**CICLO: DESARROLLO APLICACIONES WEB-DISTANCIA**

**MÓDULO: DESARROLLO WEB EN ENTRONO CLIENTE**

**TAREA: DWEC-01**

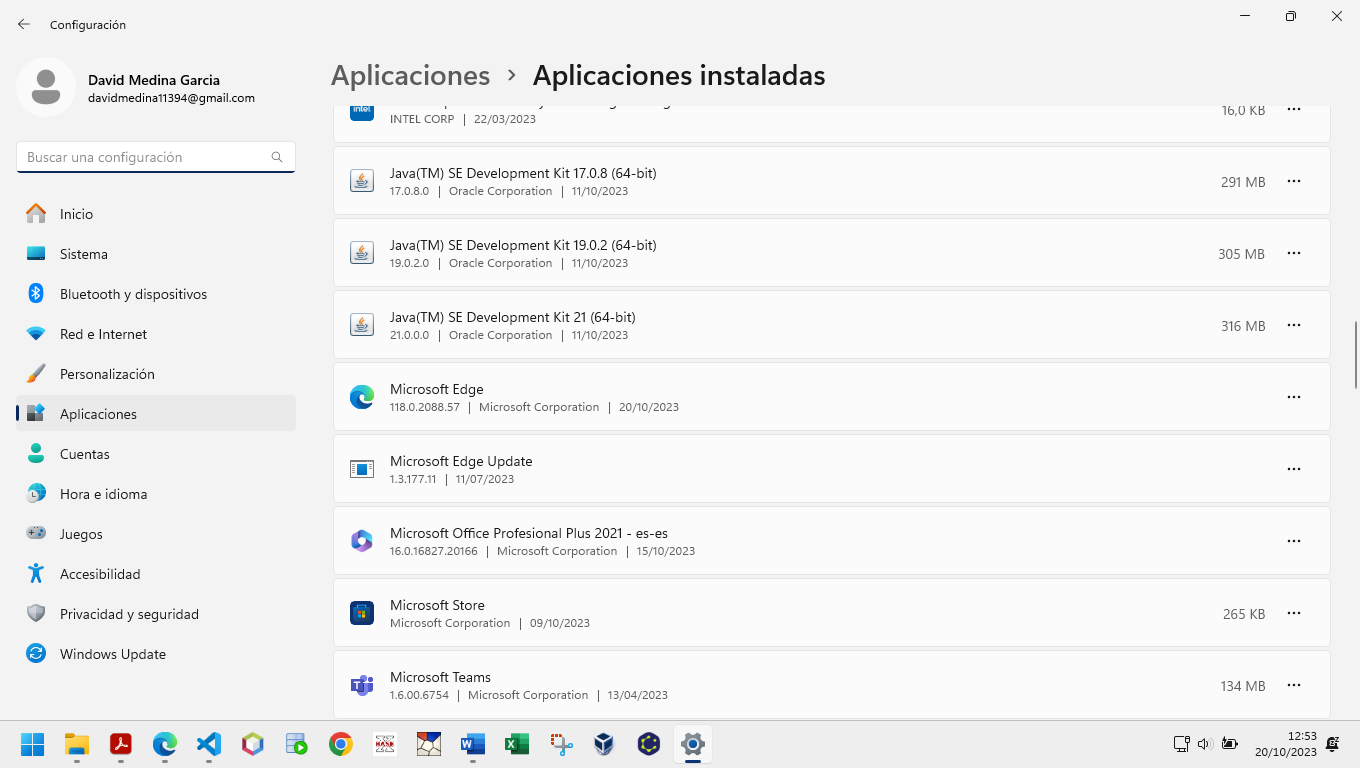
**ALUMNO: DAVID MEDINA GARCIA**

1. **Instalar navegadores adicionales.**

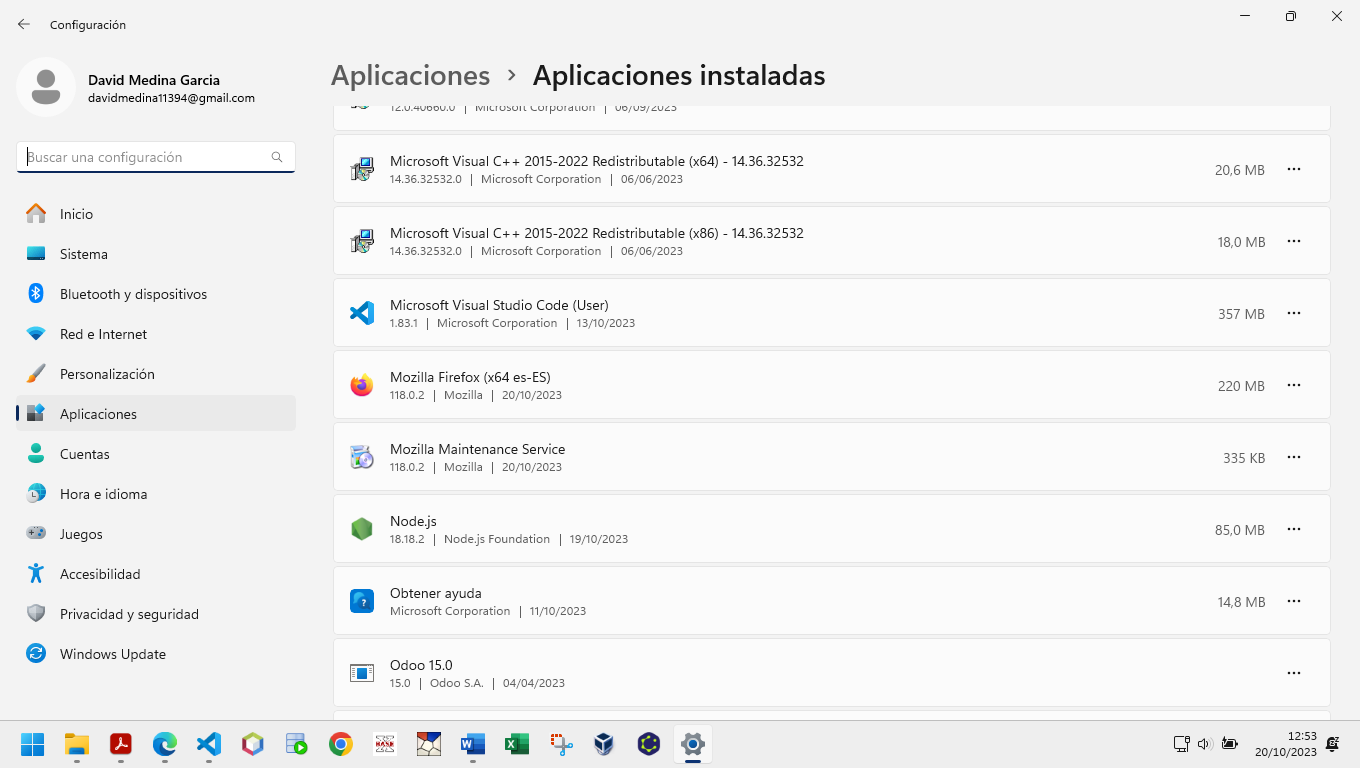
Navegadores adicionales instalados: Microsoft Edge y Mozilla Firefox. Chrome estaba previamente instalado. Desde google buscamos la pagina web de cada navegador, nos lo descargamos y lo instalamos con el asistente que se ejecuta, tal como se muestra en las siguientes imágenes.

**Navegadores intalados**

**Microsoft Edge**



**Mozilla Firefox**



1. **Instalación de editor web. Razones para elegir el editor web.**

**Editor elegido**: Apache Netbeans

Después de buscar información de varios programas para comparar, me decidí por Apache Netbeans por las siguientes razones:

**-Es gratuito**: Puedes acceder a todas sus funciones sin necesidad de realizar ningún pago

**-Sencillez:** Es más sencillo de usar que otras alternativas.

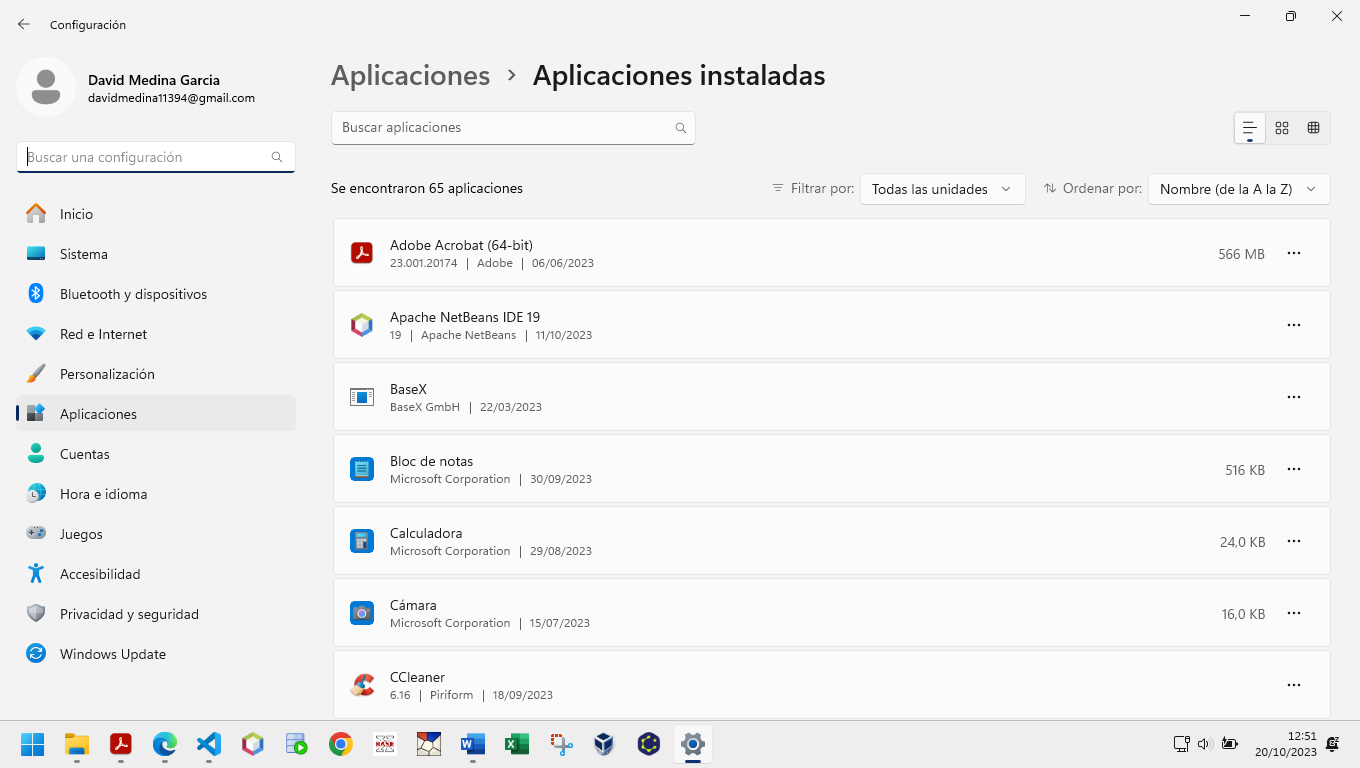
**-Multiplataforma:** Se puede emplear en distintos dispositivos, así como ejecutarse en diferentes sistemas operativos (Mac OS, Windows, Linux y Solaris).

**-Multilenguaje:** Permite desarrollar aplicaciones multilenguaje. Es decir, no solo opera con Java; sino que Eclipse puede utilizar otros lenguajes como PHP, Python, C o Ruby, entre otros.

Una razón adicional para elegir este editor es que es utilizado en otras asignaturas del ciclo y así estaremos más familiarizados con él.

Para instalarlo nos vamos a su pagina web , nos lo descargamos y lo instalamos con el asistente que se ejecuta.

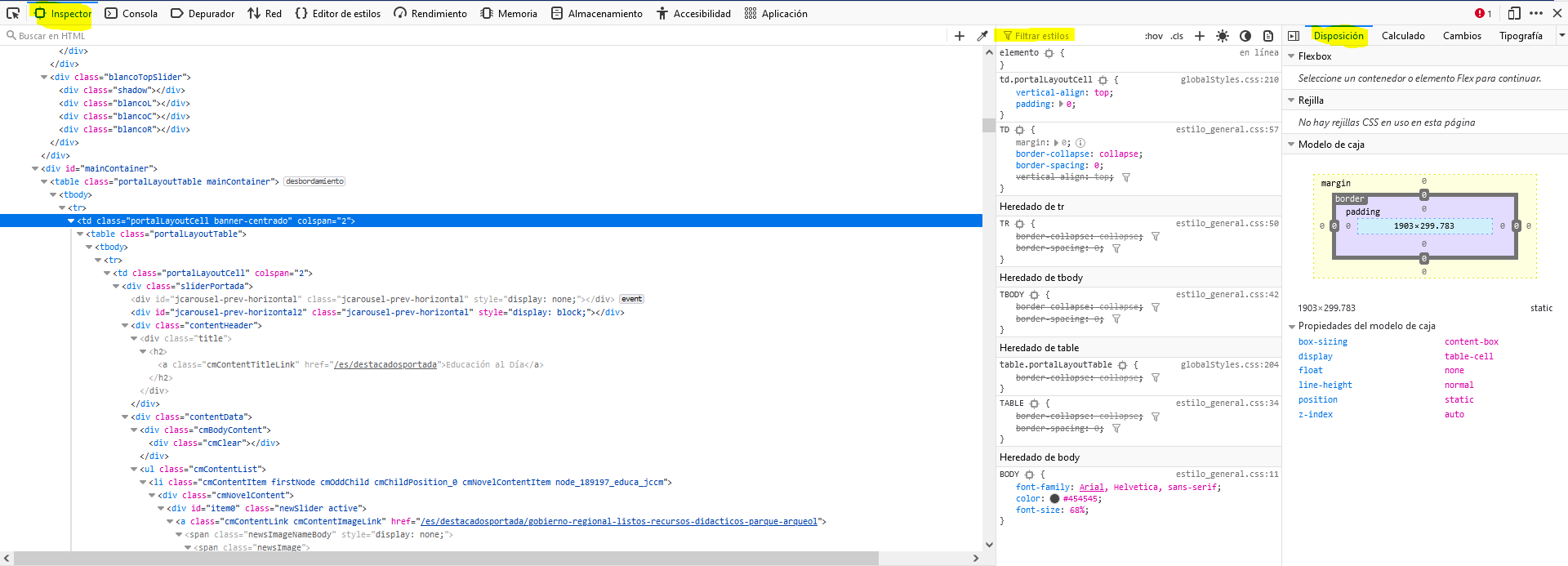
**Apache Netbeans**



**Herramienta de desarrolladores:**

* **Firefox**

Podemos abrir la herramienta para desarrolladores pulsando F12:

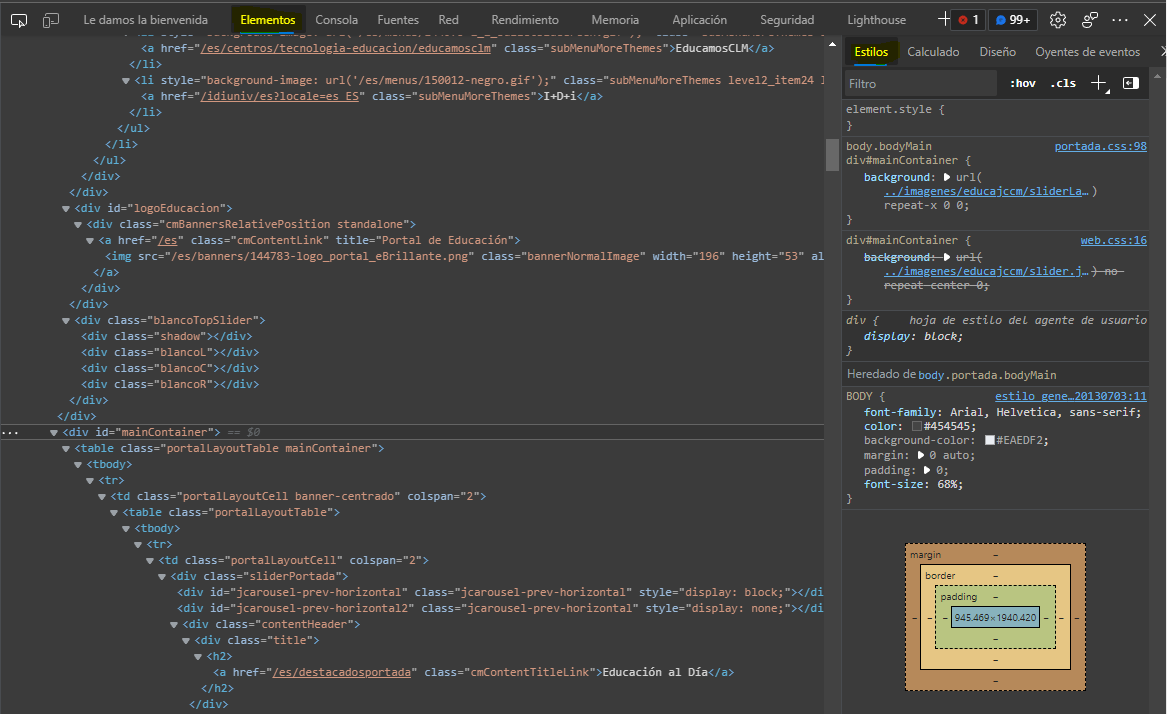


Se nos abrirá una ventana en la parte inferior de la página abierta en la pestaña **Inspector** (que vendría a ser lo mismo que la pestaña **Elementos** en Edge y Chrome). Desde ahí podemos ver todo el código **HTML** de la página (se puede editar el código **HTML** desde aquí y ver los cambios al instante). En la pestaña Consola podemos probar scripts de **Javascript**.

A la derecha tenemos una pestaña que hace referencia a los estilos. Al seleccionar un elemento en esta pestaña nos mostrará los estilos que tiene asociados. Destacamos la pestaña **Disposición** de la derecha, donde podemos observar la disposición del elemento.

* **Microsoft Edge**

Podemos abrir la herramienta para desarrolladores en pulsando F12:

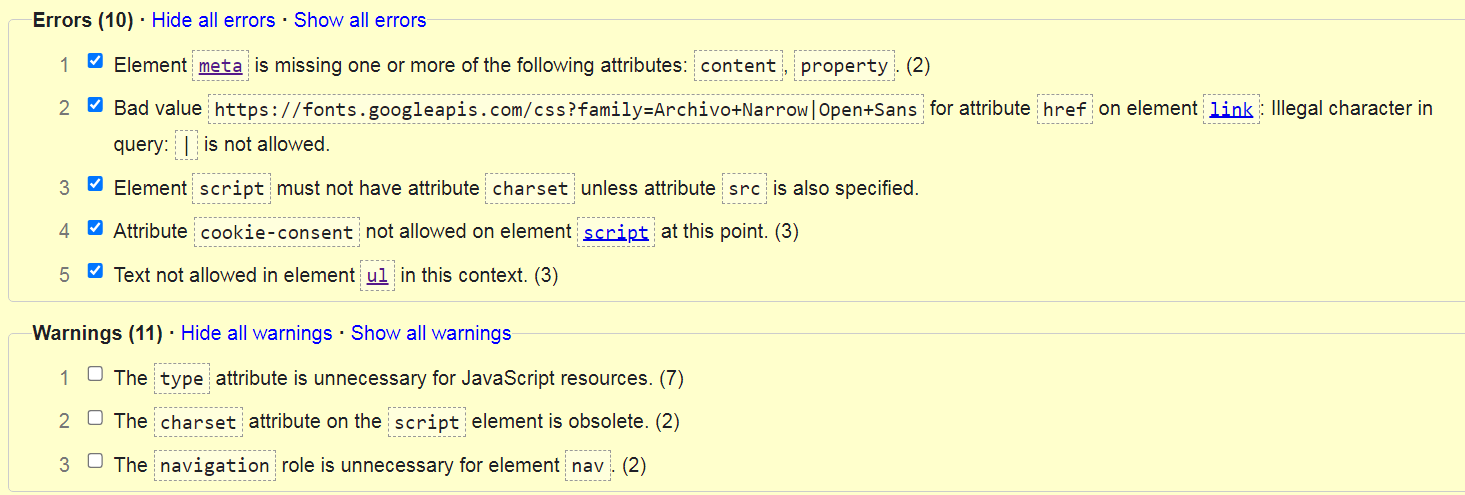


Se nos abrirá una ventana en la parte derecha de la página abierta en la pestaña **Elementos** para ver todo el código **HTML** de la página (se puede editar el código **HTML** desde aquí y ver los cambios al instante). En la pestaña Consola podemos probar scripts de **Javascript**.

A la derecha tenemos la pestaña **Estilos**. Al seleccionar un elemento de la pestaña **Elementos**, nos mostrará los estilos que tiene asociados, así como la disposición del mismo.

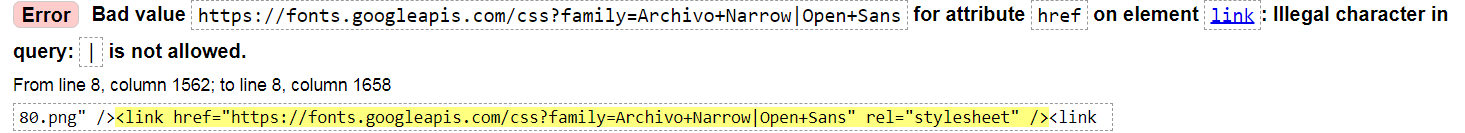
1. **Validación página web y corrección de errores**

Usando la dirección de validación del código HTML del W3C (validator.w3.org) realizaremos la validación de la página de la Universidad de Castilla-La Mancha (www.uclm.es) e indicaremos los tipos de errores encontrados, nos ha dado **10** errores y **11** warnings.



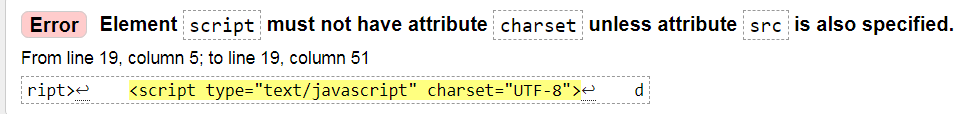
A continuación se propone una solución para 3 de estos errores:

* **Error 1:**

****

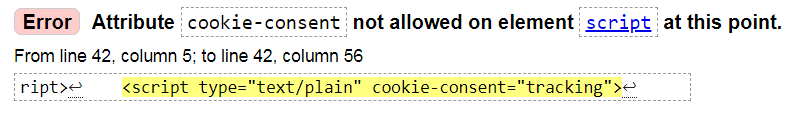
En este error vemos que existe un carácter no permitido (|) en la consulta del atributo **href**. Para solucionarlo debemos sustituir dicho carácter (|) por su codificación “%7C”.

* **Error 2:**

****

Este error nos indica que el atributo **charset** no debería estar si no lo indica el atributo **src**. La solución pasa por eliminar el atributo **charset** ya que no es necesario usarlo en esta ocasión.

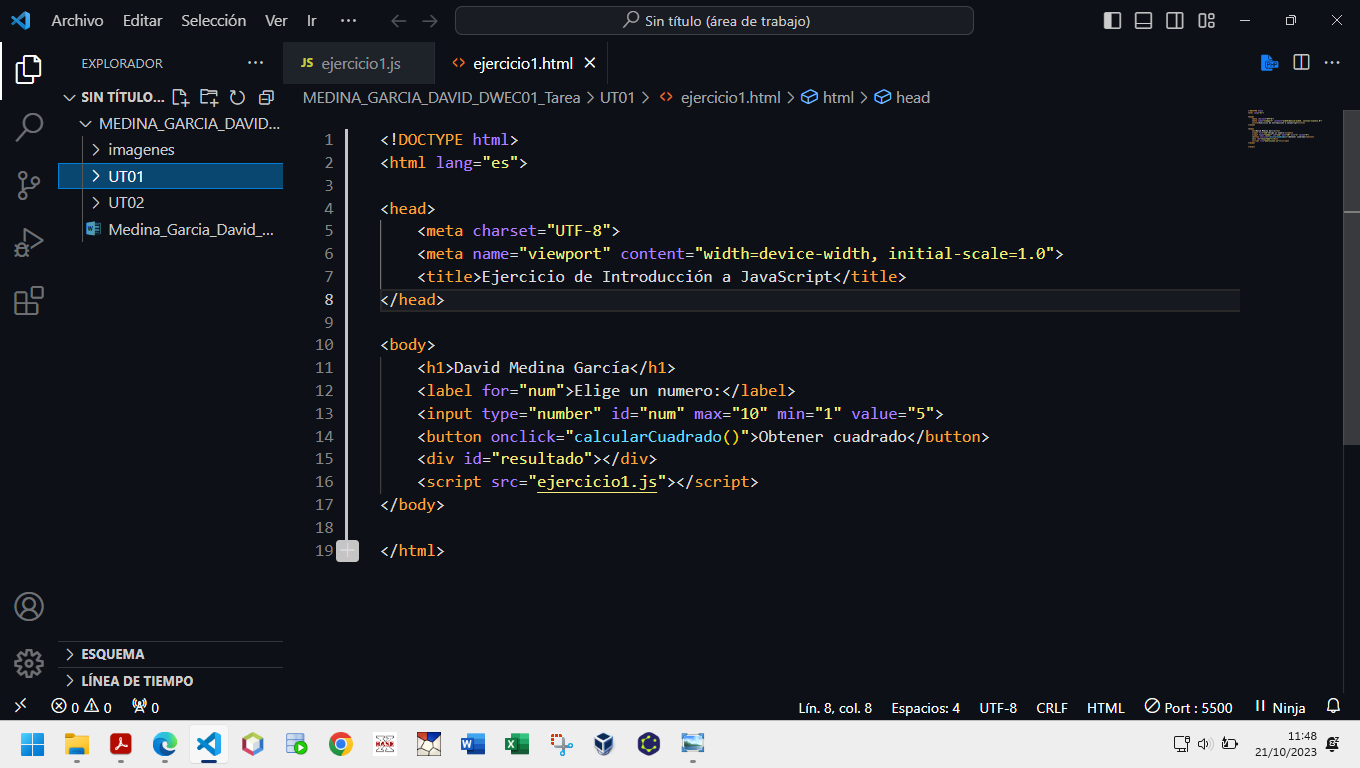
* **Error 3:**

****

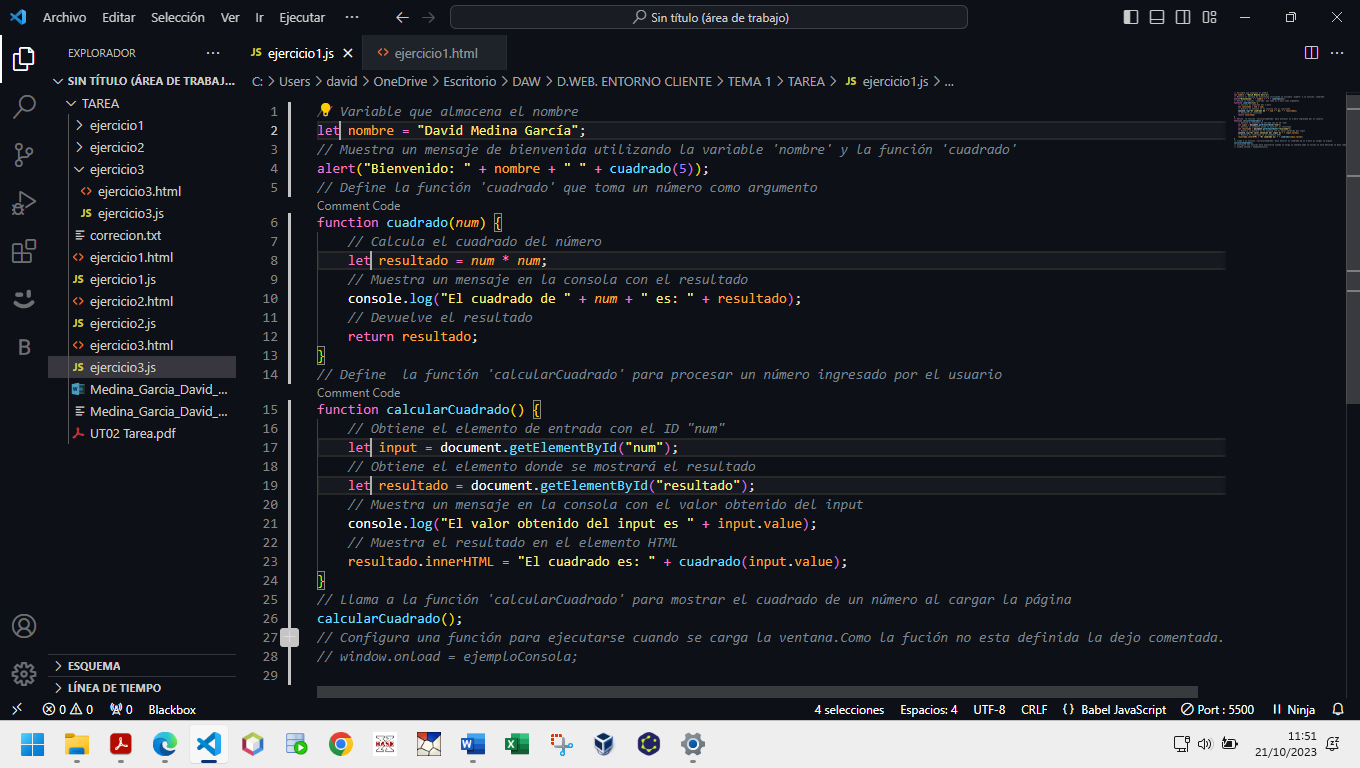
Este error nos indica que el elemento **script** no permite que este el atributo **cookie-consent.** Una solución sería usar un atributo personalizado (**data-cookie-consent=”tracking”).**

1. **Ejercicio UT01.1: Integrar el código JavaScript en nuestro sitio web**

* **CODIGO HTML**

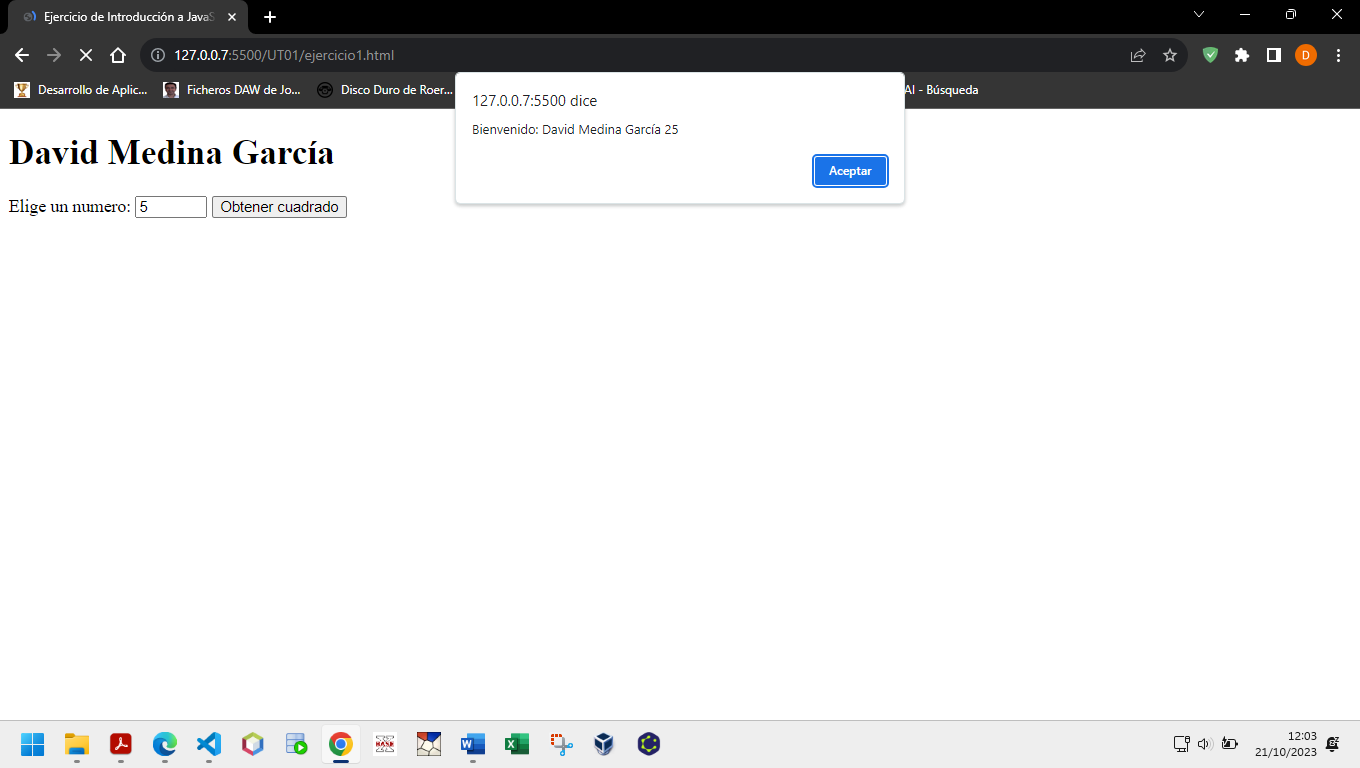


* **CODIGO JAVASCRIPT**

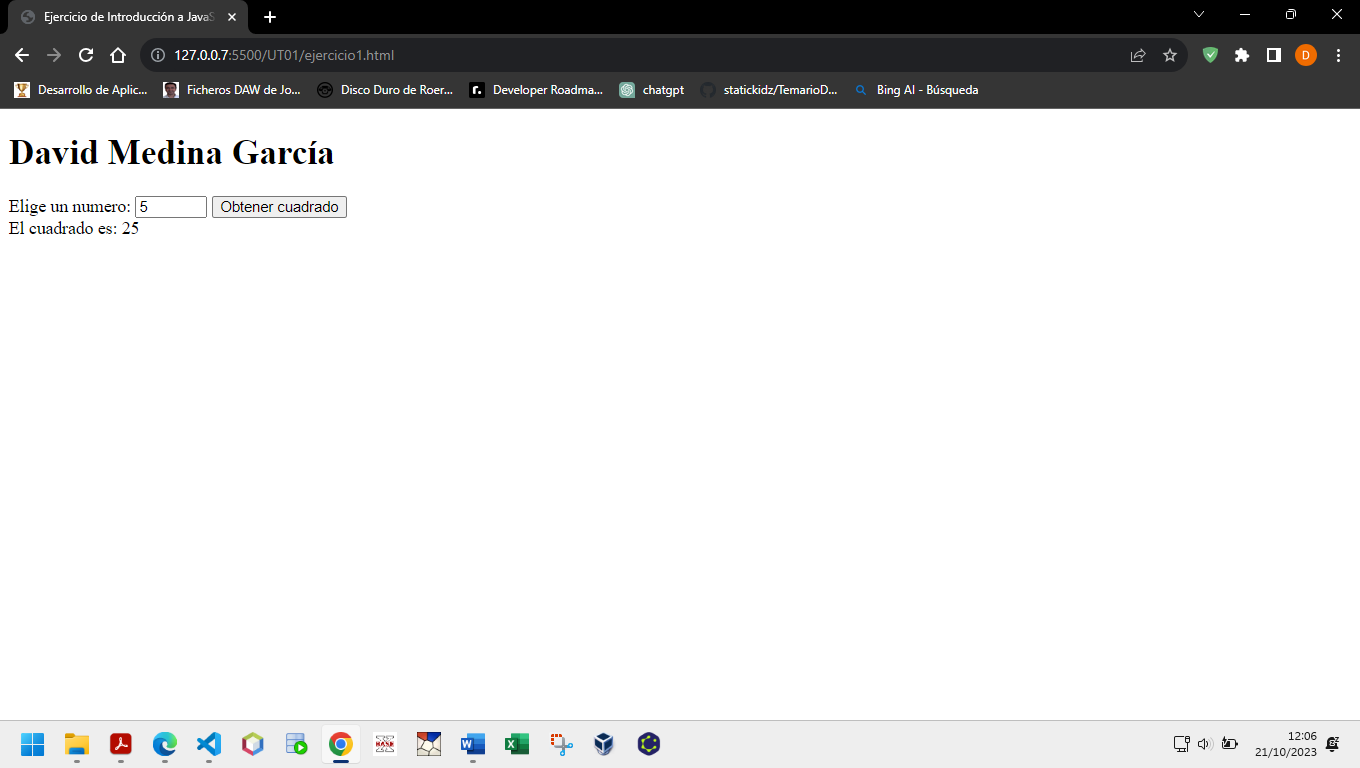


* **RESULTADO**

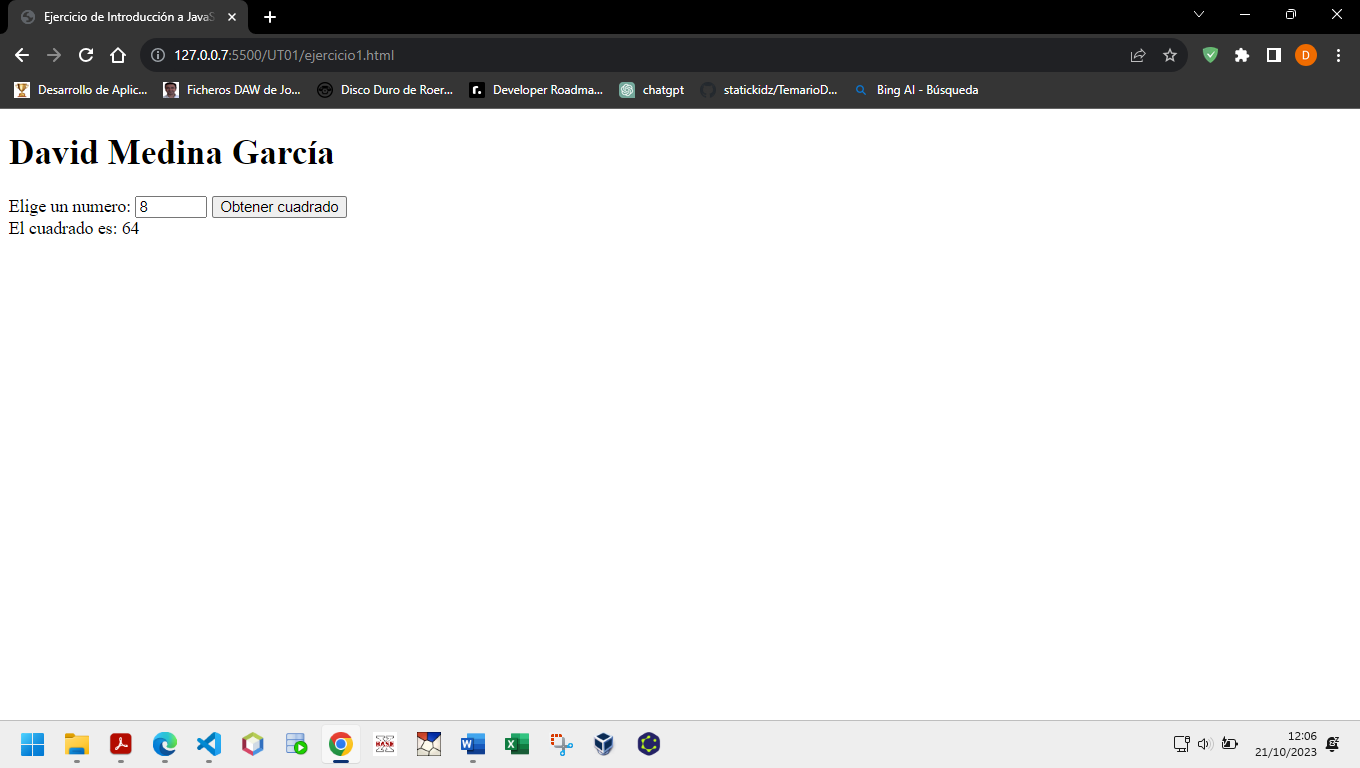
-Primero se carga la página y salta el método **alert().**



-A continuación, se muestra la página cargada, habiendo llamado a la función **calcularCuadrado**().



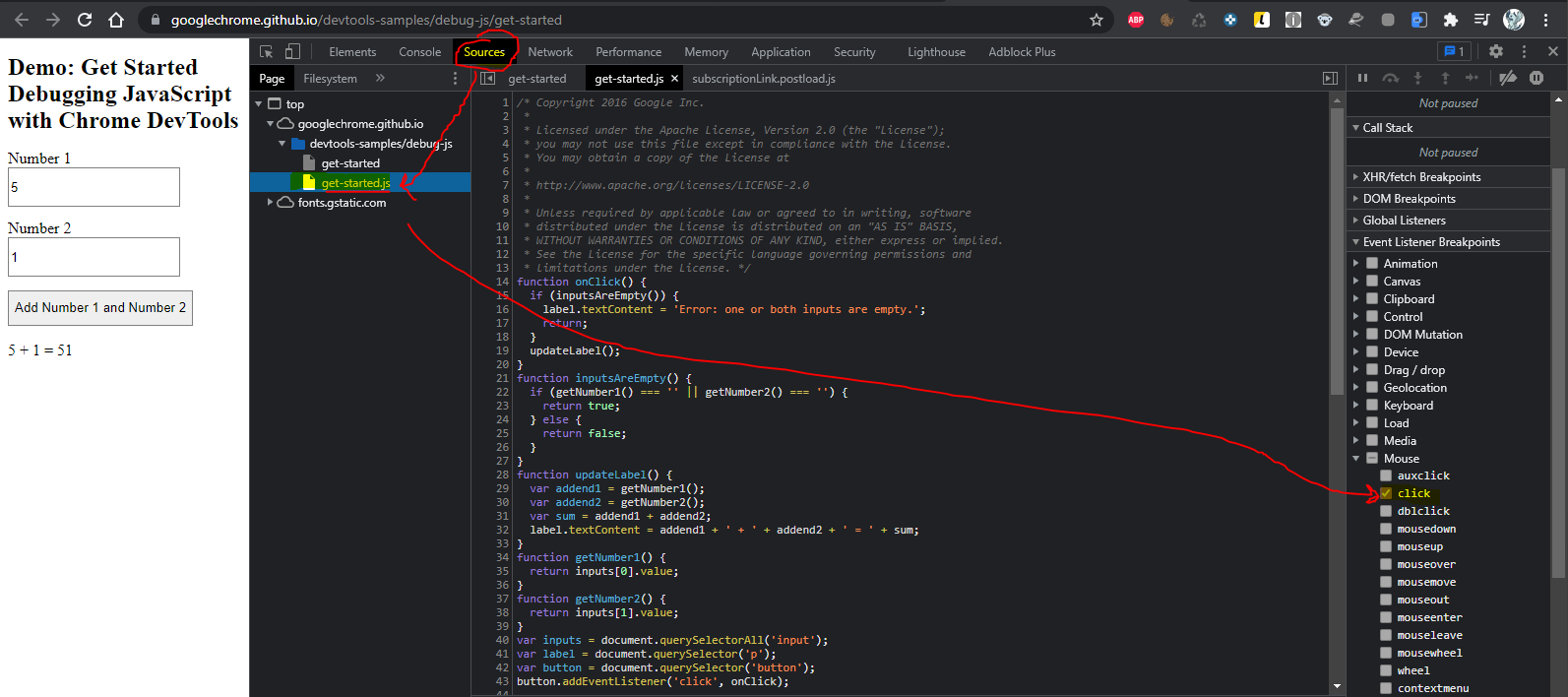
-Y pulsamos el botón para que haga la raíz cuadrada del número insertado



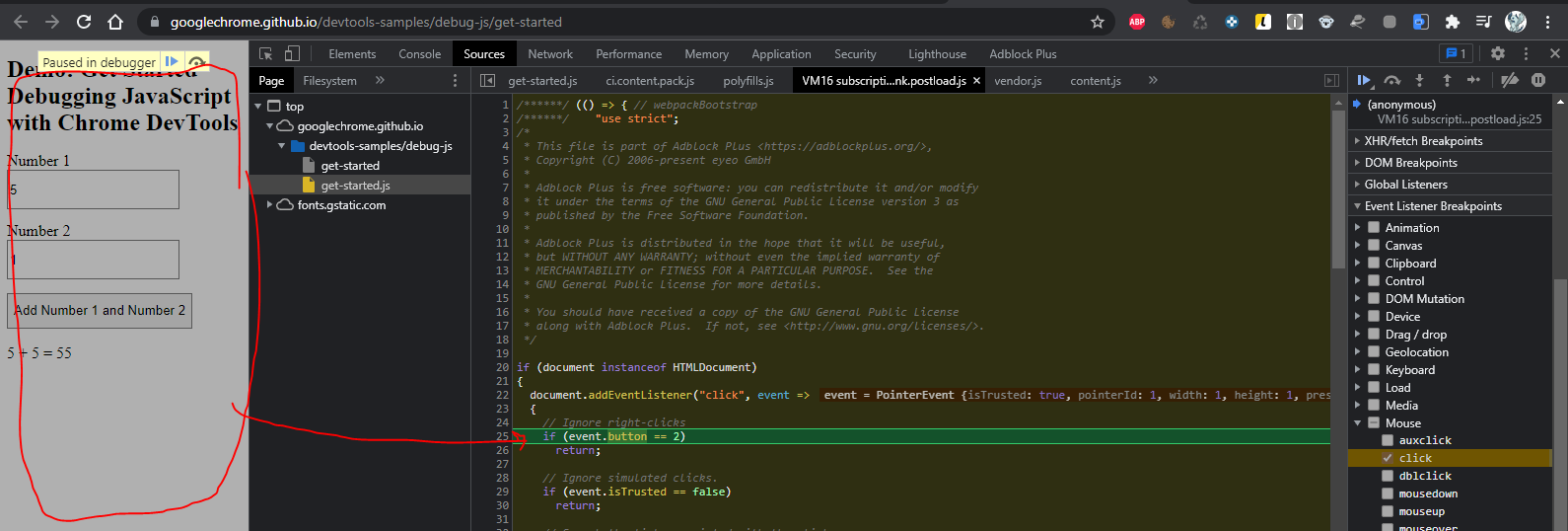
1. **Ejercicio UT01.2: Utilización del depurador**

Para este ejercicio he usado mi navegador habitual, Chrome.

Abrimos la página del ejemplo, accedemos a la Herramienta para desarrolladores (F12) y vamos a la pestaña **Sources** y seleccionamos el archivo **.js**. Abrimos **Event Listener Breakpoints** **>** **Mouse** > **Click** para indicar que debe realizarse un punto de ruptura ante la detección de un click.



Si hacemos un click en cualquier parte de la página, detectará el evento.

****

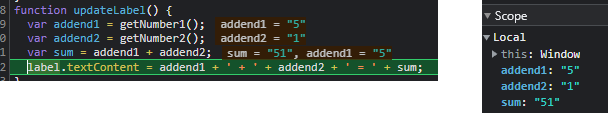
Normalmente nos interesa evaluar un click válido (una función **onClick**) o un botón concreto, no cualquier click. Para ello podemos poner el punto de ruptura en la línea de código pertinente.



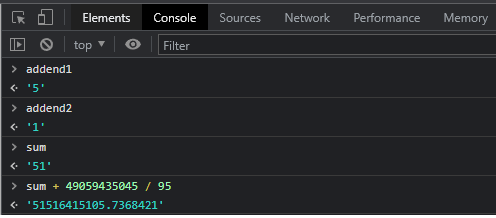
Para analizar paso a paso las líneas de código que se ejecutan tras realizar la acción tenemos los siguientes botones.

C:\Users\USUARIO\Desktop\DAW\D.WEB. ENTORNO CLIENTE\TEMA 1\TAREA\UT02\botones debug.png

Con ellos podemos inspeccionar todos los pasos (incluido los valores que toman las variables involucradas) que se realizan hasta que la función completa su función.



También es útil utilizar la pestaña consola para verificar las variables y realizar operaciones.



En resumen, es indispensable saber utilizar esta herramienta (**debug**) para diagnosticar los problemas que nos aparezcan en nuestro código.