

DevTools

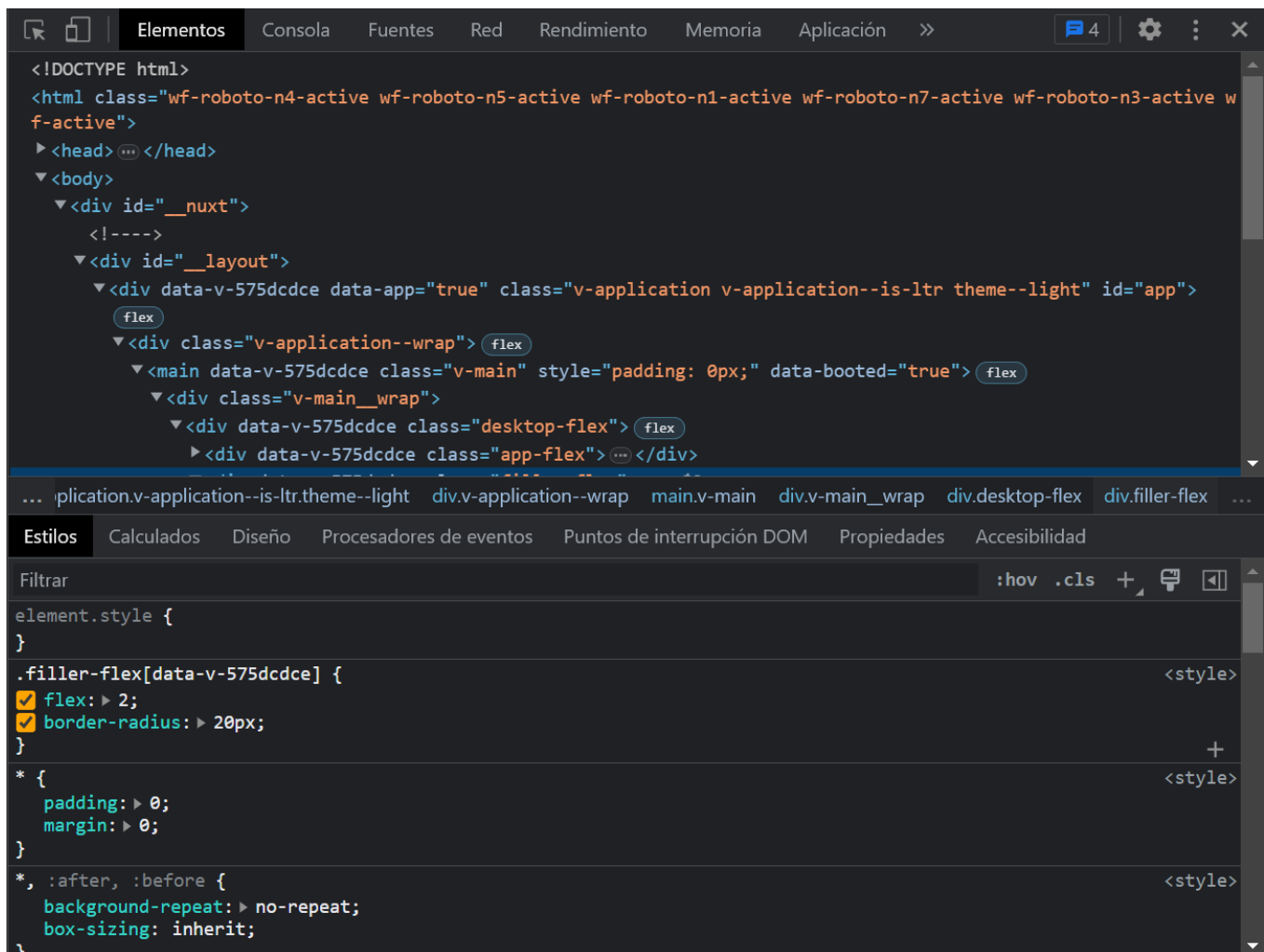
¿Qué son las DevTools?

Las DevTools, o herramientas de desarrollador, son un conjunto de herramientas integradas en la mayoría de los navegadores web modernos que permiten a los desarrolladores web analizar y depurar sus aplicaciones web.

¿Cuáles son las DevTools?

Las herramientas de desarrollador que se encuentran en los navegadores son:

- **Inspector de elementos:** una herramienta que permite a los desarrolladores inspeccionar/analizar y modificar el código HTML, CSS y JavaScript de una página web.



- **Consola:** una herramienta que permite a los desarrolladores ejecutar código JavaScript en tiempo real y ver los resultados en la consola, se utiliza comúnmente con la función **console.log()** durante el desarrollo de webs y apps, para ver información en consola.

```

> let reloj = ()=>{
  setInterval(()=>{
    let hora = new Date();
    console.log(hora.toLocaleTimeString());
  },1000)
}
< undefined
> reloj()
< undefined
4:06:22 VM3953:4
4:06:23 VM3953:4
4:06:24 VM3953:4
4:06:25 VM3953:4
4:06:26 VM3953:4
4:06:27 VM3953:4
4:06:28 VM3953:4
4:06:29 VM3953:4
4:06:30 VM3953:4
4:06:31 VM3953:4
4:06:32 VM3953:4
>
  
```

Sin embargo, existen otras funciones de gran utilidad, como:

- **console.table():** Imprime una tabla en la consola. Es muy útil para mostrar datos en formato de tabla. Ejemplo:

```

> let datos = [{nombre: "Juan", edad: 28},
  {nombre: "María", edad: 35},
  {nombre: "Pedro", edad: 20},
  {nombre: "Juan", edad: 28},
  {nombre: "María", edad: 35},
  {nombre: "Pedro", edad: 20}];

console.table(datos);
VM533:8
  
```

(índice)	nombre	edad
0	'Juan'	28
1	'María'	35
2	'Pedro'	20
3	'Juan'	28
4	'María'	35
5	'Pedro'	20

```

  ▶ Array(6)
  < undefined
  >
  
```

- **console.error():** Imprime un mensaje de error en la consola. Es muy útil para detectar errores en el código.

```
> console.error("El proceso no se ejecuto correctamente")
✖ ▶ El proceso no se ejecuto correctamente
< undefined
>
```

- **console.warn():** Imprime una advertencia en la consola. Es muy útil para señalar posibles problemas en el código.

```
> console.warn("El proceso no se ejecutara correctamente")
⚠ ▶ El proceso no se ejecutara correctamente
< undefined
> |
```

- **console.clear():** Borra la consola. Es útil para eliminar mensajes antiguos y hacer que la consola sea más fácil de leer.

```
Se han borrado los datos de la consola
< undefined
> console.clear()
```

- **console.time():** Se utiliza para medir el tiempo que tarda una sección de código en ejecutarse.

```
> function calcular() {  
  console.time('tiempo de ejecucion');  
  
  let resultado = 0;  
  for (let i = 0; i < 1000000; i++) {  
    resultado ++;  
  }  
  
  console.timeEnd('tiempo de ejecucion');  
  console.log("Variable resultado: " + resultado);  
}  
  
calcular();
```

tiempo de ejecucion: 3.06005859375 ms

[VM2705:9](#)

Variable resultado: 1000000

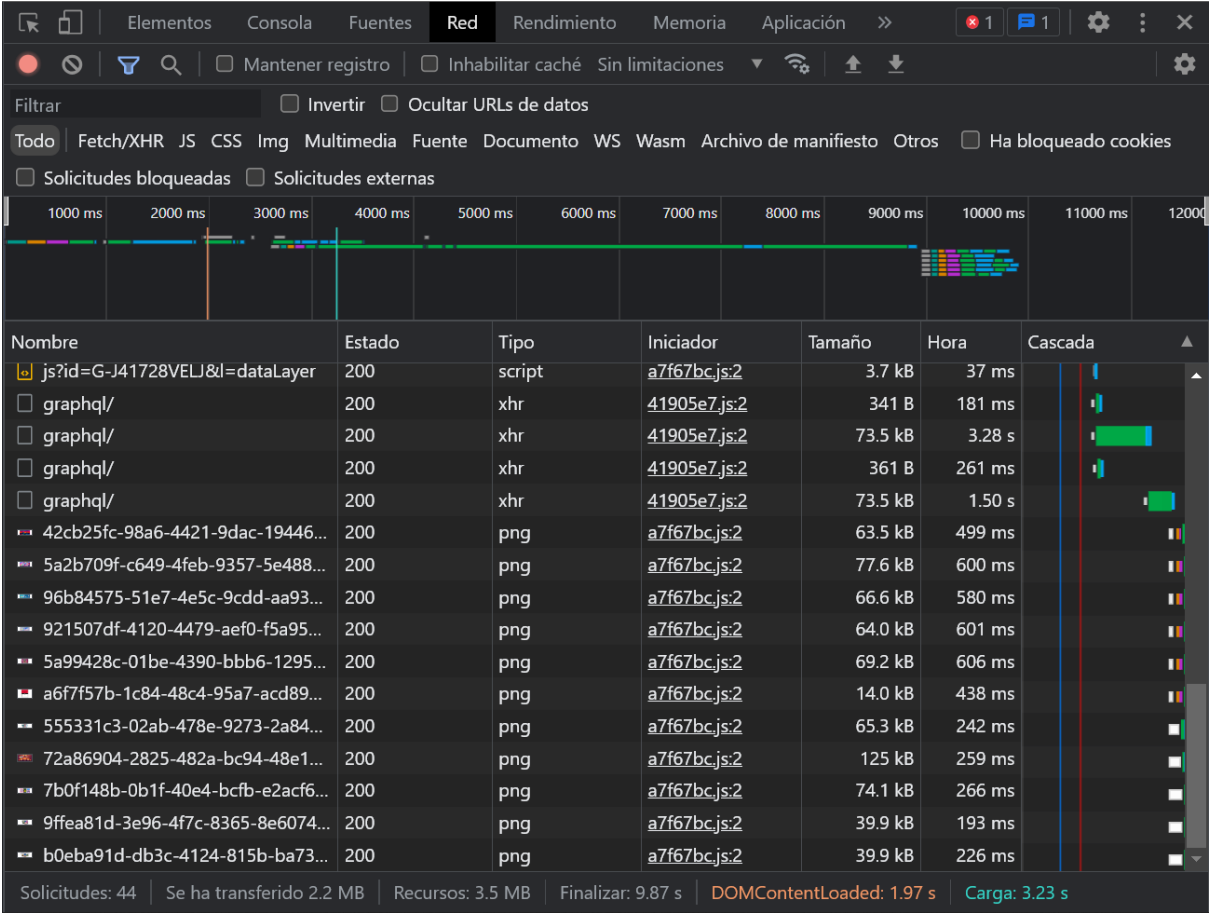
[VM2705:10](#)

◀ undefined

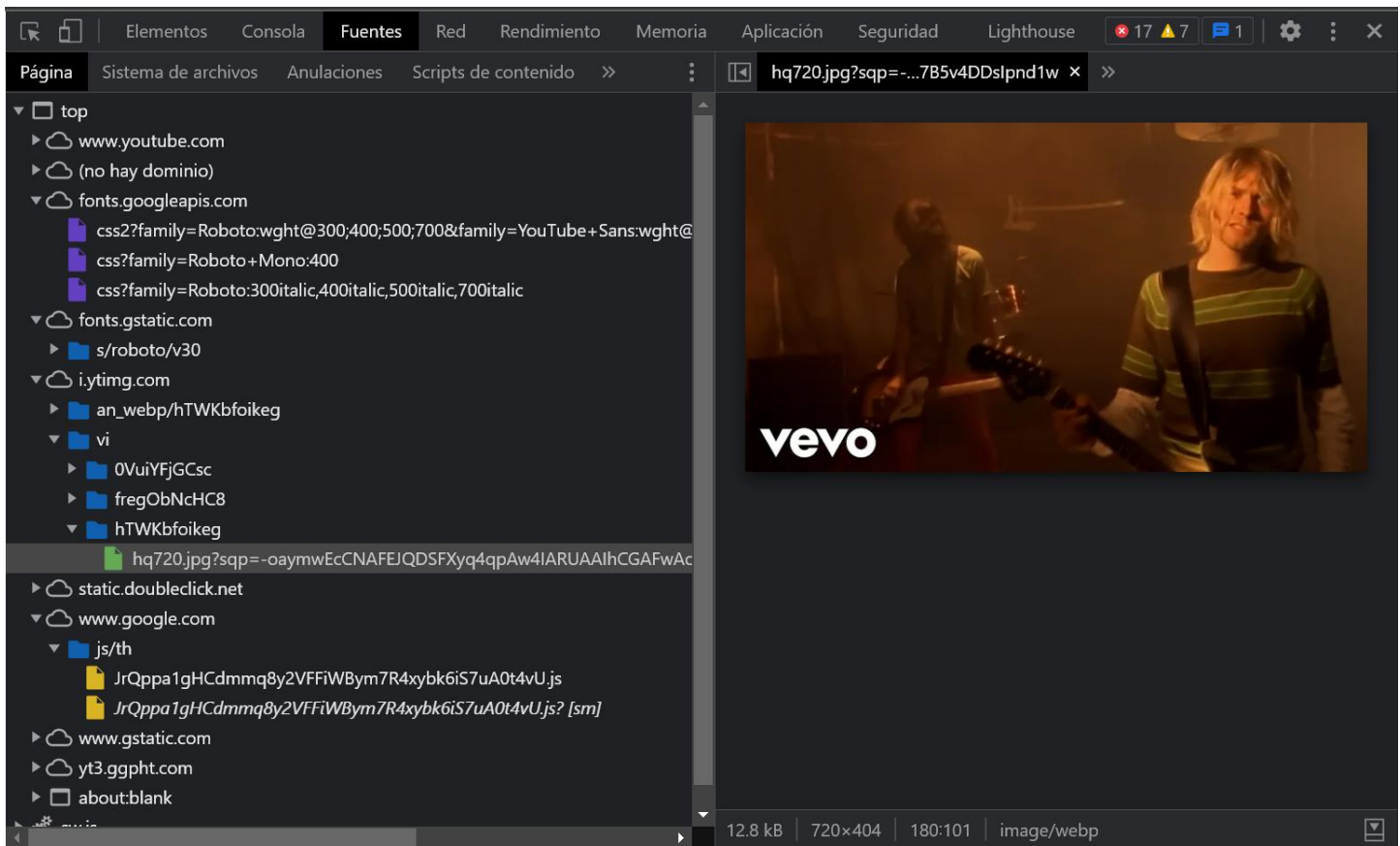
> |

Estas son solo algunas funciones de consola disponibles en los navegadores web modernos. Los desarrolladores utilizan estas funciones para interactuar con el DOM de una página web, depurar el código y mejorar la experiencia del usuario.

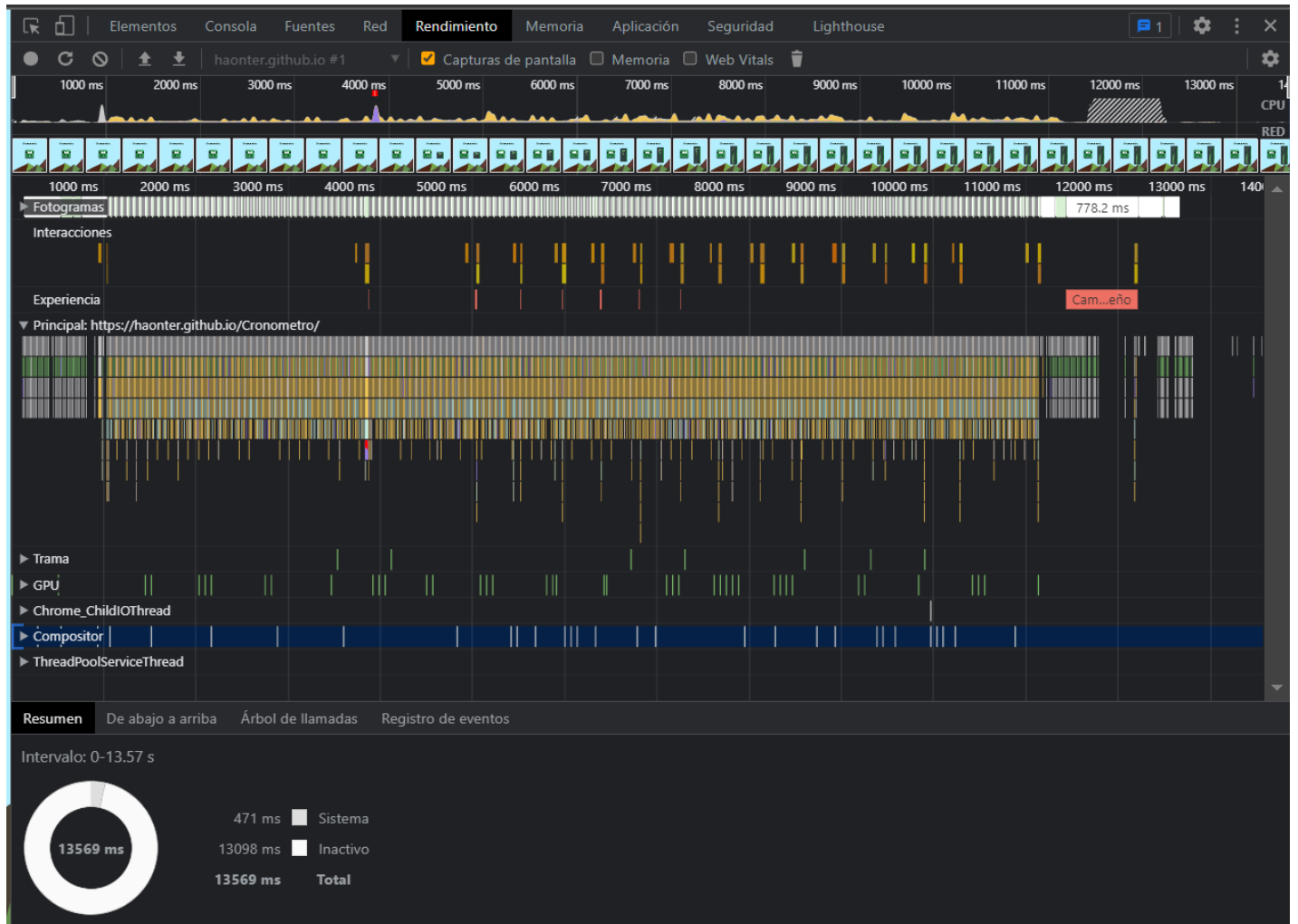
- **Red/Network:** Es una herramienta que permite a los desarrolladores ver y analizar las solicitudes de red que se realizan al cargar una página web, y durante el tiempo que permanezca abierta, Se puede observar desde el tiempo de carga hasta los detalles de las solicitudes, se utiliza comúnmente para detectar fallos en la carga de elementos o durante la optimización de tiempo de carga de una web.



- **Fuentes/Sources:** Es una herramienta que permite a los desarrolladores ver los archivos que componen un sitio o página, con la finalidad de depurar.



- Rendimiento/Performance:** una herramienta que permite a los desarrolladores medir el rendimiento de su aplicación web y analizar la causa de cualquier retraso, esta herramienta permite capturar el comportamiento de un sitio durante un tiempo estimado en el que se realizaran las acciones a evaluar, para luego al finalizar la captura, obtener un análisis detallado de todo lo ocurrido, de esta manera podemos detectar el comportamiento de un sitio o aplicación web.



¿Como se accede a ellas?

Para abrir las herramientas de desarrollador, normalmente se puede hacer clic con el botón derecho en cualquier parte de la página y seleccionar "Inspeccionar" o presionando F12 en el teclado. Cada herramienta de desarrollador tiene su propia interfaz y conjunto de opciones, pero en general, la mayoría de las herramientas son fáciles de usar y con un poco de práctica pueden ayudar a los desarrolladores a mejorar significativamente su flujo de trabajo y eficiencia.