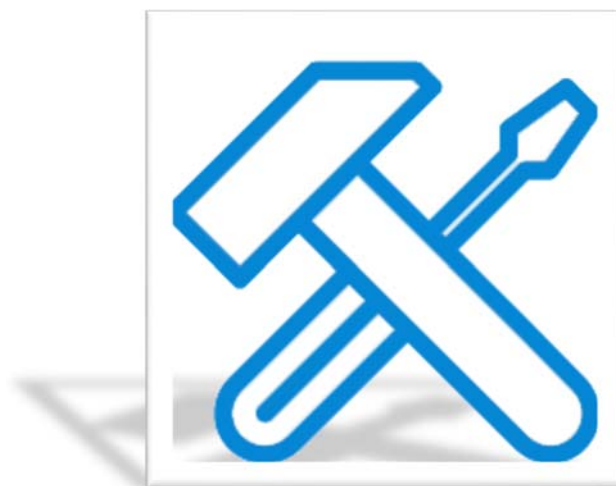




CONOCIENDO LAS HERRAMIENTAS PARA DESARROLLADORES DE LOS NAVEGADORES

UT1-T02



AYMEDIACOÁN MAULEÓN QUINTANA
21/9/2017
DESARROLLO WEB EN ENTORNO CLIENTE

Índice

1.	Enunciado	1
1.1	Parte 1	1
1.2	Parte 2	1
2.	Objetivo	2
3	Herramienta de consola	3
3.1	Herramienta consola.....	3
3.1.1	Navegador Chrome.....	3
3.1.2	Navegador Firefox	3
3.1.3	Navegador Edge.....	4
3.2	Estado de la consola	4
3.2.1	Navegador Chrome.....	4
3.2.1.1	Ejecución de programa	4
3.2.1.2	Resolución del problema.....	5
3.2.2	Navegador Firefox	6
3.2.2.1	Ejecución de programa	6
3.2.2.2	Resolución del problema.....	7
3.2.3	Navegador Edge.....	7
3.2.3.1	Ejecución de programa	7
3.2.3.2	Resolución del problema.....	8
3.3	Conclusión.....	8

1. Enunciado

1.1 Parte 1

Utilizando los navegadores Firefox, Chrome y Edge, localizar y abrir la herramienta Consola. Dichas consolas probablemente mostrarán muchos avisos relativos a todos los errores que ha ido encontrando el navegador en las páginas que hayas ido visitando desde que lo iniciaste.

Utiliza el botón "Limpiar" para borrar dichos errores y mensajes de aviso; a continuación, recarga alguna de las páginas que estés visitando y comprueba qué nuevos errores han aparecido en la consola de errores. Filtra para mostrar únicamente los errores.

1.2 Parte 2

Aquí tienen un programa sencillo que calcula el exponencial de un número.

1) Ejecútala en el navegador Chrome. Proporciona el dígito que te solicita. ¿Que ha ido mal?

Establece un punto de ruptura en el código de la función, recarga la web y vuelve a introducir los dígitos. Monitoriza el valor de las variables a y n del formulario y utiliza la opción adecuada para trazar el programa y ver cómo varían los valores para detectar donde se ha producido el error. Puedes ayudarte del siguiente documento para poder realizar la traza en el navegador Chrome

Ten en cuenta que a medida que se pasa por las llamadas a las funciones recursivas, el subpanel "Call Stack" de la derecha te permite realizar un seguimiento de la pila de llamadas, junto con los parámetros que se pasan en cada llamada a la función.

2) ¿Podrías solucionar el problema que presenta el código que se te ha proporcionado?

3) Repetir la actividad en los navegadores Firefox y Edge. ¿Qué herramienta prefieres? Razona la respuesta, evaluando pros y contras de cada una de ellas.

Una vez realizada la tarea elaborarás un único documento en formato PDF donde figuren las respuestas correspondientes (1, 2, 3). El envío se realizará a través de la plataforma de la forma establecida para ello.

2. Objetivo


El objetivo de la práctica familiarizarse con la herramienta de desarrollador.

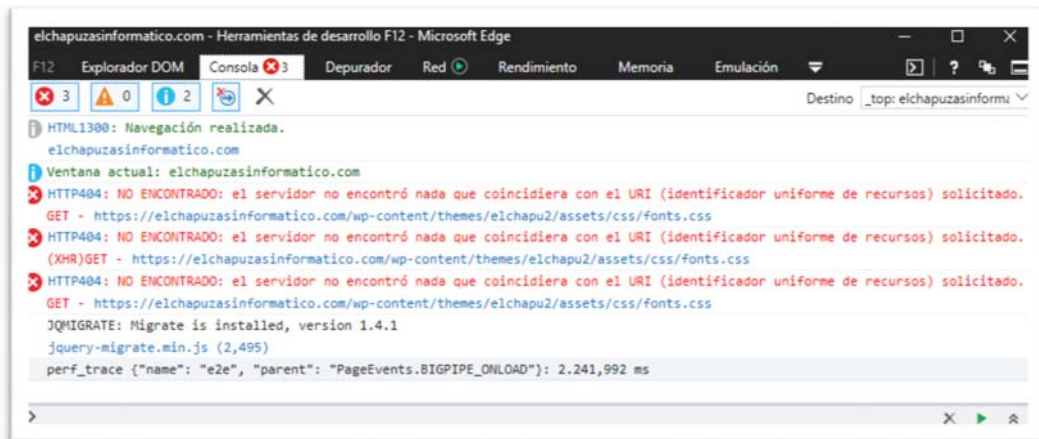
Se trata de comparar la herramienta de desarrollador de los navegadores más populares; Firefox, Chrome y Edge. Exponiendo como conclusión las ventajas y desventajas de realizar la depuración de un código en JavaScript con cada uno de ellos.


Actualizamos la web y filtramos para ver sólo los errores de JavaScript clicando en el botón tal y como se muestra en la imagen.

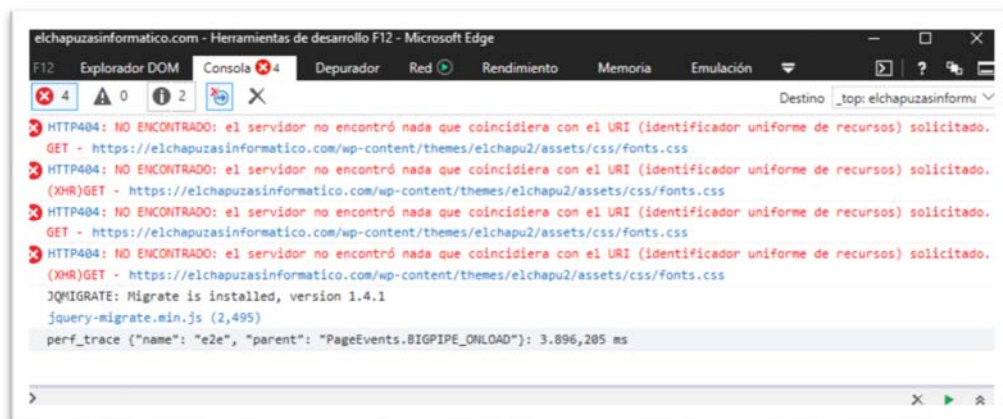


3.1.3 Navegador Edge

Pulsamos f12 para entrar a la herramienta, pulsamos en el icono 



Actualizamos la web y filtramos para ver sólo los errores dejando únicamente activo el icono 

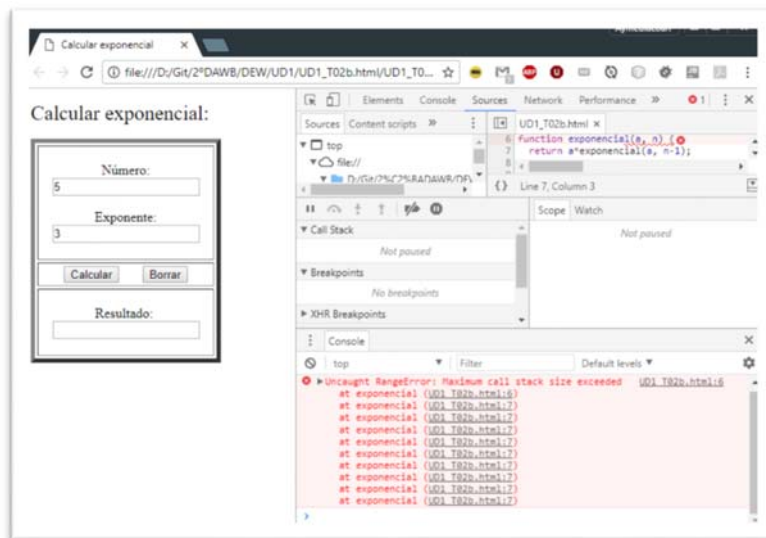



3.2 Estado de la consola

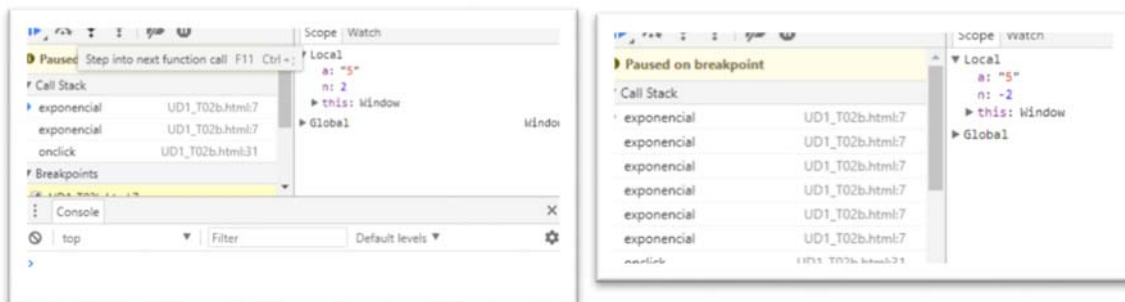
3.2.1 Navegador Chrome

3.2.1.1 Ejecución de programa

Comenzamos la ejecución del programa introduciendo los números y pulsando calcular, vemos que nos devuelve un error llamado “Uncaught RangeError: Maximum call stack size exceeded” del cual deducimos que ha hecho más llamadas de las debidas y la pila se ha desbordado.



Colocamos un punto de interrupción en la función para poder realizar la traza paso a paso (con la tecla F11 o clicando en el icono ) y analizar donde está el error.



En las capturas anteriores mostramos como cambian los valores de las variables en la llamada a la función en el Call Stack. En la segunda imagen podemos observar que la variable “n” adopta valores negativos cuando debería parar en 0.

Continuamos con la traza y se detecta que continua obteniendo valores negativos indefinidamente, hasta que se desborda la pila. Por lo cual, deberemos controlar cuando debe finalizar el bucle.

3.2.1.2 Resolución del problema

Para solucionar el problema detectado anteriormente primero controlamos que el exponente no sea menor que cero. Cuando el exponente es mayor que cero se realiza la operación de exponencial la cual deja de ejecutarse cuando “n==0”, rompiendo así el bucle de llamada a la función evitando que se desborde la pila.


```

> function exponencial(a,n){
  if(n<0){
    alert("El exponente no debe ser menor a 0");
  }
  else{
    if(n==0){
      return 1;
    }
    else{
      return a*exponencial(a,n-1)
    }
  }
}

```

Calcular exponencial:

Número:
5

Exponente:
3

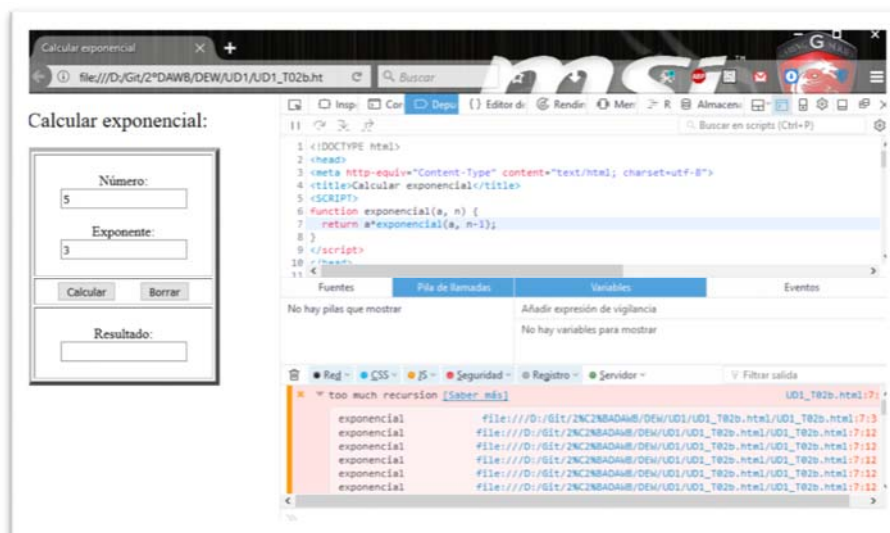
Calcular Borrar


Resultado:
125

3.2.2 Navegador Firefox

3.2.2.1 Ejecución de programa

Comenzamos la ejecución del programa introduciendo los números y pulsando calcular, vemos que nos devuelve un error llamado “too much recursion” del cual deducimos que ha hecho más llamadas de las debidas y la pila se ha desbordado.



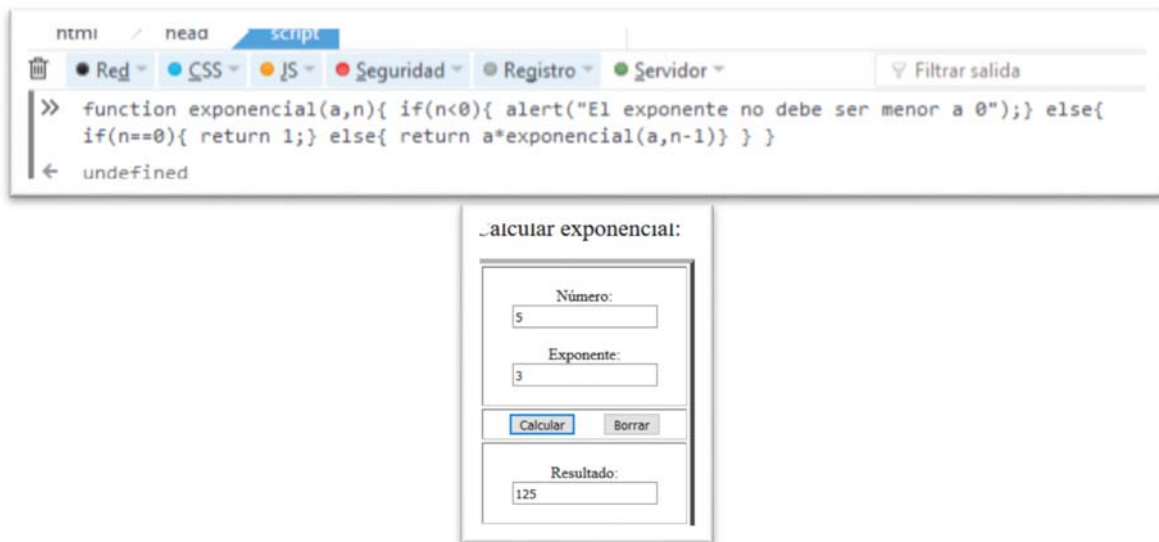
Colocamos un punto de interrupción en la función para poder realizar la traza paso a paso (con la tecla F11 o clicando en el icono ) y analizar donde está el error.



En las capturas anteriores mostramos como cambian los valores de las variables en la llamada a la función en la Pila de llamadas. En la segunda imagen podemos observar que la variable “n” adopta valores negativos cuando debería parar en 0.

Continuamos con la traza y se detecta que continua obteniendo valores negativos indefinidamente, hasta que se desborda la pila. Por lo cual, deberemos controlar cuando debe finalizar el bucle.

3.2.2.2 Resolución del problema

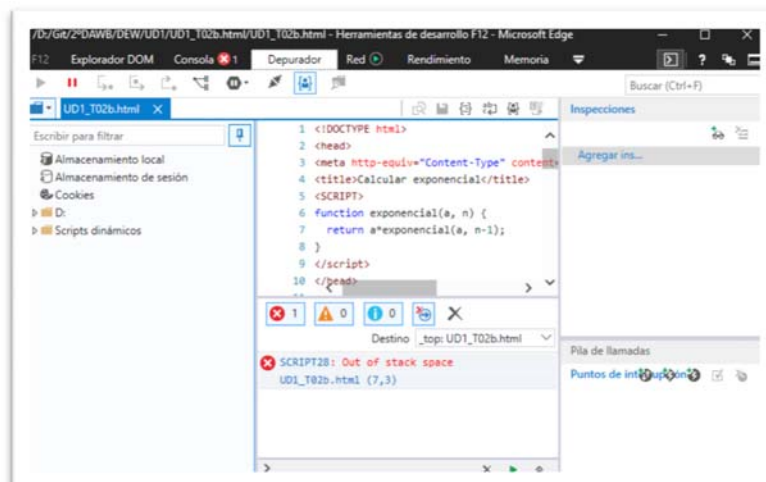



Para solucionar el problema detectado anteriormente primero controlamos que el exponente no sea menor que cero. Cuando el exponente es mayor que cero se realiza la operación de exponencial la cual deja de ejecutarse cuando “n=0”, rompiendo así el bucle de llamada a la función evitando que se desborde la pila.

3.2.3 Navegador Edge

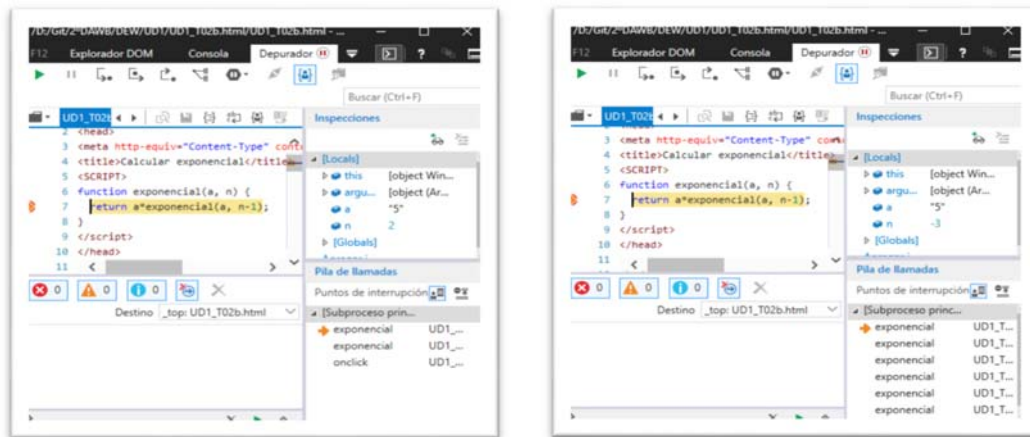
3.2.3.1 Ejecución de programa

Comenzamos la ejecución del programa introduciendo los números y pulsando calcular, vemos que nos devuelve un error llamado “Out of stack space” del cual deducimos que ha hecho más llamadas de las debidas y la pila se ha desbordado.



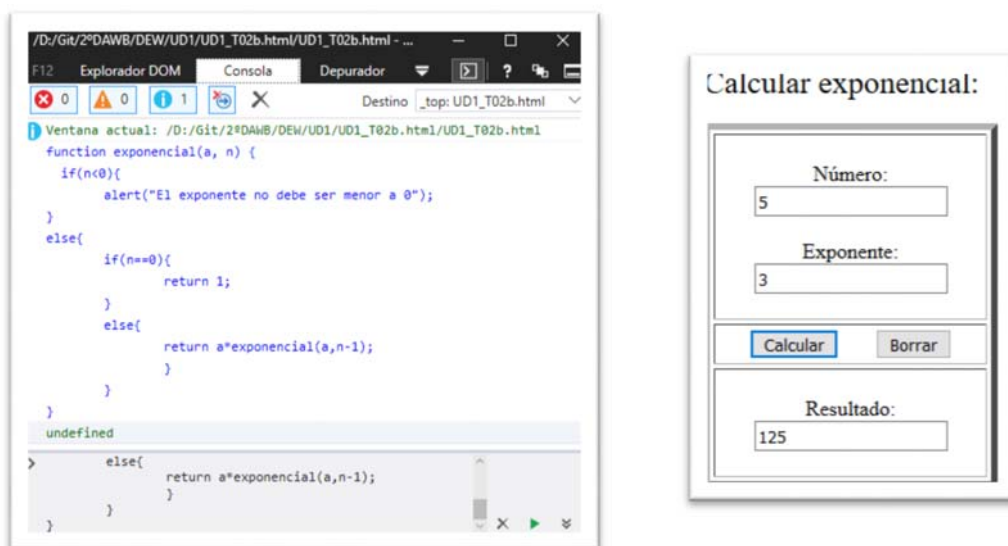
Colocamos un punto de interrupción en la función para poder realizar la traza paso a paso (con la tecla F11 o clicando en el icono ) y analizar donde está el error.

En las capturas anteriores mostramos como cambian los valores de las variables en la llamada a la función en la Pila de Llamadas. En la segunda imagen podemos observar que la variable “n” adopta valores negativos cuando debería parar en 0.



Continuamos con la traza y se detecta que continua obteniendo valores negativos indefinidamente, hasta que se desborda la pila. Por lo cual, deberemos controlar cuando debe finalizar el bucle.

3.2.3.2 Resolución del problema



Para solucionar el problema detectado anteriormente primero controlamos que el exponente no sea menor que cero. Cuando el exponente es mayor que cero se realiza la operación de exponencial la cual deja de ejecutarse cuando “n==0”, rompiendo así el bucle de llamada a la función evitando que se desborde la pila.

3.3 Conclusión

Basándonos en su rapidez y fiabilidad considero que el Chrome cumple mejor con estos requisitos, en cuanto a diseño y menús, el Edge está mejor distribuido.

Ahora bien, en cuanto al rendimiento, mi opinión, es que el Edge es mucho más lento y se bloquea con facilidad, incluso con programas tan sencillos como este.

En cuanto a Firefox, está bastante equilibrado, lo único a destacar es la forma en la que la consola muestra el código, ya que, no lo muestra tabulado, lo que hace más difícil la comprensión del código a primera vista.