

PRACTICA 1

MANEJO Y USO DE LOS ELEMENTOS CON EL DOM

Actividad 1

Aplicaciones
WEB

Matinez Ruth

Basto Tumux Miguel

Introducción

En este documento se reportarán los pasos a utilizar para el manejo y uso de elementos Dom, haciendo uso de la terminal de tu navegador o del VCL. El Dom permite a los lenguajes de programación, como JavaScript, interactuar con el contenido, la estructura y los estilos de una página web de manera dinámica, En esta práctica, exploraremos las capacidades del Dom y cómo podemos utilizarlo para crear páginas web interactivas y dinámicas. Aprenderemos a seleccionar elementos específicos, modificar su contenido y estilo, agregar nuevos elementos a la página y responder a eventos del usuario.

Contenido

Ejercicio 1: Uso del método getElementById y de la propiedad innerHTML	4
Ejercicio 2: Búsqueda de elementos HTML por nombre de etiqueta	6
Ejercicio 3: Encontrar elementos por nombre de clases	7
Ejercicio 4: Búsqueda de elementos HTML mediante selectores de CSS	8
Ejercicio 5: Búsqueda de elementos HTML por colecciones de objetos HTML	11
Ejercicio 6: Agregar nuevos elementos al documento HTML	12
Ejercicio 7: Usando el método addEventListener() para escuchar eventos en la página.....	13
Ejercicio 8: Validación de formulario (requiere formulario en el HTML)	14
Ejercicio 9: Cambiar el valor de un atributo	15
Ejercicio 10: Contenido dinámico	16
Conclusión.....	18

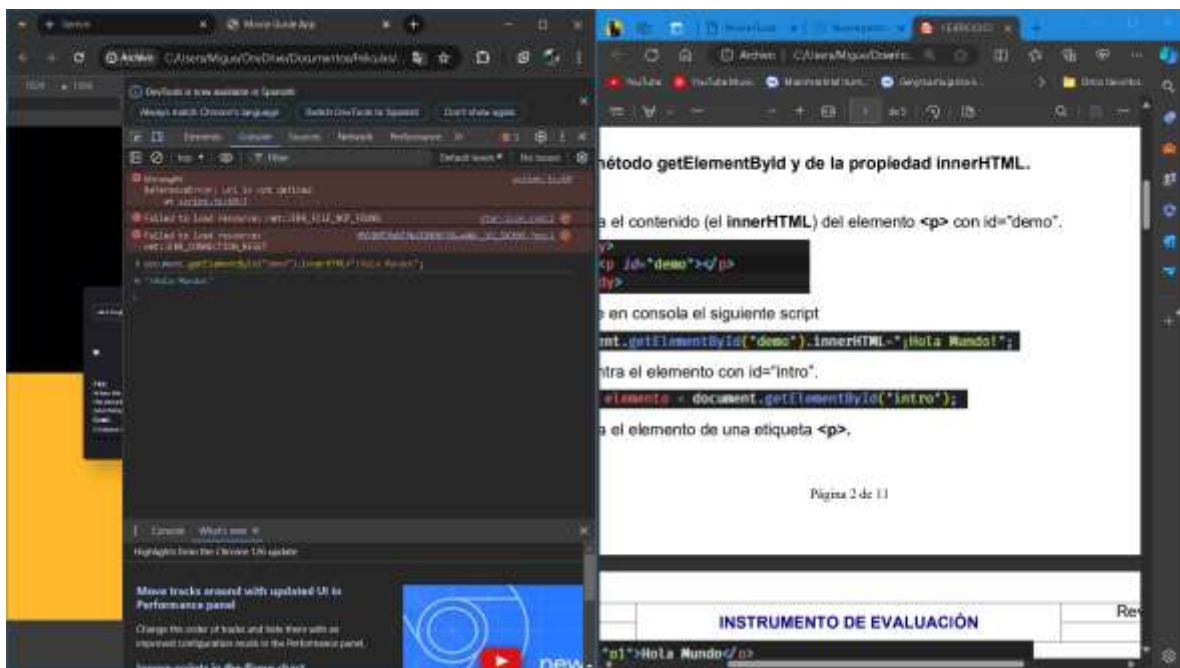
Ejercicio 1: Uso del método getElementById y de la propiedad innerHTML

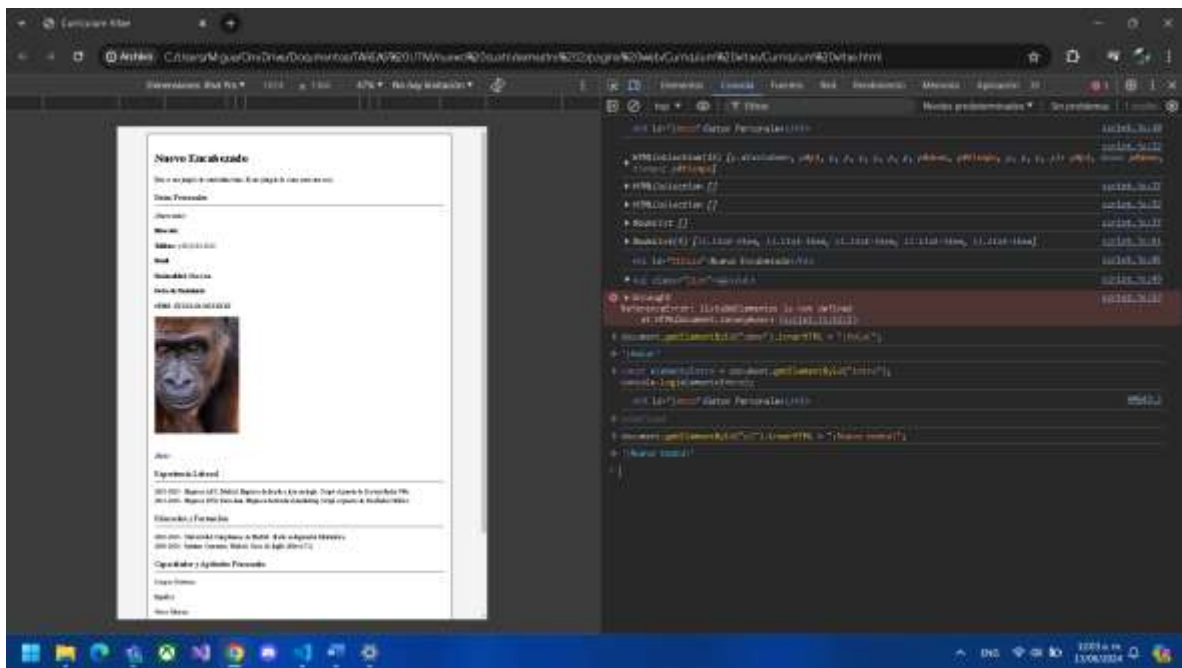
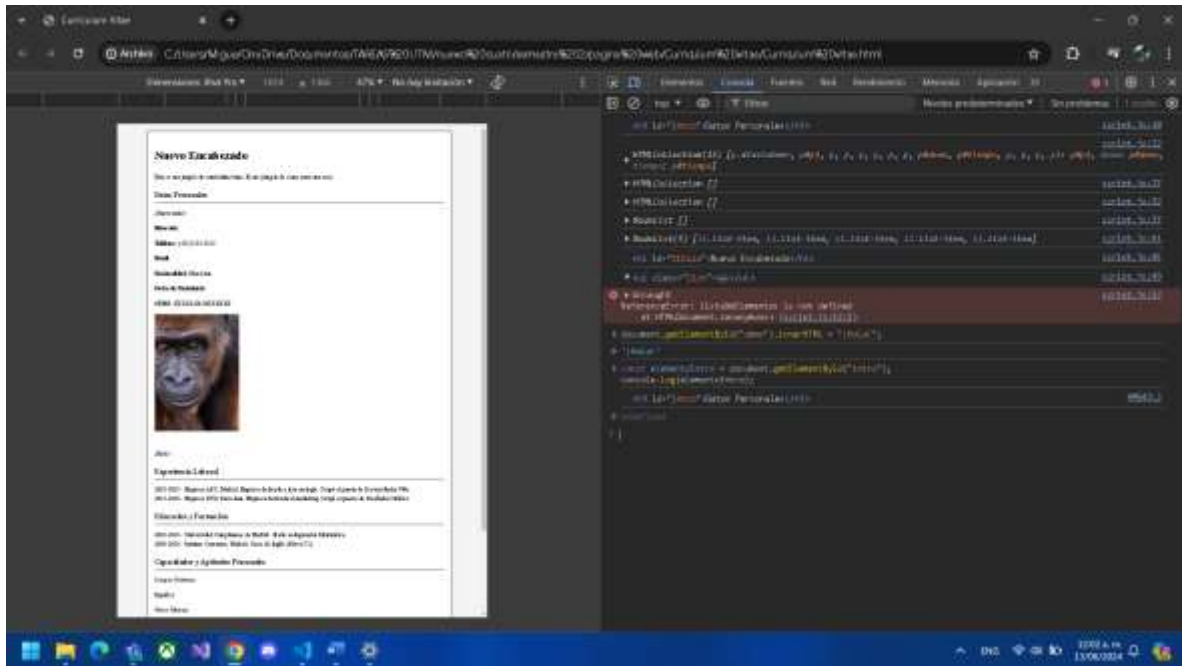
Enunciado:

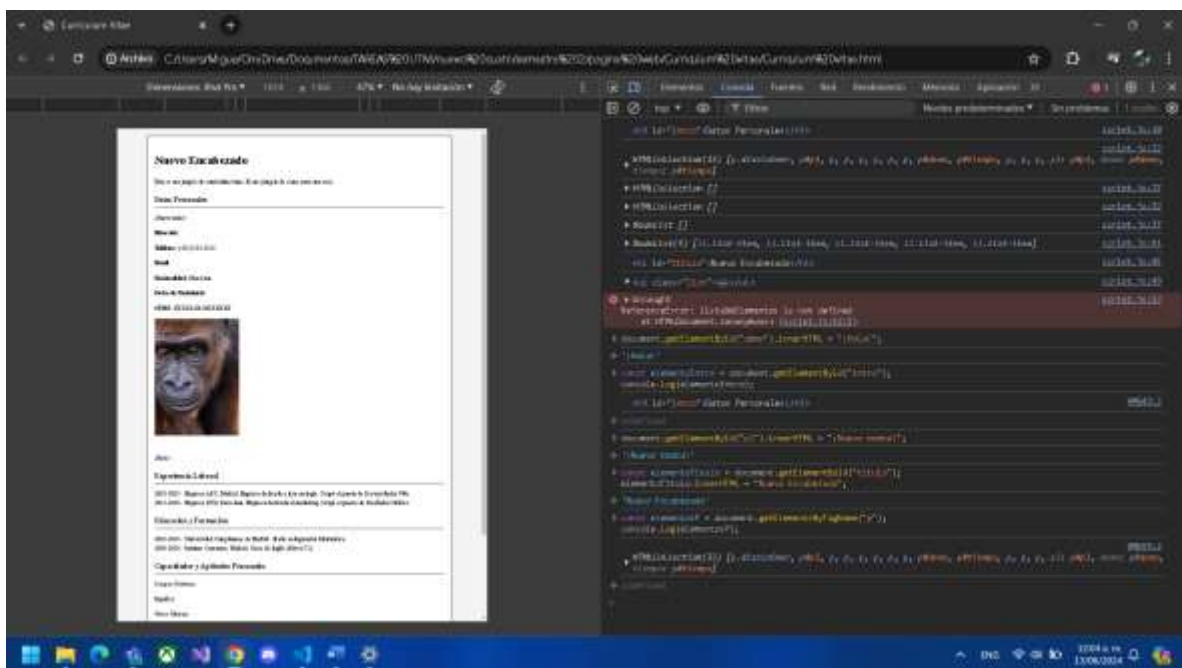
a) Cambia el contenido (el innerHTML) del elemento <p> con id="demo". b) Encuentra el elemento con id="intro". c) Cambia el contenido de una etiqueta <p>. d) Cambia el contenido de un elemento <h1>.

Explicación:

Este ejercicio se enfoca en seleccionar elementos específicos del HTML mediante su ID utilizando el método getElementById. Luego, se modifica el contenido de esos elementos utilizando la propiedad innerHTML, que permite establecer el contenido HTML de un elemento.



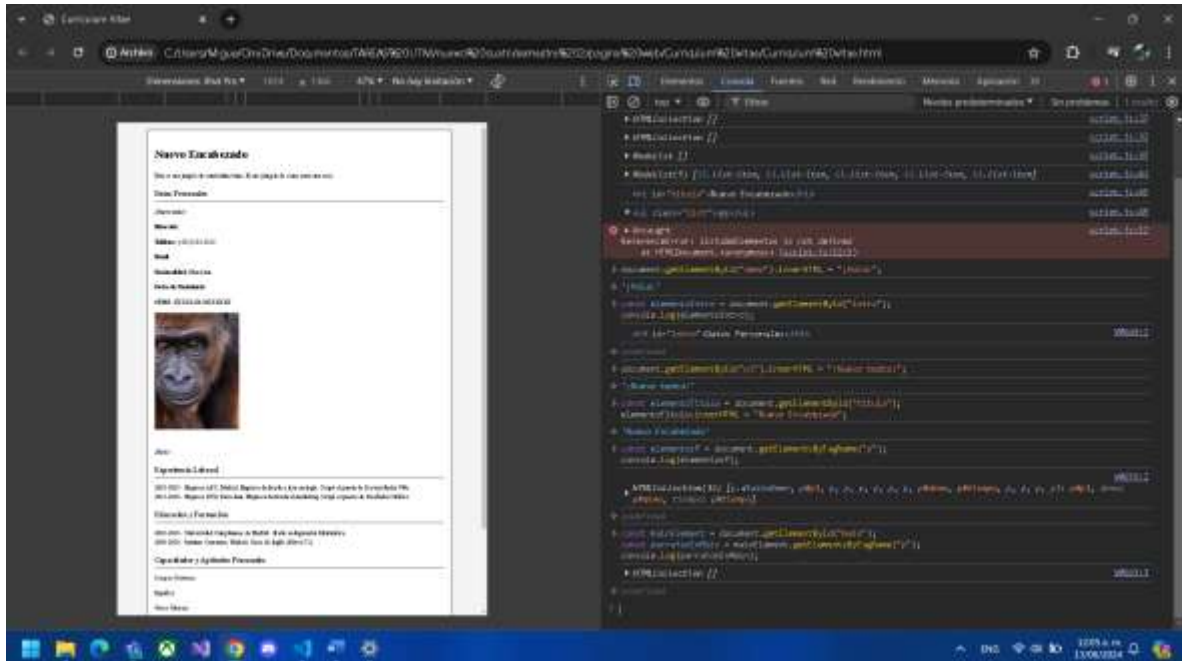




Enunciado:

Explicación:

Este ejercicio enseña a buscar elementos HTML por su nombre de etiqueta utilizando el método `getElementsByName`. También muestra cómo refinar la búsqueda dentro de un elemento específico.



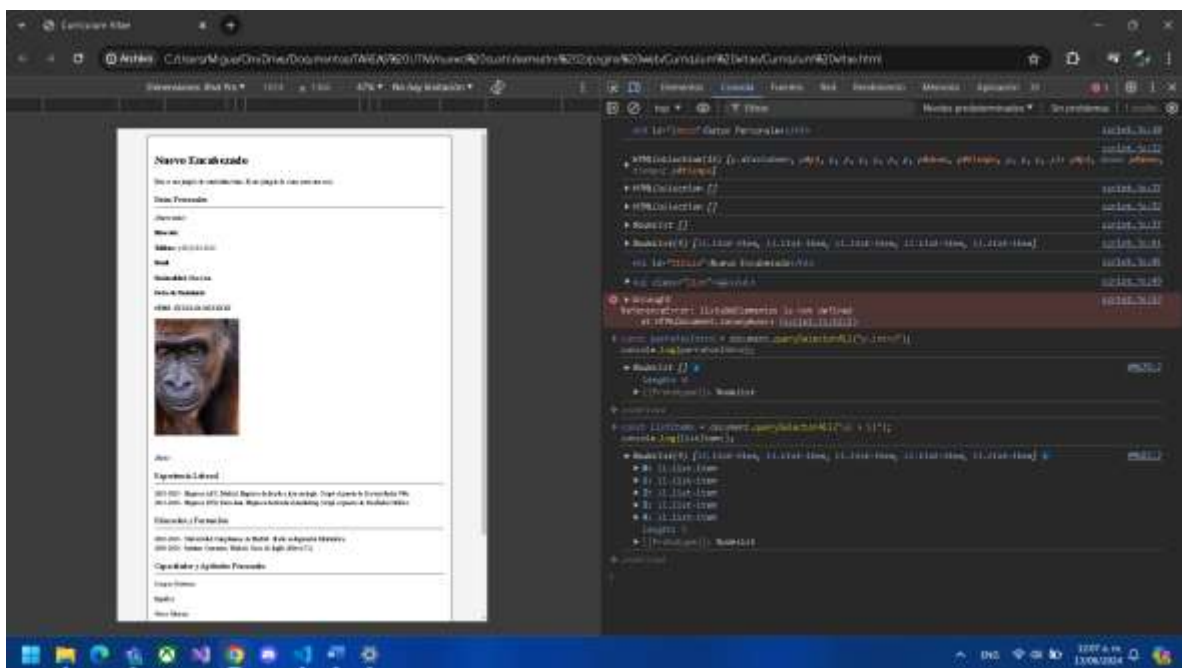
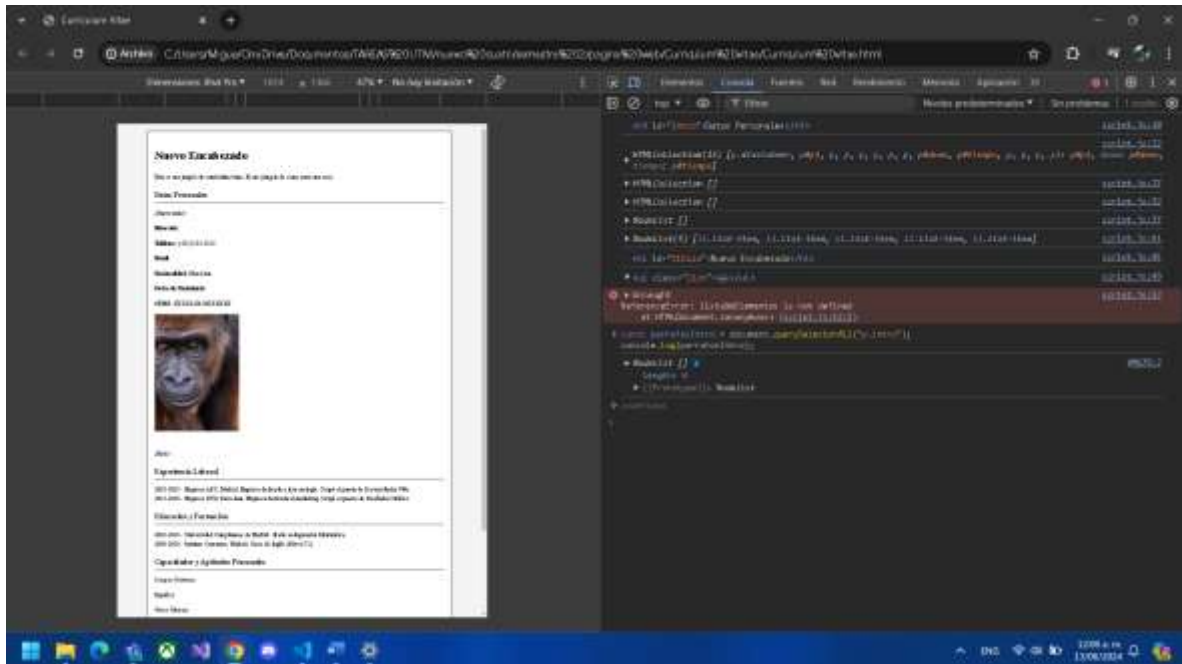
Ejercicio 3: Encontrar elementos por nombre de clases

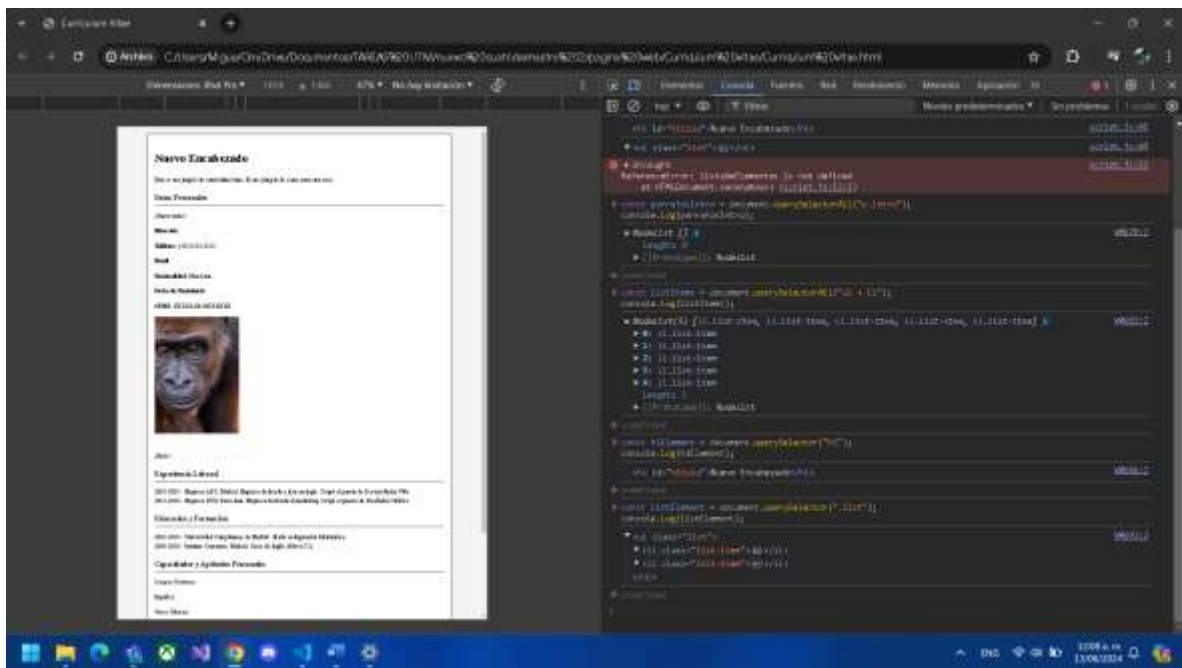
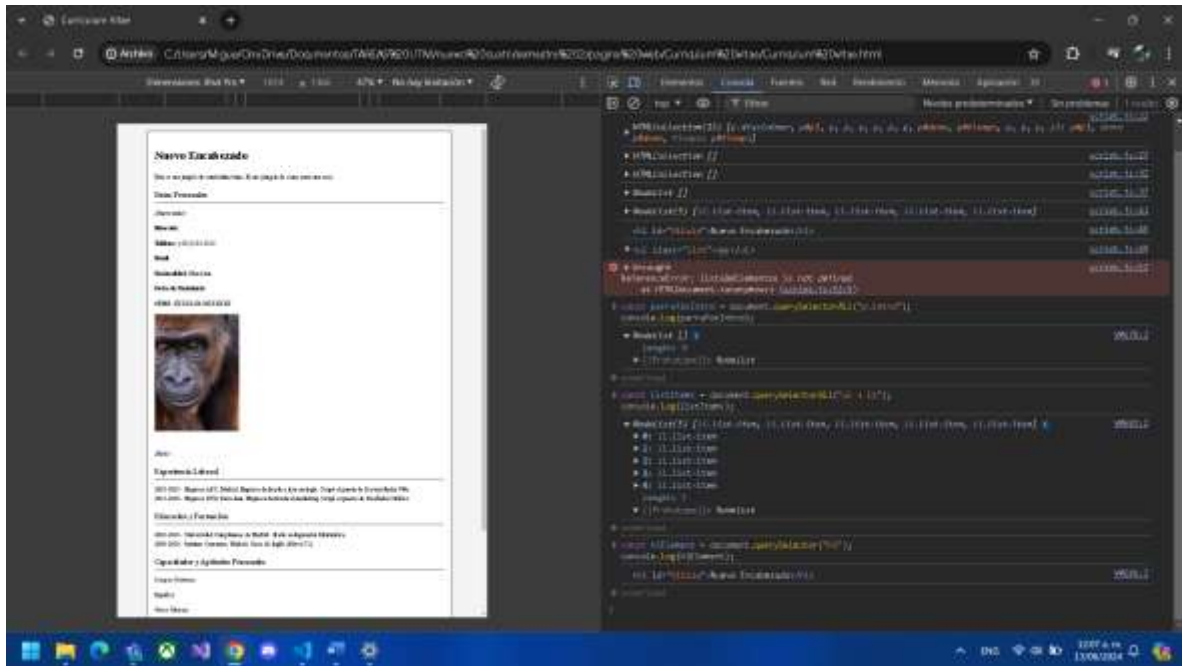
Enunciado:

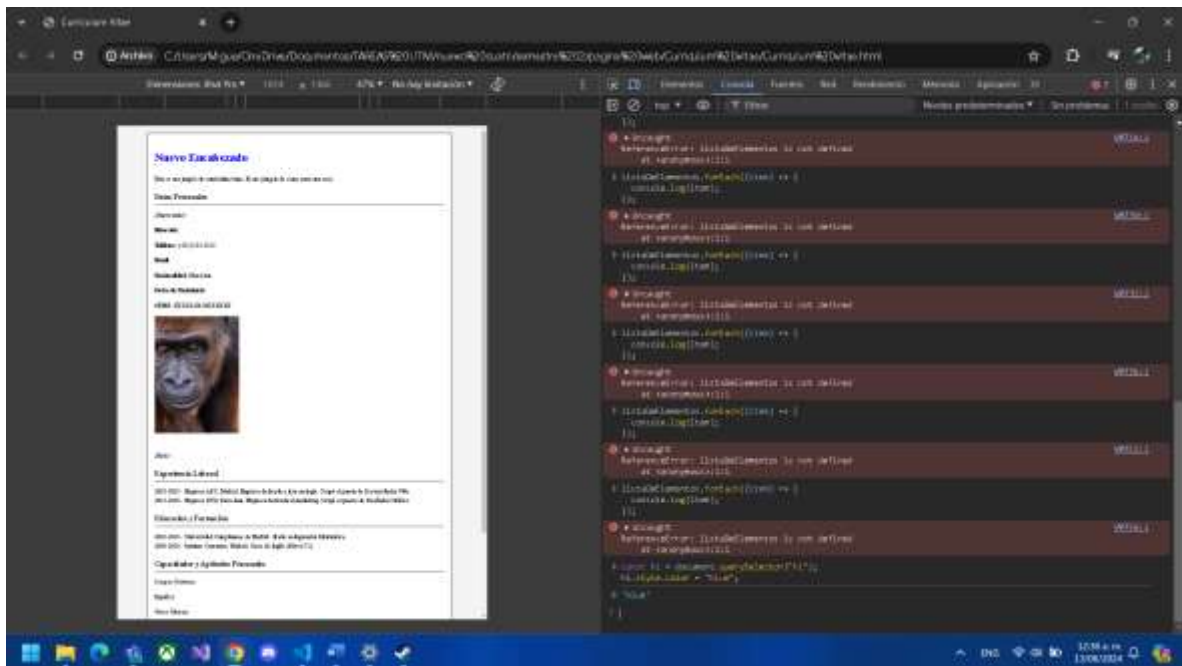
a) Si deseas encontrar todos los elementos con el mismo nombre de clase, usa `getElementsByClassName()`.

Explicación:

Este ejercicio introduce el método `getElementsByClassName` para seleccionar elementos HTML que comparten una clase específica.







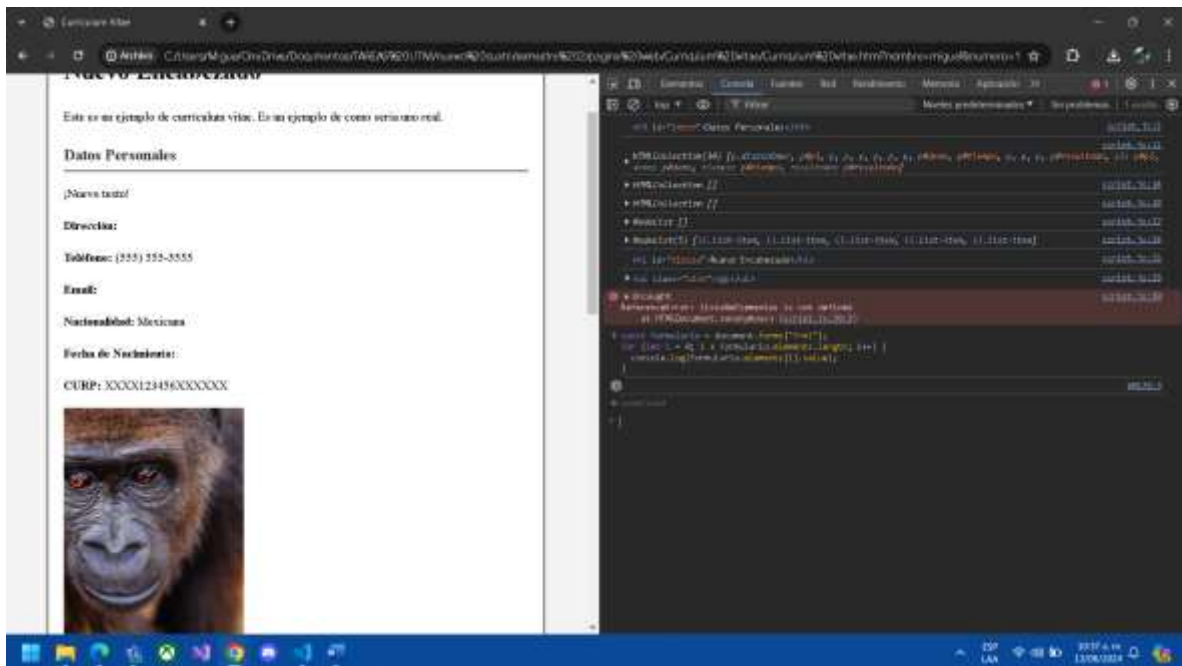
Ejercicio 5: Búsqueda de elementos HTML por colecciones de objetos HTML

Enunciado:

a) Encuentra el elemento de formulario con id="frm1", en la colección de formularios, y muestra todos los valores de los elementos.

Explicación:

Este ejercicio se enfoca en acceder a elementos de formulario a través de la colección `document.forms`. Sin embargo, como no hay un formulario en el HTML, no podemos realizar este ejercicio directamente.



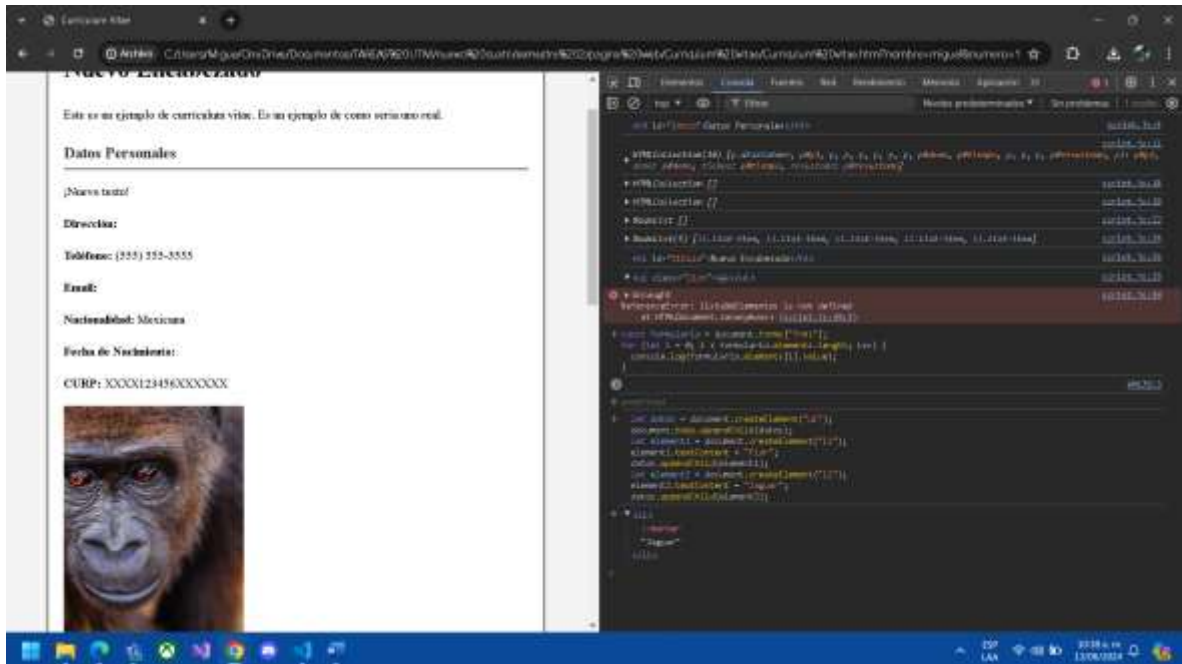
Ejercicio 6: Agregar nuevos elementos al documento HTML

Enunciado:

a) Agregar elementos al árbol del DOM usando los métodos `document.createElement()`, `appendChild()` y haciendo uso de la propiedad `textContent`.

Explicación:

Este ejercicio enseña a crear nuevos elementos HTML desde JavaScript y agregarlos al DOM (Document Object Model), que es la estructura de la página web



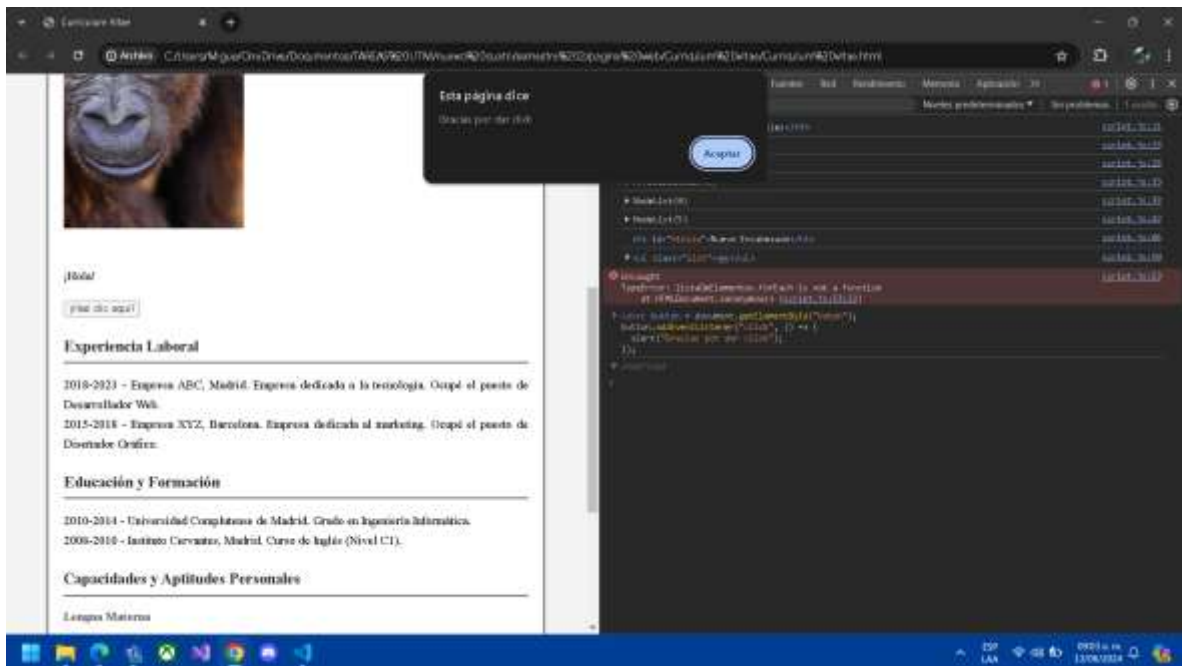
Ejercicio 7: Usando el método `addEventListener()` para escuchar eventos en la página

Enunciado:

a) Crea un botón en el documento HTML, donde al hacer clic aparezca un mensaje de alerta.

Explicación:

Este ejercicio muestra cómo usar el método `addEventListener()` para asociar una función a un evento (en este caso, un clic en un botón).



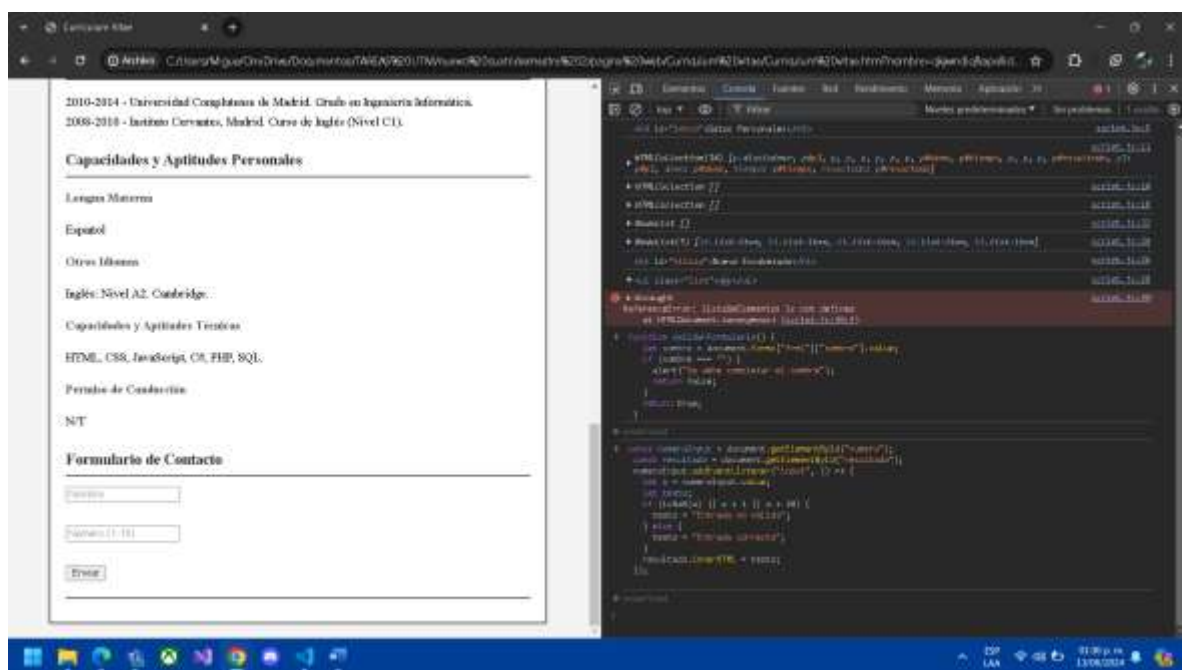
Ejercicio 8: Validación de formulario (requiere formulario en el HTML)

Enunciado:

a) Si un campo de formulario (name) está vacío, muestra una alerta y evita el envío. b) Valida que una entrada numérica esté entre 1 y 10.

Explicación:

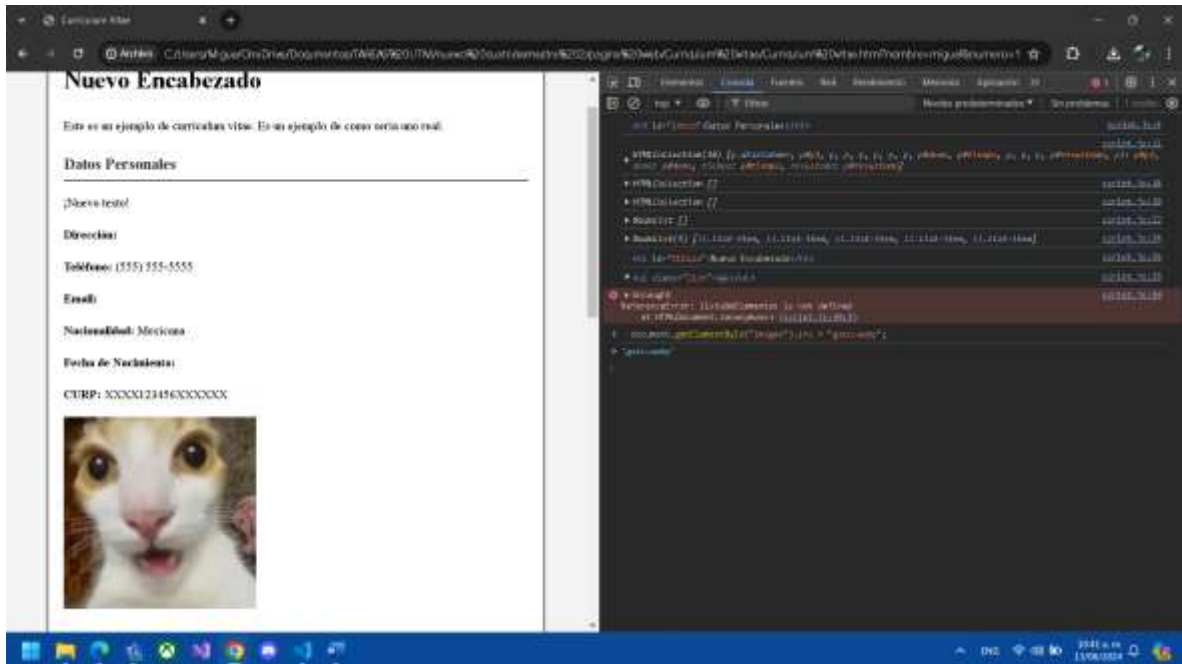
Este ejercicio se enfoca en validar datos de formularios antes de enviarlos. Sin embargo, como no tenemos un formulario, no podemos realizar este ejercicio directamente.



Enunciado:

Explicación:

Este ejercicio demuestra cómo modificar el atributo src de una imagen para cambiar la imagen que se muestra en la página.



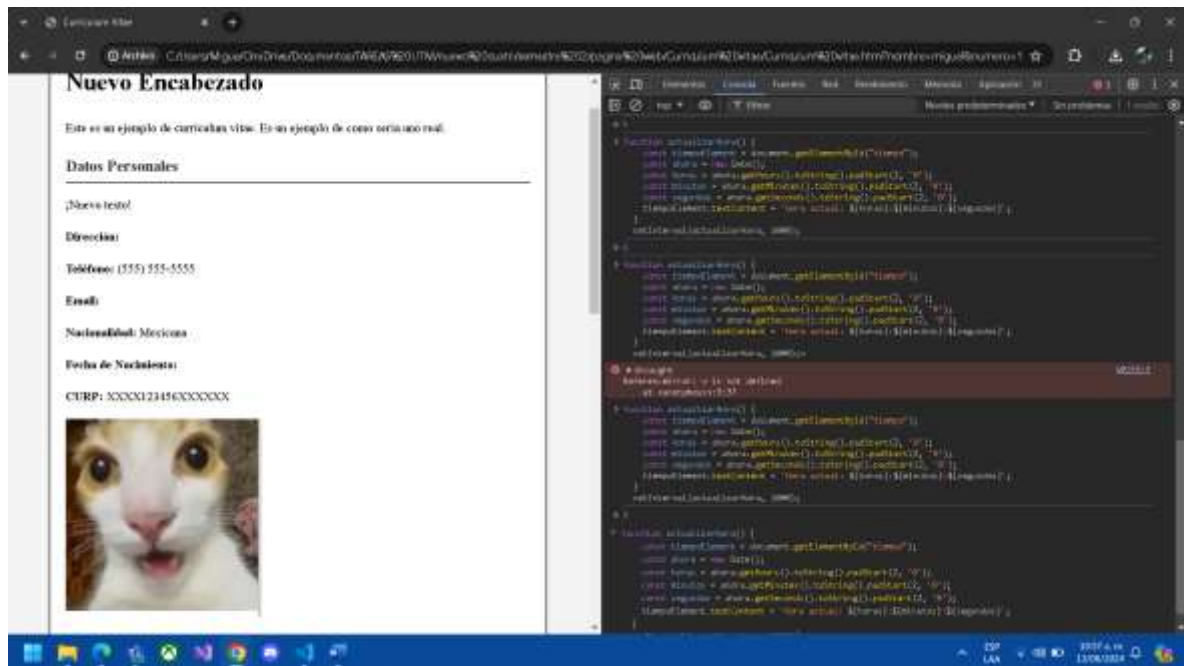
Ejercicio 10: Contenido dinámico

Enunciado:

a) Agregar la hora actual a una etiqueta con id="demo".

Explicación:

Este ejercicio implica actualizar dinámicamente el contenido de un elemento HTML para mostrar la hora actual. Se utiliza la función `setInterval` para ejecutar una función repetidamente en intervalos de tiempo específicos, en este caso, cada segundo.



Conclusión

En esta práctica, hemos explorado a fondo la manipulación del Document Object Model (DOM) utilizando JavaScript. A través de diversos ejercicios, hemos aprendido a seleccionar elementos específicos del HTML, modificar su contenido y estilo, crear nuevos elementos, manejar eventos y validar formularios.

El DOM se ha revelado como una herramienta poderosa para hacer que nuestras páginas web sean interactivas y dinámicas. Hemos visto cómo podemos cambiar el contenido de la página en tiempo real, responder a las acciones del usuario y validar la entrada de datos, todo ello utilizando JavaScript para interactuar con el DOM.

Al principio, algunos conceptos como los selectores de CSS y los eventos pueden haber parecido un poco confusos, pero con la práctica y la experimentación, hemos logrado comprenderlos y aplicarlos de manera efectiva.

En el futuro, estos conocimientos nos serán de gran utilidad para crear páginas web más atractivas y funcionales. Podremos utilizar JavaScript para actualizar el contenido de la página en función de las interacciones del usuario, crear efectos visuales y animaciones, e incluso construir aplicaciones web completas.

En resumen, esta práctica nos ha proporcionado una base sólida en la manipulación del DOM, una habilidad esencial para cualquier desarrollador web moderno. A medida que continuamos aprendiendo y explorando nuevas tecnologías, podremos aprovechar este conocimiento para crear experiencias web cada vez más sofisticadas e interactivas.

