sintaxis	5
2.3. Semántica	6
2.4. Funcionalidad	7
3. Conclusiones	12
4. Recomendaciones	13
5. Bibliografía	14

1. INTRODUCCIÓN

En la creación de aplicaciones web contemporáneas, la exigencia de desarrollar interfaces modulares, reutilizables y de fácil mantenimiento ha promovido la adopción de tecnologías emergentes. Los Componentes Web constituyen una solución sólida para el desarrollo de componentes web independientes que pueden emplearse en cualquier aplicación, sin importar el marco de trabajo empleado.

Los Componentes Web se fundamentan en tres tecnologías fundamentales, Elementos Personalizados, shadow DOM y Plantillas HTML. Estas tecnologías facilitan a los programadores la creación de interfaces web con una estructura más ordenada y escalable, mientras fomentan la reutilización de código y la modularidad.

Los Elementos Personalizados permiten crear nuevos elementos HTML personalizados con comportamientos únicos.

Esto facilita la creación de componentes reutilizables sin depender de frameworks externos. En el ámbito contrario, el shadow DOM ofrece una capa de encapsulamiento. Esto permite que cada componente cuente con su propio órbito DOM, apartado de la estructura principal de la página. Como resultado, se resuelven los problemas de interfaces en estilos y comportamientos. De forma definitiva, los plantillas HTML proporcionan una manera eficaz de establecer estructuras HTML que no se renderizan de inmediato.

2. DESARROLLO

En el Desarrollo web, los desarrolladores también intentan hacer lo mismo. Sin embargo, el entorno web incluye muchos otros factores. Por ejemplo, incluye diferentes tipos de código fuente como HTML, CSS y JavaScript. Además, cuando estos códigos se juntan en una página web, pueden afectarse entre sí. Por lo tanto, encapsular una implementación en el desarrollo web es más complicado. Requiere que siempre se use el mismo conjunto de código fuente.

Web Components

Los componentes web son un conjunto de diferentes tecnologías que le permiten crear elementos personalizados reutilizables, con su funcionalidad encapsulada lejos del resto de su código y utilizarlos en sus aplicaciones web.

Los componentes web tienen como objetivo resolver estos problemas, estos consisten en tres tecnologías principales que pueden usarse juntas para crear elementos personalizados versátiles con funcionalidad encapsulada que se puede reutilizar donde quiera sin temor a adiciones de código.

Shadow DOM

Un conjunto de APIs de JavaScript para fijar un árbol DOM "sombra" encapsulado a un elemento que es renderizado por separado del documento DOM principal y controlando funcionalidad privada. De esta forma, se puede mantener características de un elemento en privado, así pueden mantener características tener el estilo y los script sin miedo de colisiones con otras partes del documento. El DOM es una interfaz de programación de aplicaciones (API) que permite a JavaScript interactuar con un documento web, como acceder a sus elementos y modificar las propiedades de los elementos lo vemos en la figura 1.

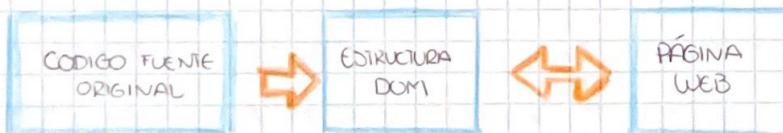


Fig 1. La estructura del DOM entre el código fuente original.

Como se describe en la figura 1 los elementos en una página web aparecen como el código fuente original cuando fueron desarrollados por un programador. Después de ser enviados al navegador a través de una conexión de red, el navegador traduce el código fuente en una estructura DOM y luego renderiza la página web en el dispositivo del usuario. JavaScript que se ejecuta en el navegador web puede modificar la página web cambiando los elementos en la estructura DOM.

- Shadow DOM no solo resuelve problemas de adición de estilos, sino también cómo facilita la creación de componentes con un alcance de estilos más controlado y predecible lo cual nos permite aplicar estilos de forma encapsulada

Web Components

- Sintaxis

Los web components implican principalmente la definición y el uso de custom elements, el uso de shadow DOM para la encapsulación y HTML Templates para definir la estructura de los componentes. Ahora vamos a detallar su sintaxis básica:

CUSTOM ELEMENTS (Elementos Personalizados)	SHADOW DOM
Es un conjunto de APIs de Javascript que permite definir elementos personalizados y su comportamiento, y puede ser utilizado como deseé en la interfaz del usuario.	Un conjunto de Javascript para fijar un círculo DOM "sombra" encapsulado en un elemento que es renderizado por separado del documento DOM principal y controla la funcionalidad asociada.
HTML TEMPLATES (Plantillas HTML)	EJ. MODULES (Módulos ECMAScript)
Los elementos <template> y <slot> nos permite escribir plantillas de marcado que no son despegables en la página renderizada. Se los puede reutilizar en múltiples ocasiones a modo base de la estructura de un elemento personalizado.	Son un estándar de Javascript para importar y exportar código entre archivos. Se utiliza para importar y exportar funcionalidades de los web components.
Exportación	Importación
<pre>export function miFuncion() { "Código función" }</pre>	<pre>import {miFuncion} from './mimodulo.js';</pre>

- Semántica

Nos permite crear elementos personalizados con una semántica clara y específica para cada aplicación. Esto quiere decir que los desarrolladores pueden crear una jerarquía de elementos que se ajuste a la semántica de su aplicación. Esto nos facilita el mantenimiento y la comprensión del código.

- Funcionalidad

A continuación vamos a ver cuál sería la funcionalidad de los web componentes:

- REUSABILIDAD - Nos permite encapsular funcionalidades específicas en elementos HTML personalizados que pueden ser reutilizados.
- ENCAPSULACIÓN - Shadow DOM proporciona encapsulación al árbol DOM de un elemento, esto quiere decir que el contenido y los estilos dentro de un web component estén aislados del resto de la página.
- DECLARATIVA - HTML Template nos permite definir el comportamiento específico de un elemento HTML personalizado, mediante la implementación de métodos y eventos personalizados.

- INTEROPERABILIDAD.- Los web components son compatibles con los estándares web existentes y que pueden integrarse fácilmente con frameworks y bibliotecas Javascripts.
- MODULARIDAD.- ES Modules nos permite organizar el código Javascript asociado con un web component en archivos separados, lo que facilita la modularidad y el mantenimiento del código.

El código fuente original puede ser similar a la estructura DOM, pero también puede ser muy diferente. Por ejemplo, la figura 2 muestra el código fuente original de una página web HTML simple.

```
<!DOCTYPE html>
<html>
| <body onload="prepare()">
| | <h1>
| | | <p></p>
| | </body>
| <script>
| | function prepare() {
| | | document.querySelector('p').innerText = "This is a dynamic content.";
| | }
| </script>
</html>
```

Fig 2. Código fuente original.

Sin embargo, la estructura DOM es diferente. Como se muestra en la figura 3, hay una sección `<head></head>` anexada por el navegador. El navegador también ejecutó el código JavaScript y modificó el elemento `<p>` añadiendo el texto "This is a dynamic content."

```
<html>
<head></head>
<body onload="prepare()">
<h1> My first Heading </h1>
<p> This is a dynamic content. </p>
<script>
| function prepare() {
| | document.querySelector('p').innerText =
| | | "This is a dynamic content.";
```

Fig 3. Estructura de DOM de página web simple

Shadow DOM soluciona los problemas de CSS y DOM.

Introduce estilos con alcance en la plataforma web. Los

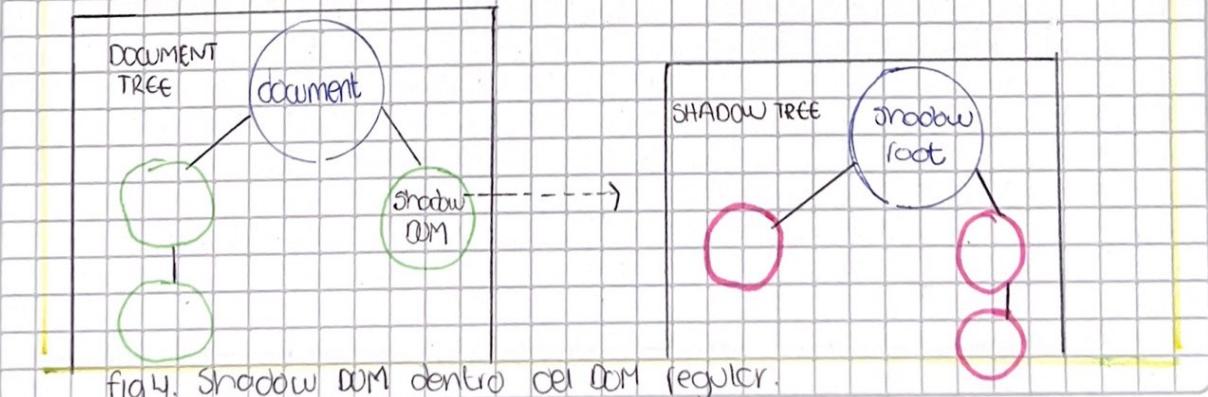
beneficios del Shadow DOM incluyen:

- CSS con alcance: los estilos definidos dentro del Shadow DOM solo se aplican a los elementos dentro de él.
- DOM Aislado: los elementos dentro de un Shadow DOM son invisibles para la página web global.
- CSS simplificado: los nombres de ID y clase de los elementos HTML pueden ser más genéricos, ya que no entran en conflicto con otros elementos fuera de su Shadow DOM.

CREACIÓN DE SHADOW DOM

El shadow DOM puede ser adjuntado a elementos del DOM regular. Estos elementos son: article, aside, blockquote, body, div, footer, h1, h2, h3, h4, h5, h6, header, main, nav, p, section, span y elementos personalizados vinculados.

La figura 4 ilustra la relación entre Shadow DOM y el DOM regular.



3. CONCLUSIONES

- Los Web Components posibilitan la creación de interfaces modulares y reutilizables, lo cual simplifica el mantenimiento y la escalabilidad de las Aplicaciones web. Es decir, estos Componentes se pueden reutilizar fácilmente en diferentes partes de la misma aplicación o incluso en proyectos distintos.
- La implementación de Shadow DOM ofrece una encapsulación eficaz, impidiendo que los estilos y scripts de la página principal perturben los elementos del componente, es así que mejora la estabilidad y aislamiento de los componentes y así permite que se gestionen de forma independiente.
- El uso de HTML Template mejora el rendimiento de las aplicaciones al evitar carga innecesaria de contenido, esto optimiza recursos y permite mayor flexibilidad y eficiencia en la creación de componentes dinámicos que pueden ser instanciados y reutilizados bajo demanda.

4. RECOMENDACIONES

- Los Web Components y los ES Modules pueden trabajar conjuntamente para mejorar la modularidad y la reutilización de código. Los ES Modules nos permite importar y exportar componente de manera eficiente lo que nos facilita su integración y distribución sin depender de herramientas de empaquetados extra.

- Shadow DOM no solo resuelve problemas de adición de estilos, sino también cómo facilita la creación de componentes con un alcance de estilos más controlado y predecible. Así así que nos permite aplicar estilos de forma encapsulada.

Bibliografía

Web Components. (s/f). MDN Web Docs. Recuperado el 5 de noviembre de 2024, de
https://developer.mozilla.org/en-US/docs/Web/API/Web_component

Álvarez, S. (2017, enero 15). *ES Modules: Qué son y cómo puedes empezar a utilizarlos.* Codecoolture.

<https://medium.com/codecoolture/es-modules-qu%C3%A9-son-y-c%C3%B3mo-puedes-empezar-a-utilizarlos-e88c49043593>

(N.d.). Retrieved November 5, 2024, from

http://chrome-extension://efaidnbmnnibpcajpcglclefindmkaj/https://www.researchgate.net/profile/Federico-Gonzalez-Brizzio/publication/320233125_Web_Components_un_nuevo_estandar_para_el_desarrollo_de_aplicaciones_HTML5_Consideraciones_para_su_implementacion/links/59d63286458515db19c4efc5/Web-Components-un-nuevo-estandar-para-el-desarrollo-de-aplicaciones-HTML5-Consideraciones-para-su-implementacion.pdf

Usando shadow DOM. (n.d.). MDN Web Docs. Retrieved November 5, 2024, from

https://developer.mozilla.org/es/docs/Web/API/Web_components/Using_shadow_DOM