

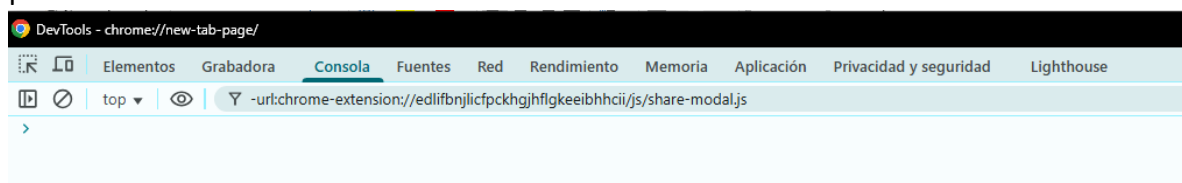
Actividad Introducción al lenguaje JavaScript

Requerimientos específicos Deberás completar una serie de sub-tareas que simulan un escenario de diagnóstico e investigación técnica. Las tareas son:

1. Explora el entorno de ejecución

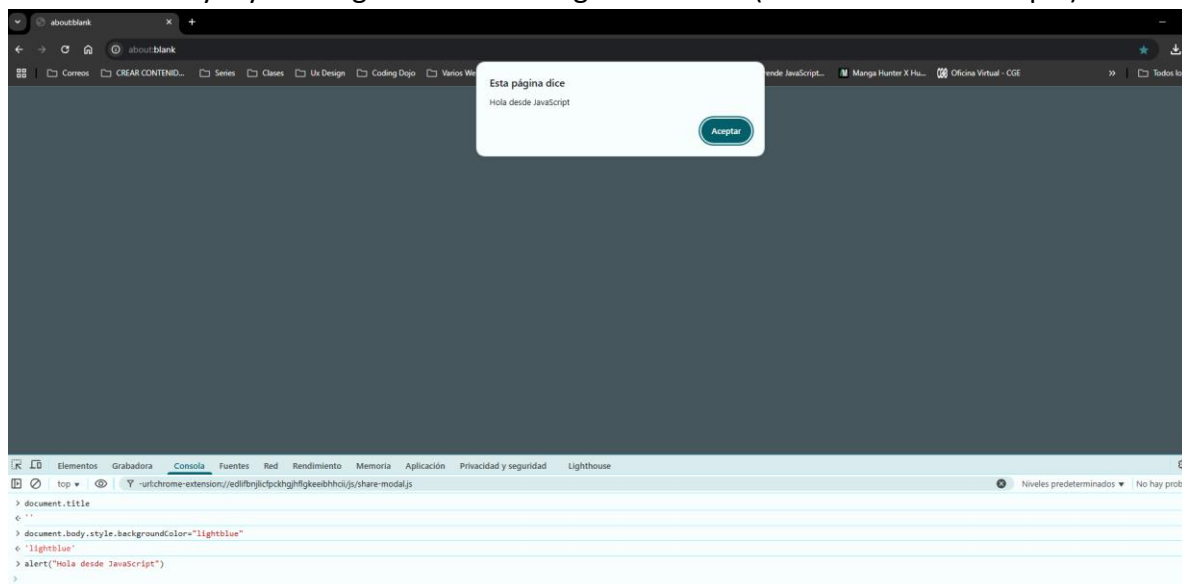
- Abre tu navegador (Chrome o Firefox).
- Accede a las herramientas de desarrollo (DevTools).
- Describe cómo acceder a la consola JavaScript y cómo usarla para probar fragmentos de código.

Botón derecho del mouse entras a inspeccionar, se despliega una ventana y la tercera pestaña es la consola.



2. Interacción con el DOM

- Usando la consola, escribe y ejecuta los siguientes comandos: `document.title`
`document.body.style.backgroundColor = "lightblue"` `alert("Hola desde JavaScript")`



- Explica qué hace cada comando y en qué parte del navegador se ve reflejado el resultado.

`document.title`, no hace nada ya que le falta la acción

`document.body.style.backgroundColor = "lightblue"`, cambie el color de todo la página a lightblue

`alert("Hola desde JavaScript")`, lanza un mensaje de alerta con el texto que se escribió adentro de la alerta

3. ¿Qué puede y qué no puede hacer JavaScript en el navegador?

- Crea una lista comparativa con al menos 5 ejemplos de cosas que JavaScript puede hacer en el navegador (e.g., modificar el DOM, capturar eventos del usuario) y 3 que no puede hacer por razones de seguridad (e.g., acceder al sistema de archivos, modificar directamente archivos del sistema operativo, enviar datos sin permiso del usuario).

JavaScript puede hacer	JavaScript no puede hacer
Reproducir, pausar y controlar audio o video en la página	Acceder a la cámara o micrófono sin el permiso explícito del usuario
Crear animaciones y efectos visuales usando CSS o canvas	Interactuar con otras pestañas o ventanas del navegador sin permiso
Detectar el tamaño de la pantalla o si el usuario está en móvil o desktop	Saber la ubicación exacta del usuario sin su consentimiento
Crear y modificar elementos <canvas> para gráficos o juegos	Guardar archivos automáticamente en el disco sin intervención del usuario
Utilizar Web APIs como Geolocation, Vibration o Clipboard con permiso	Instalar programas en el computador del usuario

4. Prueba errores comunes

- Intenta ejecutar código incorrecto en la consola (por ejemplo: `console.logg("Hola")`, `document.querySelector("###")`) y describe los mensajes de error que aparecen.

```
> conso
✖ ▶ Uncaught ReferenceError: conso is not defined
   at <anonymous>:1:1

> console.logg("Hola")
✖ ▶ Uncaught TypeError: console.logg is not a function
   at <anonymous>:1:9

> document.querySelector("###")
✖ Uncaught SyntaxError: Unexpected token ')'
```

En el primero dice que conso no esta definido

En el segundo dice console.logg no es una función

Y el tercero que hay un error de sintaxis y que hay un carácter inesperado en el código.

- Explica cómo estas herramientas ayudan a depurar problemas.

Las herramientas del navegador te muestran qué está mal, dónde ocurrió el error y cómo arreglarlo. Son muy útiles para entender y corregir problemas rápidamente, incluso si estás empezando.

5. Herramientas para desarrollo JavaScript

- Investiga y menciona al menos dos extensiones o entornos populares para desarrollo en JavaScript en el navegador (por ejemplo: Live Server, Visual Studio Code DevTools Extension).

Live server Es una extensión para Visual Studio Code que permite abrir un proyecto en el navegador y ver los cambios en tiempo real cada vez que se guarda

DevTools Extension Son herramientas que vienen integradas en navegadores

- Explica brevemente su utilidad en el flujo de trabajo de desarrollo web.

Live server es útil porque acelera el desarrollo y evita tener que recargar manualmente la página.

DevTools es útil porque permiten inspeccionar el HTML y ver errores de JavaScript, probar código en la consola y depurar paso a paso.