JavaScript Debugging

Sergio Tabares Hernández

Adrián Epifanio Rodríguez Hernández

Index

1. What is a debugger?

- Program used to test and run the target program under controlled conditions that permit the programmer monitoring the changes while it runs that may indicate malfunctioning code.

Visual Studio Code

1.1 Syntax or type errors

- These are always caught by the compiler, and reported via error messages. Typically, an error message clearly indicates the cause of error; for instance, the line number, the incorrect piece of code, and an explanation.

```
1  // Syntax error
2  for (let i = 0; < 10; i++) {
3    console.log(i);
4  }
5
6  // Type error
7  const test = 1;
8  test = 3;
9
```

1.2 Typos and other simple errors

- That have pass undetected by the type-checker or the other checks in the compiler. Once these are identified, they can easily be fixed. Passing parameters in incorrect order, or using the wrong element order in tuples.

  `x + y * z` instead of `(x + y) * z;`

1.3 Reference errors

- Represents an **error** when a non-existent variable is referenced.

```
10  // Reference error
11  for(let i = 0; i < 10; i++) {
12      console.log(non_existing_variable);
13  }
14
```

1.4 Implementarion and logical errors

- It may be the case that logic in the high-level algorithm of a program is correct, but some low-level, concrete data structures are being manipulated incorrectly, breaking some internal representation invariants. If the algorithm is logically flawed, the programmer must re-think the algorithm. Fixing such problems is more difficult, especially if the program fails on just a few corner cases.

```
// Logical error in function n ^ n
  function wrongRaisedTo(number) {
    let result = 1;
    for(let counter = 0; counter <= number; counter++) {
      result *= number;
    }
    return result;
  }
  let number = wrongRaisedTo(3);
  console.log(number);
```

2. Strategies

1. Incremental and bottom-up program development
2. Instrument program with assertions
3. Use debuggers
4. Backtracking
5. Binary search
6. Problem simplification
7. Scientific method: form hypotheses
8. Bug clustering

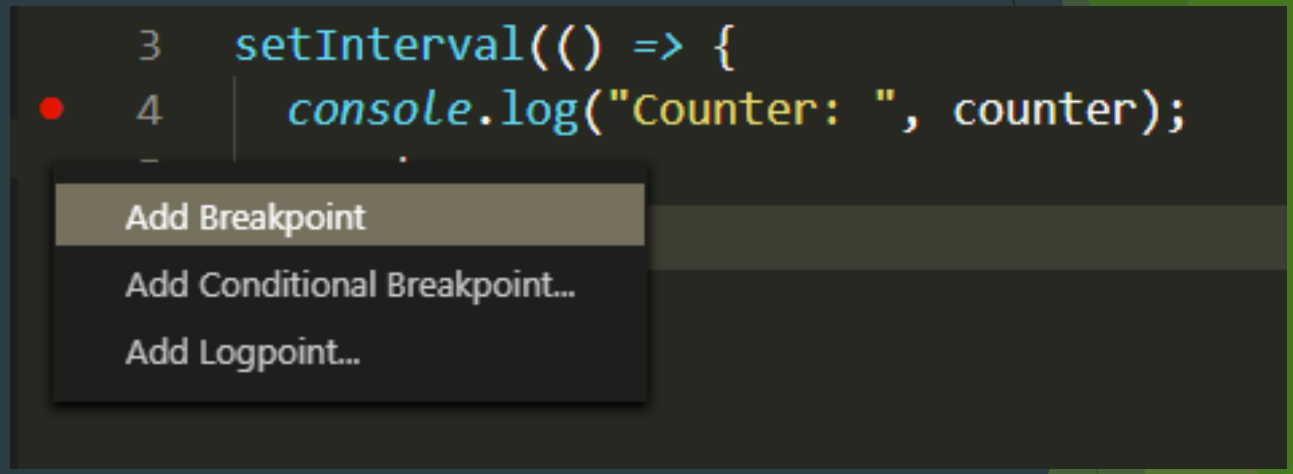
2.1 Strict mode

```
function canYouSpotTheProblem() {
  "use strict";
  for (counter = 0; counter < 10; counter++) {
    console.log("Happy happy");
  }
}

canYouSpotTheProblem();
// → ReferenceError: counter is not defined
```

3. Breakpoints / Logpoints

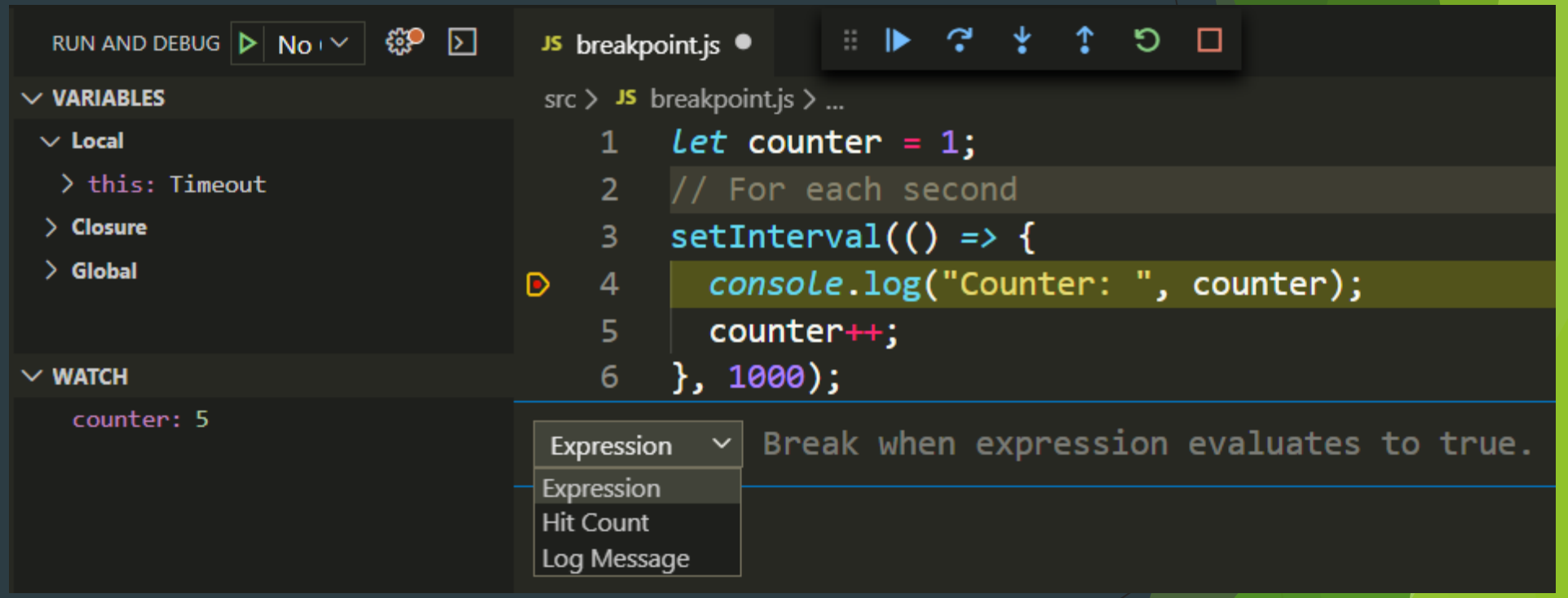
- A line where the program prints a message or stops till the user clicks to continue the execution.



- They are used for controlling the variables while the program is running helping the programmer to find errors in the code.

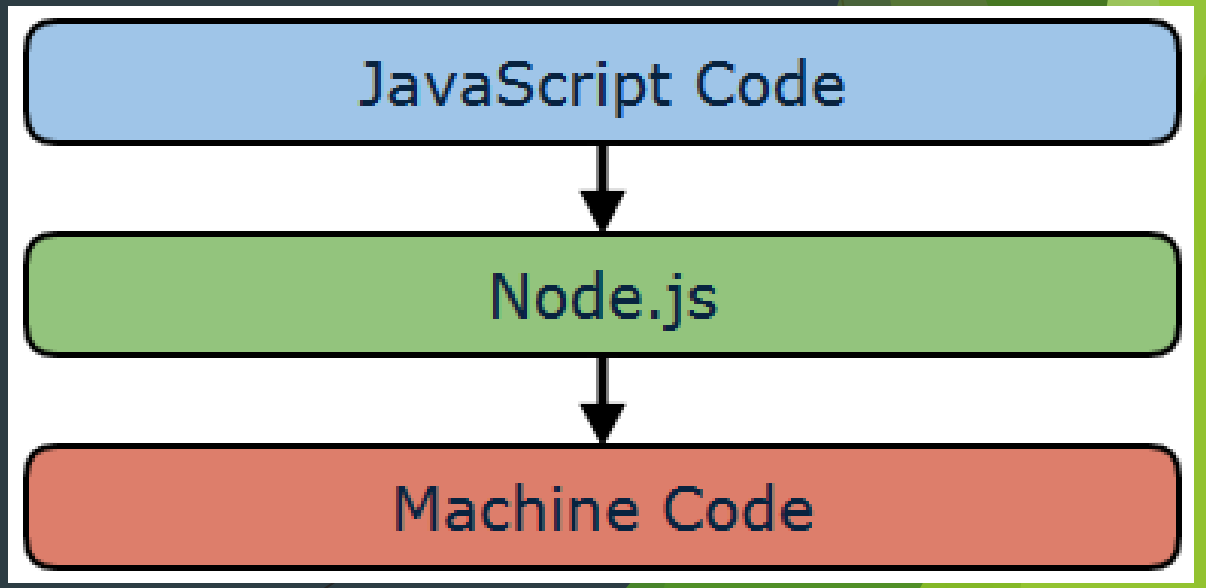
3.1 Breakpoints / Logpoints Types

- Normal
- Conditional
- Hit count
- Log message



4. Node.js

- Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser.Node.js lets developers use JavaScript to write command line tools and for running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser



4.1 Stepping

- cont, c: Continue execution
- next, n: Step next
- step, s: Step in
- out, o: Step out
- pause: Pause running code

4.2 Execution control

- Run
- Restart
- Kill

```
debug> restart
< Debugger listening on ws://127.0.0.1:9229/8ac5a686-b062-4462-a18e-c2489a5c9a24
< For help, see: https://nodejs.org/en/docs/inspector
< Debugger attached.
Warning: script 'file:///home/sergio/Dropbox/Documentos/Curso%202019-2020%20-%203o%20ULL/Programaci%C3%B3n%20de%20Aplicaciones%20Interactivas/Presentaci%C3%B3n/JavaScriptDebugger/src/chrome_test/testfile.js' was not loaded yet.
1 breakpoints restored.
Break on start in testfile.js:52
 50 }
 51
>52 main();
 53
debug> c
break in testfile.js:42
 40   }
 41   do {
>42     result = String(n % base) + result;
 43     n /= base;
 44   } while (n > 0);
debug>
```

4.3 Breakpoints

- setBreakpoint(), sb()
- setBreakpoint(line), sb(line)
- setBreakpoint('fn()'), sb(...)
- setBreakpoint('script.js', 1), sb(...)
- clearBreakpoint('script.js', 1), cb(...)

```
debug> list(10)
 42      result = String(n % base) + result;
 43      n /= base;
 44    } while (n > 0);
 45    return sign + result;
 46 }
 47
 48 function main() {
 49    console.log(numberToString(13, 10));
 50 }
 51
>52 main();
 53
debug> sb(42)
 37    if (n < 0) {
 38      sign = "-";
 39      n = -n;
 40    }
 41    do {
>42      result = String(n % base) + result;
 43      n /= base;
 44    } while (n > 0);
 45    return sign + result;
 46 }
 47
debug>
```

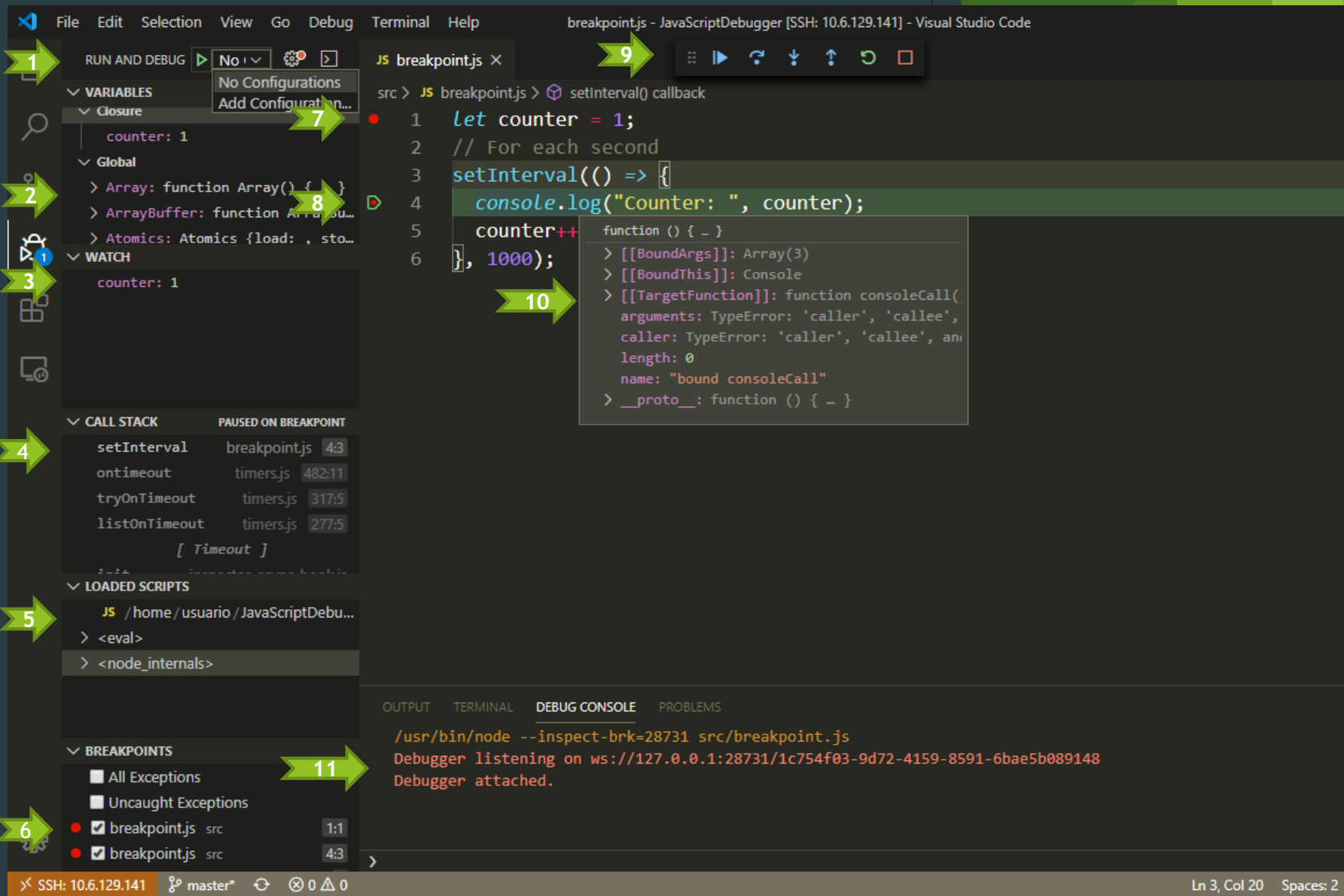
4.4 Information

- backtrace, bt
- list(5)
- watch(expr)
- unwatch(expr)
- watchers
- repl
- exec expr

```
debug> list(5)
 38      sign = "-";
 39      n = -n;
 40    }
 41    do {
*42      result = String(n % base) + result;
>43      n /= base;
 44    } while (n > 0);
 45    return sign + result;
 46 }
 47
 48 function main() {
debug> watch("n /= base")
debug> watchers
  0: n /= base = 1.3
debug>
```

5. Visual Studio Code

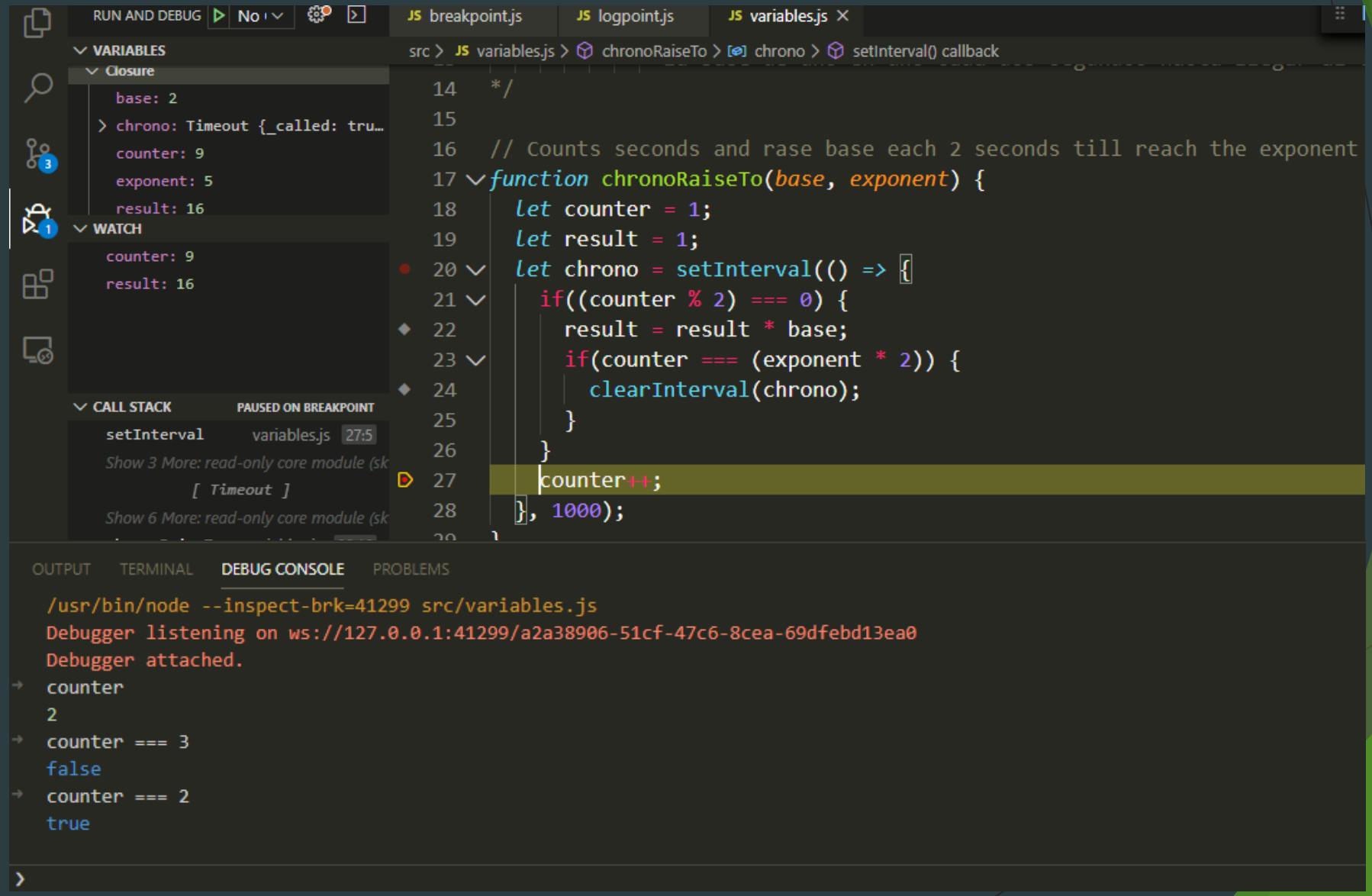
1. Debug selector
2. Navigator
3. Watch expresión
4. Stack traces
5. Debugging scripts
6. Breakpoints list
7. Breakpoint line
8. Program pointer
9. Debug Actions
10. Variable values
11. Debug console



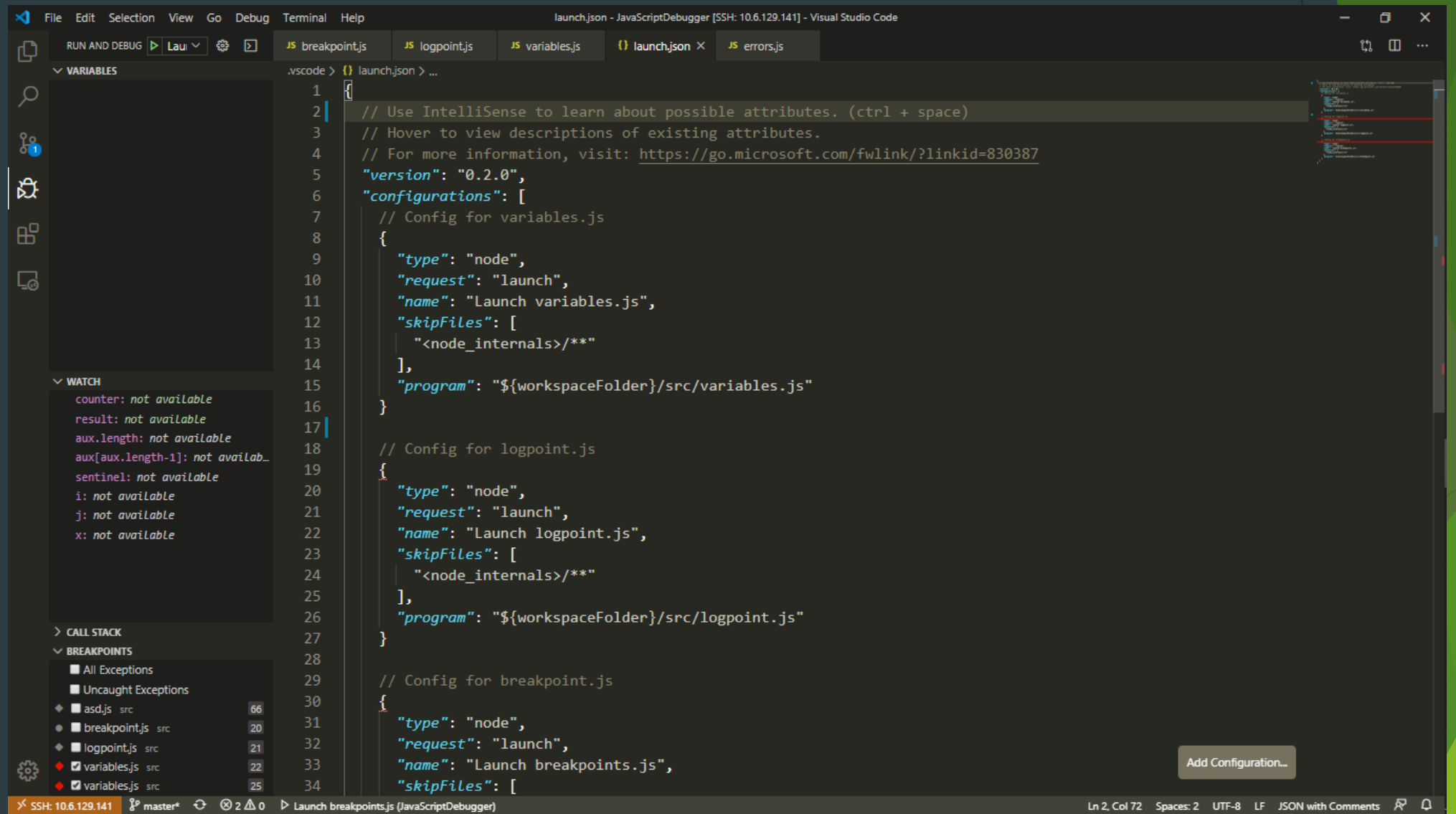
5.1 Keyboard shortcuts

- Continue / Pause F5
- Step Over F10
- Step Into F11
- Step Out Shift + F11
- Restart Ctrl + Shift + F5
- Stop Shift + F5
- Add Breakpoint F9

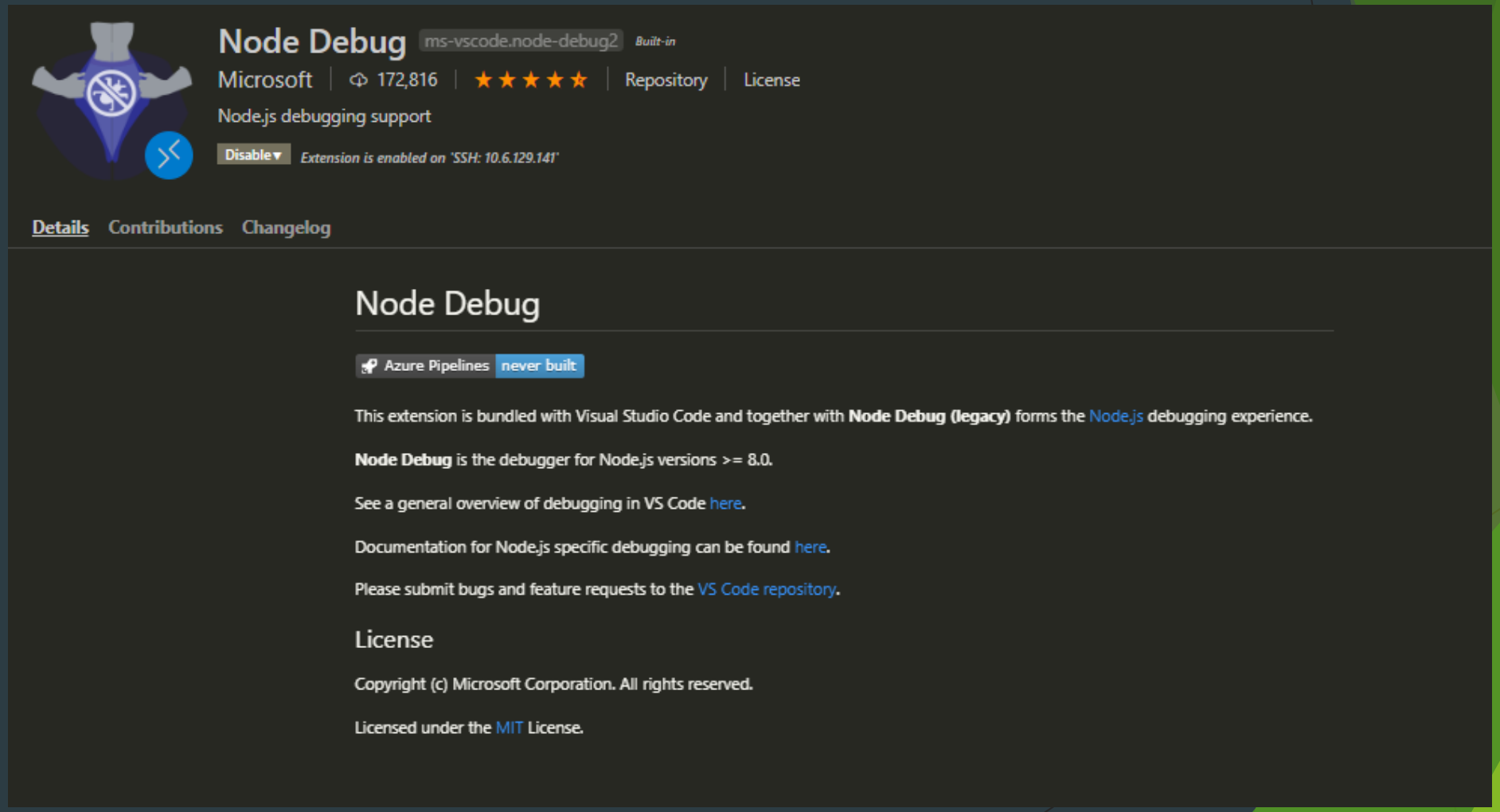
5.2 Variables



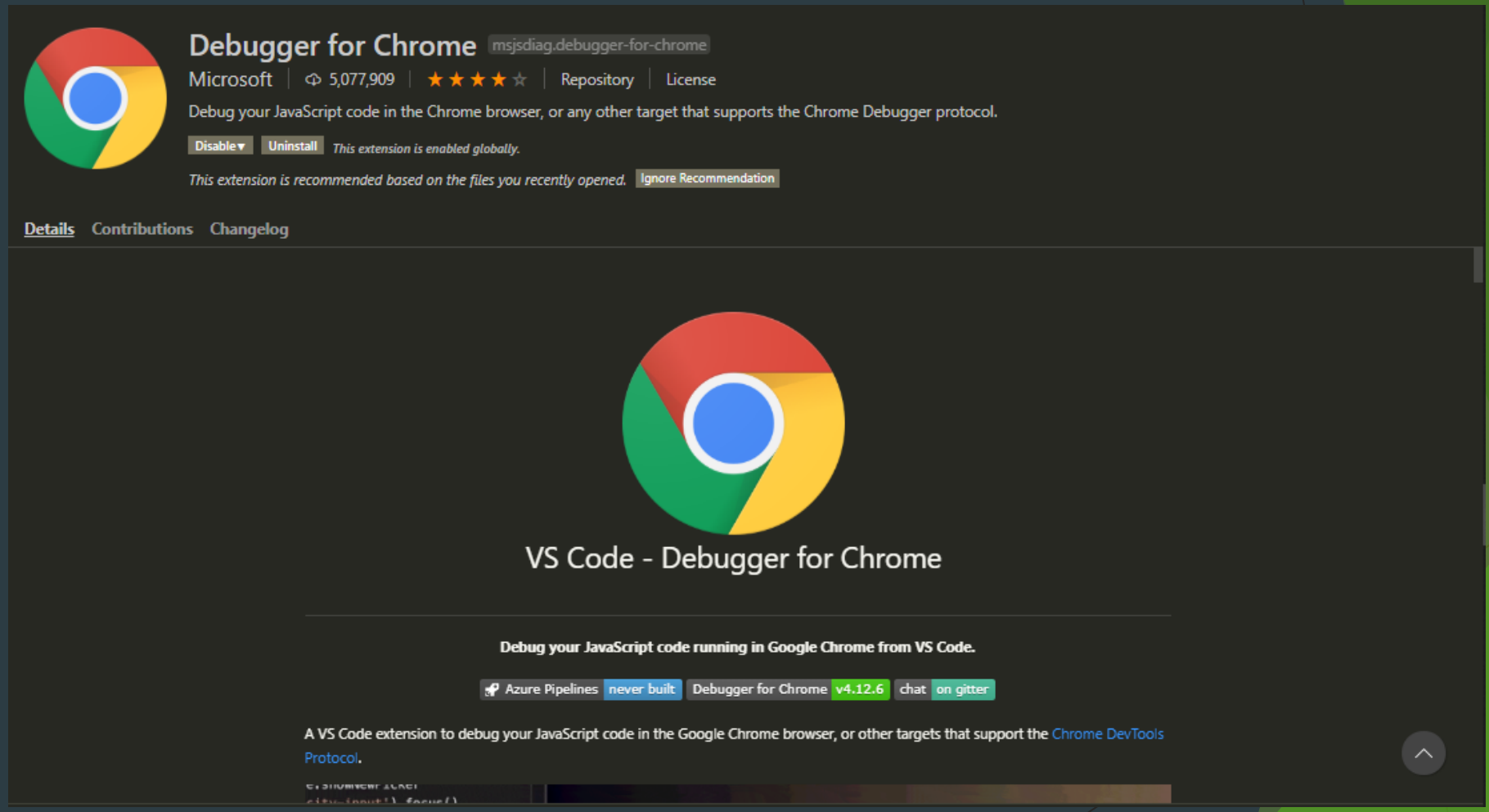
5.3 Launch.json



5.4 Node Debug



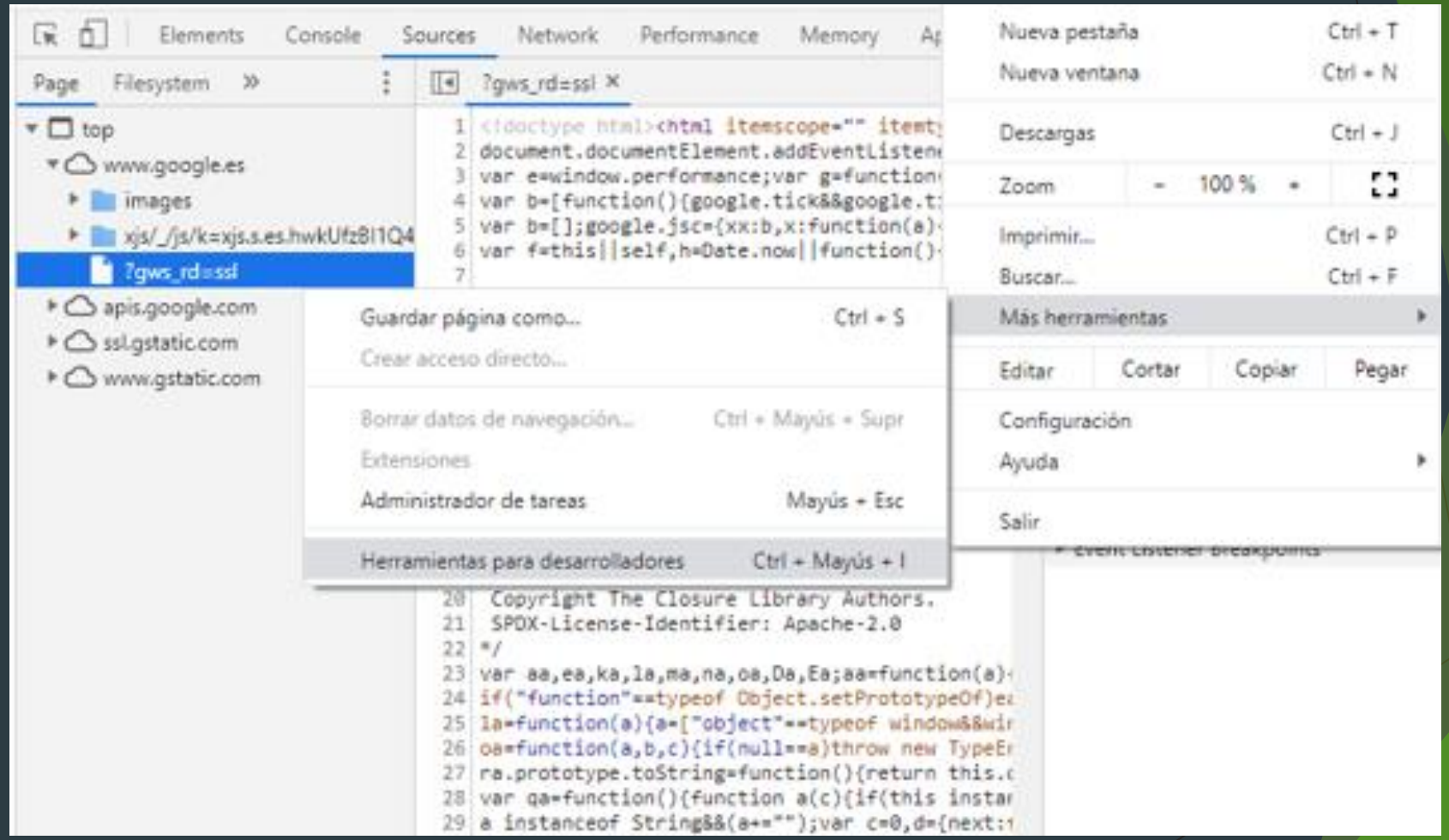
5.5 Debugger for Chrome



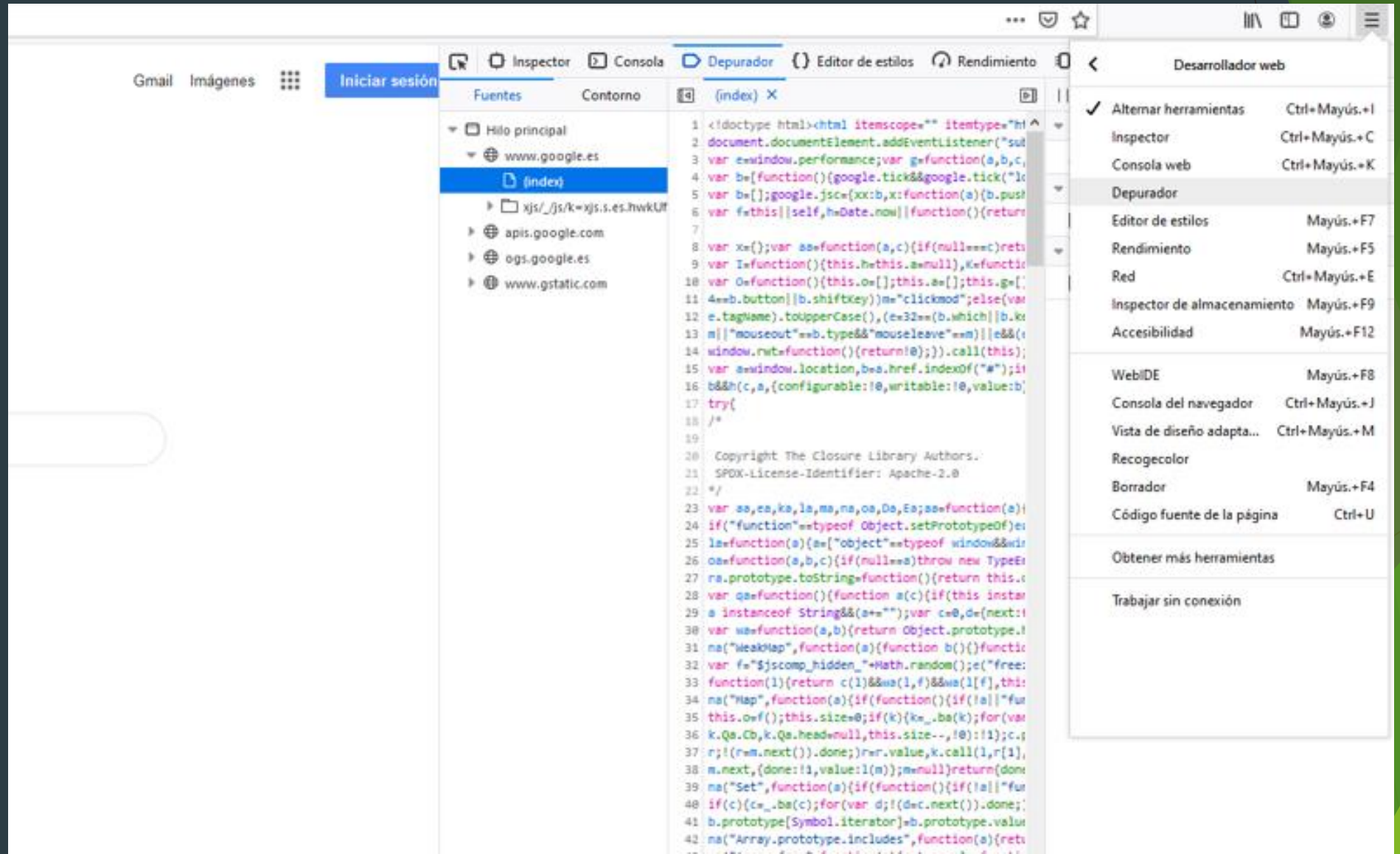
6. Chrome and DevTools

- Chrome DevTools is a set of web developer tools built directly into the Google Chrome browser. DevTools can help you edit pages on-the-fly and diagnose problems quickly, which ultimately helps you build better websites, faster.

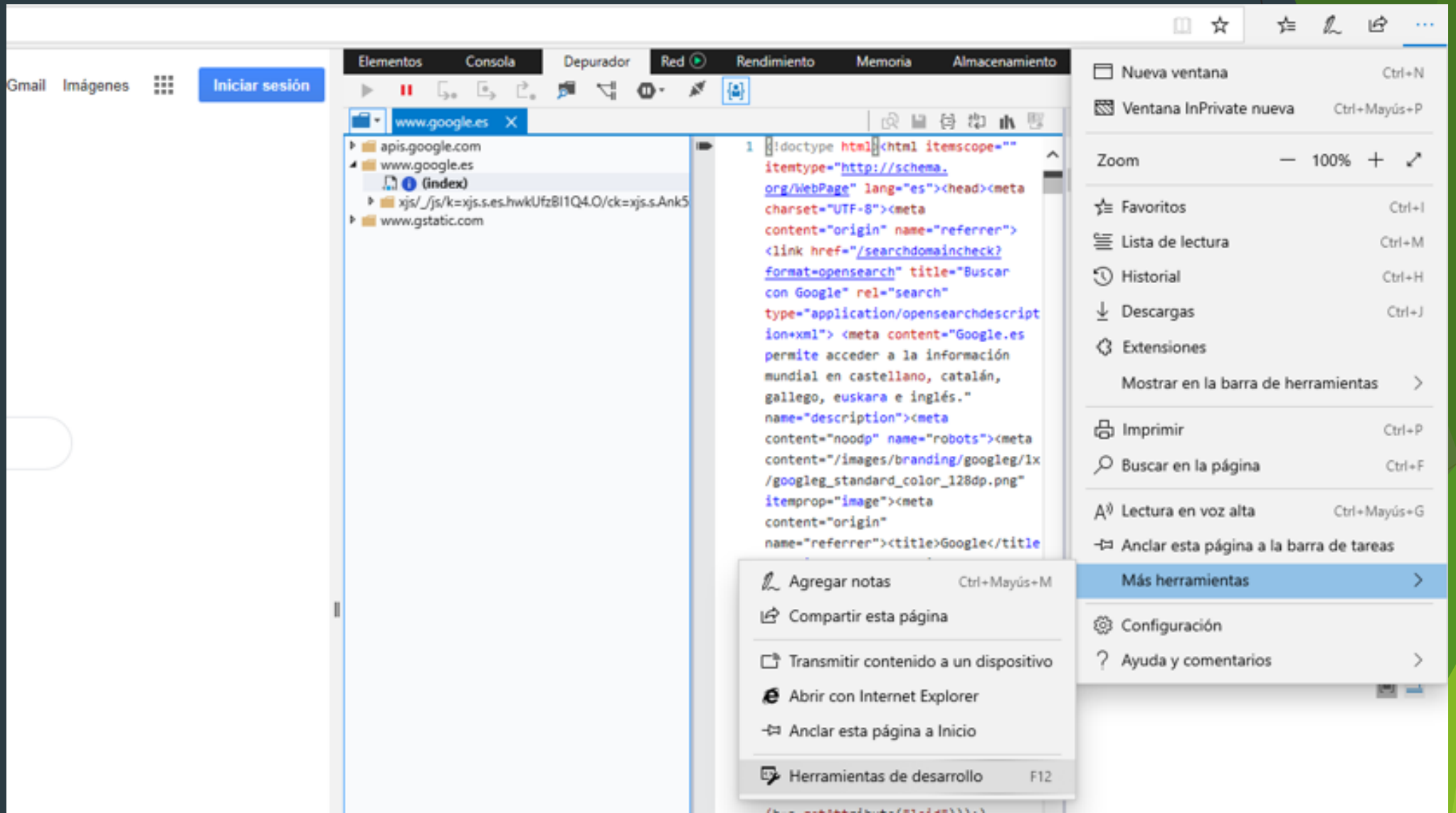
6.1 Chrome and Dev Tools



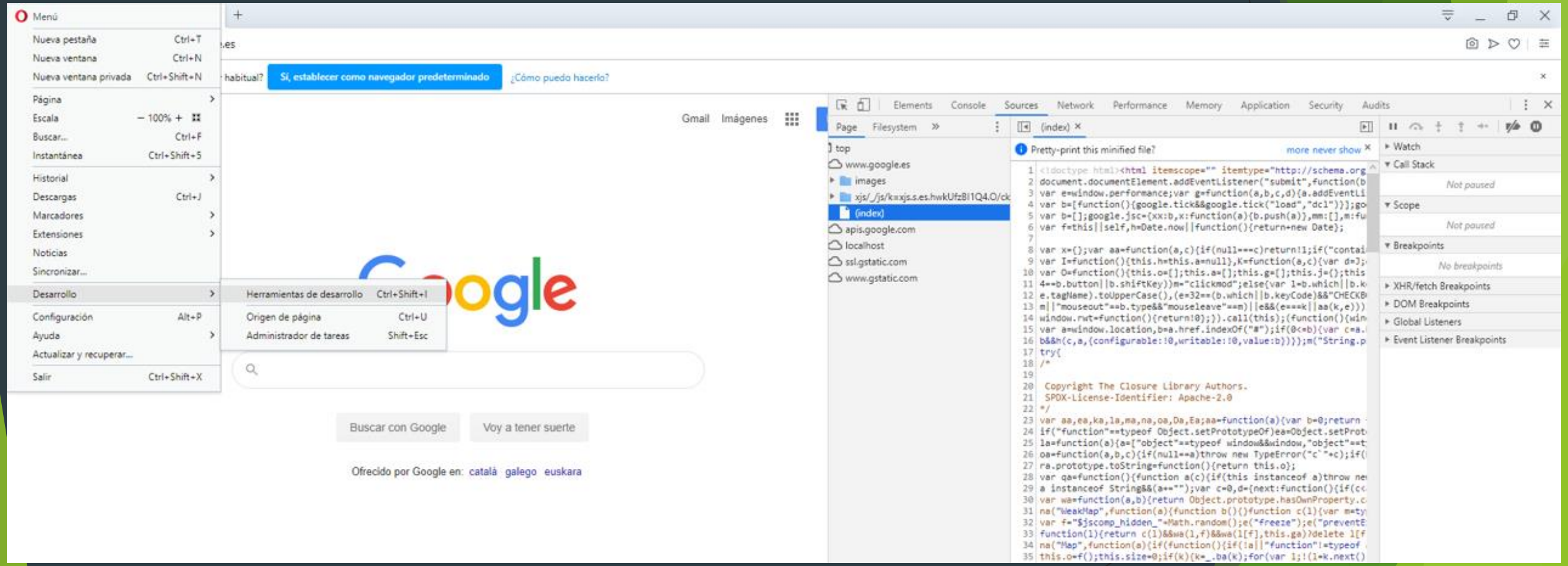
6.2 Firefox


Gmail
Imágenes
Iniciar sesión
Inspector
Consola
Depurador
Editor de estilos
Rendimiento
Fuentes
Contorno
(index)
Hilo principal
www.google.es
(index)
xjs/_/js/k=xjs.s.es.hwkUf
apis.google.com
ogs.google.es
www.gstatic.com
Desarrollador web
Alternar herramientas Ctrl+Mayús.+I
Inspector Ctrl+Mayús.+C
Consola web Ctrl+Mayús.+K
Depurador
Editor de estilos Mayús.+F7
Rendimiento Mayús.+F5
Red Ctrl+Mayús.+E
Inspector de almacenamiento Mayús.+F9
Accesibilidad Mayús.+F12
WebIDE Mayús.+F8
Consola del navegador Ctrl+Mayús.+J
Vista de diseño adapta... Ctrl+Mayús.+M
Recogecolor
Borrador Mayús.+F4
Código fuente de la página Ctrl+U
Obtener más herramientas
Trabajar sin conexión

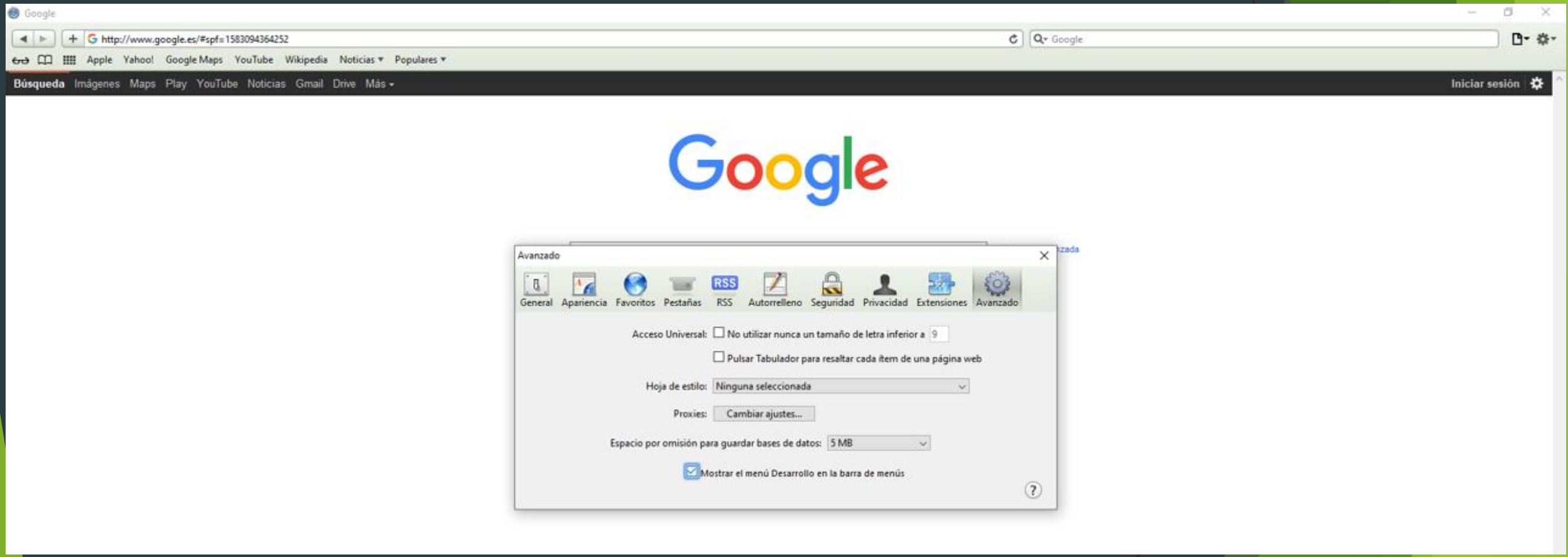
6.3 Microsoft Edge



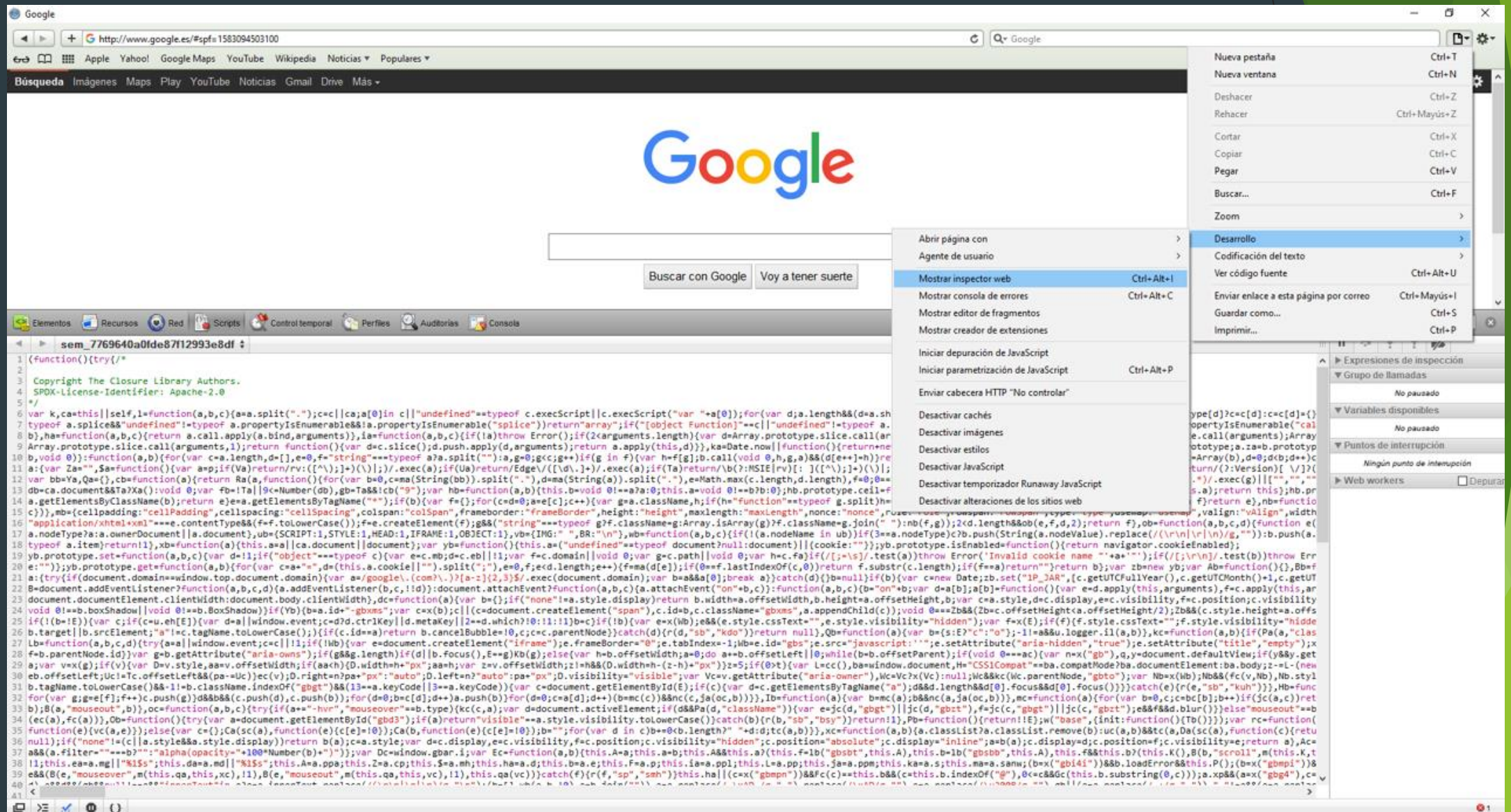
6.4 Opera


Menú
Nueva pestaña Ctrl+T
Nueva ventana Ctrl+N
Nueva ventana privada Ctrl+Shift+N
Página
Escala − 100% +
Buscar... Ctrl+F
Instantánea Ctrl+Shift+5
Historial
Descargas Ctrl+J
Marcadores
Extensiones
Noticias
Sincronizar...
Desarrollo
Configuración Alt+P
Ayuda
Actualizar y recuperar...
Salir Ctrl+Shift+X
Herramientas de desarrollo Ctrl+Shift+I
Origen de página Ctrl+U
Administrador de tareas Shift+Esc
habitual?
Sí, establecer como navegador predeterminado
¿Cómo puedo hacerlo?
Gmail
Imágenes
Buscar con Google
Voy a tener suerte
Ofrecido por Google en: català galego euskara
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
Page
Filesystem
(index)
Pretty-print this minified file?
more never show
top
www.google.es
images
(index)
apis.google.com
localhost
ssl.gstatic.com
www.gstatic.com
Watch
Call Stack
Not paused
Scope
Not paused
Breakpoints
No breakpoints
XHR/fetch Breakpoints
DOM Breakpoints
Global Listeners
Event Listener Breakpoints

6.5 Safari


Google
http://www.google.es/#spf=1583094364252
Apple Yahoo! Google Maps YouTube Wikipedia Noticias Populares
Búsqueda Imágenes Maps Play YouTube Noticias Gmail Drive Más
Iniciar sesión
Google
Avanzado
General Apariencia Favoritos Pestañas RSS Autorrelleno Seguridad Privacidad Extensiones Avanzado
Acceso Universal: No utilizar nunca un tamaño de letra inferior a 9
Pulsar Tabulador para resaltar cada ítem de una página web
Hoja de estilo: Ninguna seleccionada
Proxies: Cambiar ajustes...
Espacio por omisión para guardar bases de datos: 5 MB
Mostrar el menú Desarrollo en la barra de menús

6.5.2 Safari


Google
http://www.google.es/#spf=1583094503100
Apple Yahoo! Google Maps YouTube Wikipedia Noticias Populares
Búsqueda Imágenes Maps Play YouTube Noticias Gmail Drive Más
Google
Buscar con Google
Voy a tener suerte
Nueva pestaña Ctrl+T
Nueva ventana Ctrl+N
Deshacer Ctrl+Z
Rehacer Ctrl+Mayús+Z
Cortar Ctrl+X
Copiar Ctrl+C
Pegar Ctrl+V
Buscar... Ctrl+F
Zoom
Desarrollo
Codificación del texto
Ver código fuente Ctrl+Alt+U
Enviar enlace a esta página por correo Ctrl+Mayús+I
Guardar como... Ctrl+S
Imprimir... Ctrl+P
Abrir página con
Agente de usuario
Mostrar inspector web Ctrl+Alt+I
Mostrar consola de errores Ctrl+Alt+C
Mostrar editor de fragmentos
Mostrar creador de extensiones
Iniciar depuración de JavaScript
Iniciar parametrización de JavaScript Ctrl+Alt+P
Enviar cabecera HTTP "No controlar"
Desactivar cachés
Desactivar imágenes
Desactivar estilos
Desactivar JavaScript
Desactivar temporizador Runaway JavaScript
Desactivar alteraciones de los sitios web
Elementos Recursos Red Scripts Control temporal Perfiles Auditorías Consola
sem_7769640a0fde87f12993e8df
Expresiones de inspección
Grupo de llamadas
No pausado
Variables disponibles
No pausado
Puntos de interrupción
Ningún punto de interrupción
Web workers

Bibliography

- Debugger definition
- Strategies
- Visual Studio Code
- Using Nodejs in Visual Studio Code

- Debugging a chat in VSC
- Tutorial and Install Nodejs in VSC
- Node.js
- Google Dev Tools

Github

- This presentation with all the examples used on it are available in our public repository at github:

https://github.com/ULL-ESIT-INF-PAI-2019-2020/2019-2020-pai-trabajo-debugging-adrian-rodriguez-sergio-tabares

Contact

- Sergio Tabares Hernández: alu0101124896@ull.edu.es
- Adrián Epifanio Rodríguez Hernández: alu0101158280@ull.edu.es

Thanks you for your attention, if you have any question please let us to know it.