

FORMACIÓN PROFESIONAL  
MONTECASTELO

# Depuración de JavaScript

Desarrollo Web en Entorno Cliente

Ciclo Superior de Desarrollo de Aplicaciones Web

2023/2024

# Herramientas de desarrollo

- **Editor** (creación y modificación de código)
  - Los archivos de JavaScript son ficheros de texto plano.
  - Por tanto pueden ser editados en cualquier editor de texto.
  - Desde el centro se recomienda el empleo de [Visual Studio Code](#).



# Herramientas de desarrollo

- **Navegador Web** (depuración de código)
  - Cualquier navegador web moderno puede ser empleado para probar los archivos desarrollados.
  - Desde el centro, se recomienda el empleo del navegador web [Google Chrome](#) o [Mozilla Firefox](#).

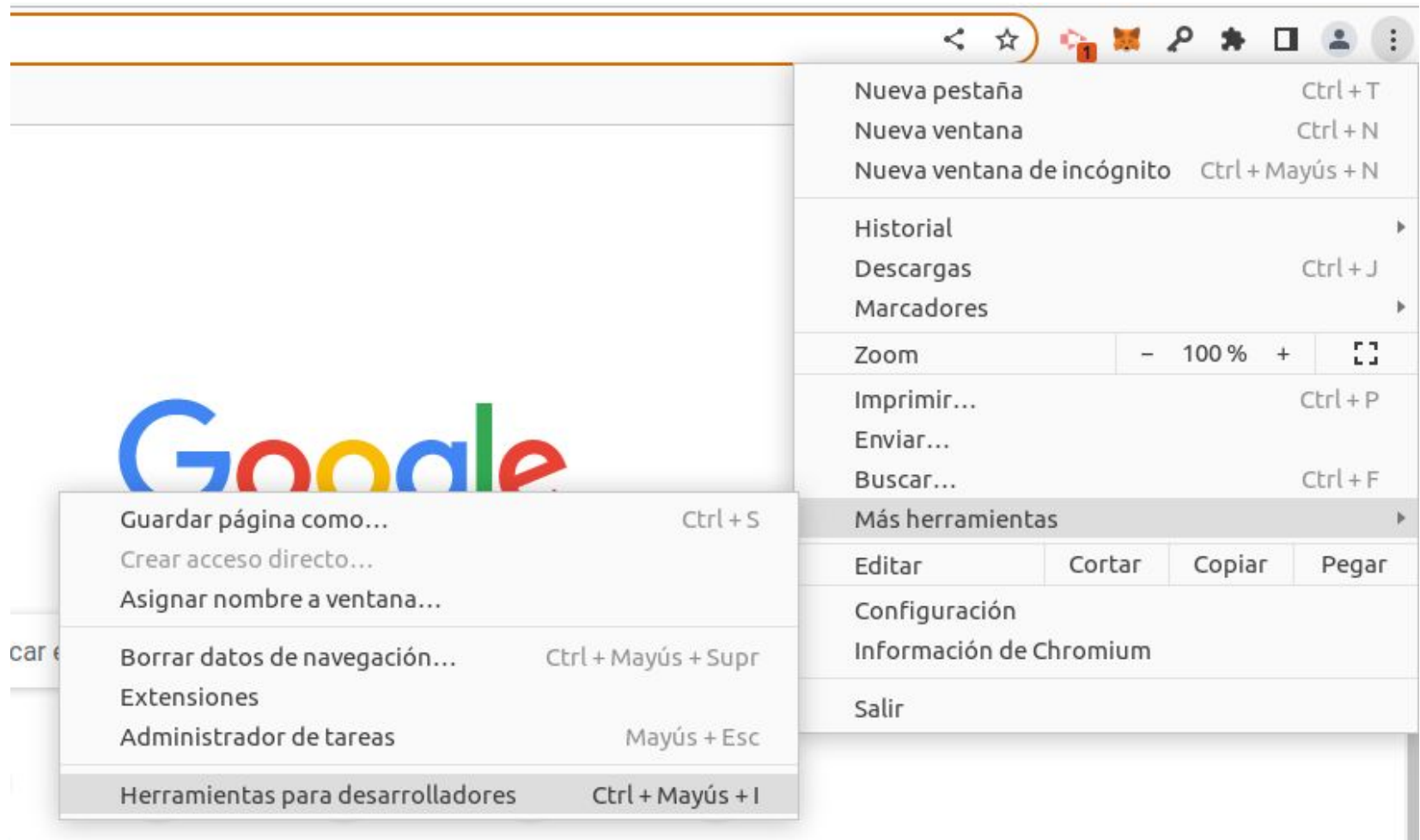


# Depuración en navegador web

- Todos los navegadores web modernos incluyen una sección *Herramientas para desarrolladores*.
- Contienen utilidades varias que asisten al desarrollador de aplicaciones web.
- Accesible:
  - pulsando la tecla rápida F12.
  - opciones del navegador → más herramientas



# Depuración en navegador web

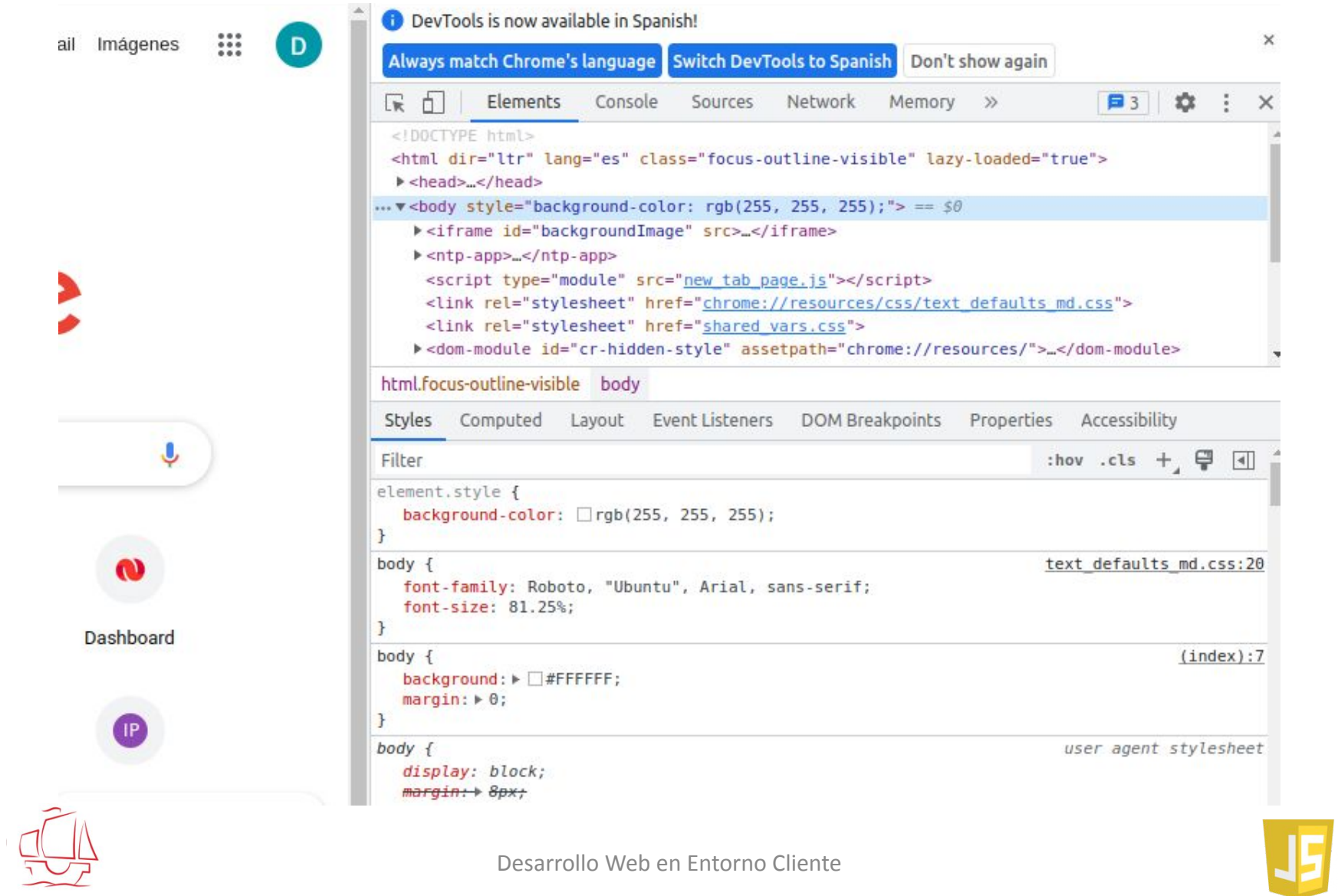


# Depuración en navegador web

- **Elementos** (pestaña *elements*).
  - Permite inspeccionar y editar todos los elementos HTML y CSS de la página cargada.
  - Situando el cursor encima del código HTML se pueden ver de manera interactiva los bloques visuales a los que hacen referencia.



# Depuración en navegador web



The screenshot shows a web browser interface with the DevTools panel open. The top bar includes a search icon, the text "ail Imágenes", and a circular profile icon with the letter "D". A notification banner at the top of DevTools states "DevTools is now available in Spanish!" with buttons for "Always match Chrome's language", "Switch DevTools to Spanish", and "Don't show again".

The DevTools panel has tabs for Elements, Console, Sources, Network, and Memory. The Elements tab is active, displaying the DOM tree. The selected element is the `body` tag, which has the following attributes:

```
<!DOCTYPE html>
<html dir="ltr" lang="es" class="focus-outline-visible" lazy-loaded="true">
  <head>...</head>
  <body style="background-color: rgb(255, 255, 255);"> == $0
    <iframe id="backgroundImage" src=...</iframe>
    <ntp-app>...</ntp-app>
    <script type="module" src="new_tab_page.js"></script>
    <link rel="stylesheet" href="chrome://resources/css/text_defaults_md.css">
    <link rel="stylesheet" href="shared_vars.css">
    <dom-module id="cr-hidden-style" assetpath="chrome://resources/">...</dom-module>
```

Below the DOM tree, the Styles pane is open, showing the computed styles for the selected `body` element. The styles are:

- `background-color: rgb(255, 255, 255);`
- `font-family: Roboto, "Ubuntu", Arial, sans-serif;`
- `font-size: 81.25%;`
- `background: #FFFFFF;`
- `margin: 0;`
- `display: block;`
- `margin: 8px;`

The styles are sourced from `text_defaults_md.css:20`, `(index):7`, and the `user agent stylesheet`.

On the left side of the browser, there is a sidebar with a search icon, a "Dashboard" button, and an "IP" button. At the bottom left, there is a small red icon of a sailboat. At the bottom right, there is a yellow JavaScript logo.

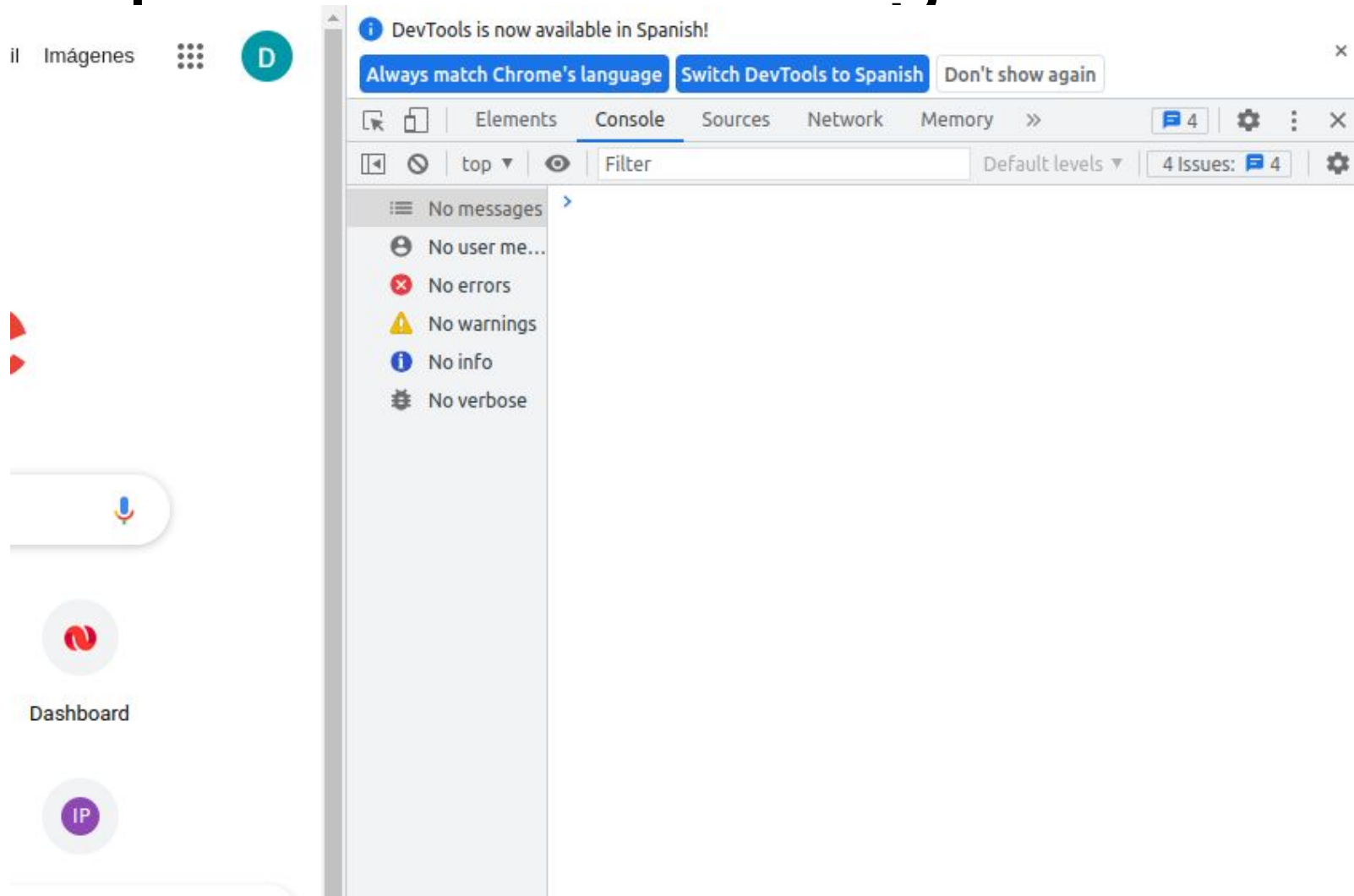
# Depuración en navegador web

- **Consola** (pestaña *console*).
  - Utilidad muy potente incluida en las herramientas para desarrolladores del navegador web que permite imprimir todo tipo de variables, objetos complejos y *arrays*, con la posibilidad también de inspeccionar todas sus propiedades.





# Depuración en navegador web



# Depuración en navegador web

- **Consola** (pestaña *console*).
  - Muestra los mensajes generados por el intérprete de JavaScript, tanto durante la carga como durante la ejecución de una página (errores, warnings, etc).
  - Además, muestra el resultado de las llamadas a `console.log()` desde nuestro código de programación.



# Depuración en navegador web

- **Consola** (pestaña *console*).
  - Cuando estamos desarrollando, es buena práctica abrir la consola cada vez que se carga una página, ya que si hubiese algún error en el código inicial de la misma, la consola lo mostraría nada más cargarla.

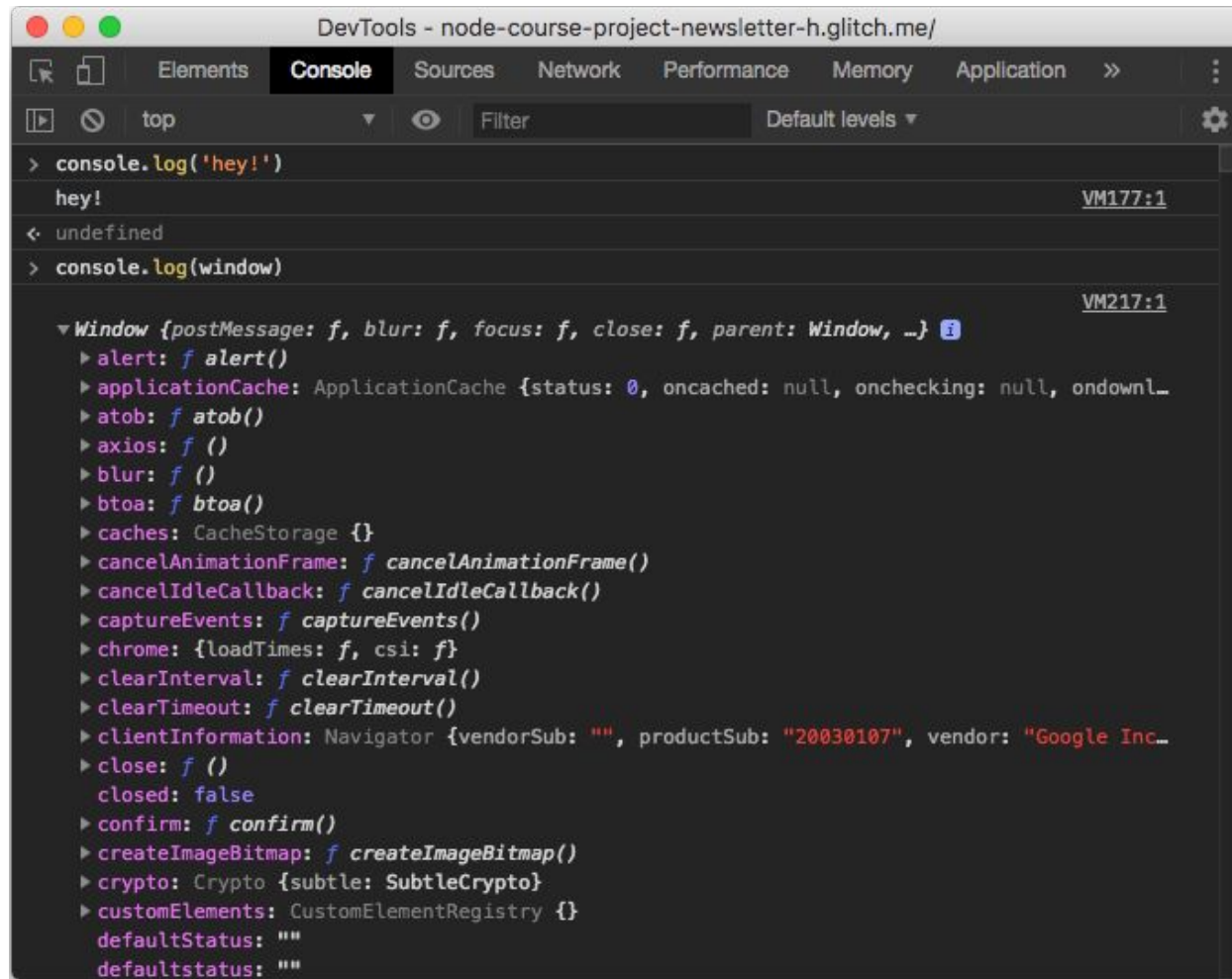


# Depuración en navegador web

- **Consola** (pestaña *console*).
  - También permite ejecutar cualquier instrucción o fragmento de código JS con tan solo escribirlo en la misma y pulsar *ENTER*.
  - Para introducir varias líneas de código se puede pulsar *SHIFT+ENTER*, para que la consola cambie de línea en lugar de ejecutar el código introducido.



# Depuración en navegador web

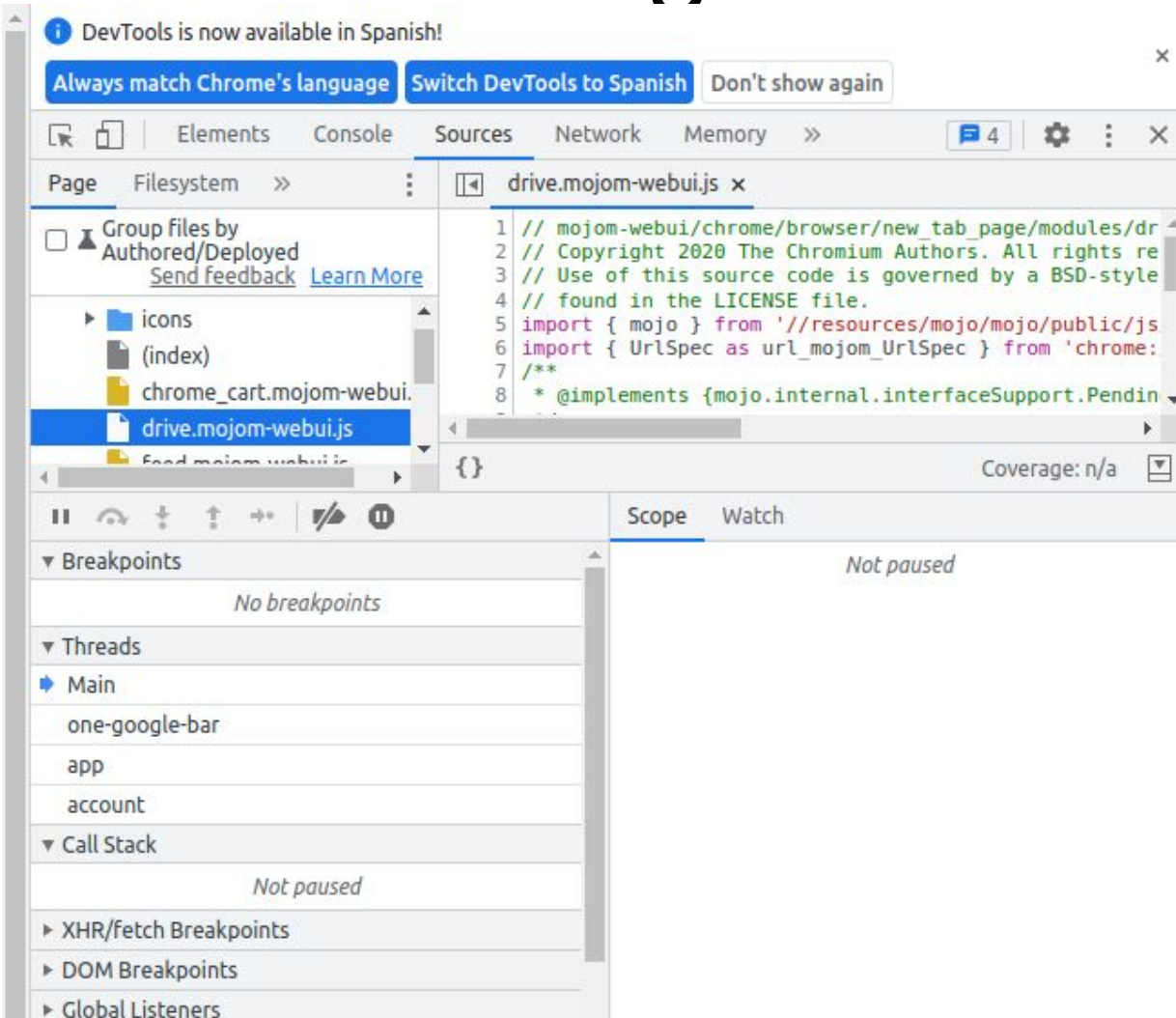


The screenshot shows the Chrome DevTools Console for the page 'node-course-project-newsletter-h.glitch.me/'. The 'Console' tab is active, showing two log entries. The first entry is 'hey!' from 'VM177:1'. The second entry is the result of 'console.log(window)', which is a 'Window' object. The object is collapsed, showing a list of properties and methods including 'alert', 'applicationCache', 'atob', 'axios', 'blur', 'btoa', 'caches', 'cancelAnimationFrame', 'cancelIdleCallback', 'captureEvents', 'chrome', 'clearInterval', 'clearTimeout', 'clientInformation', 'close', 'confirm', 'createImageBitmap', 'crypto', 'customElements', 'defaultStatus', and 'defaultstatus'.

```
> console.log('hey!')
hey! VM177:1
< undefined
> console.log(window)
Window {postMessage: f, blur: f, focus: f, close: f, parent: Window, ...} i VM217:1
  ▶ alert: f alert()
  ▶ applicationCache: ApplicationCache {status: 0, oncached: null, onchecking: null, ondownl...
  ▶ atob: f atob()
  ▶ axios: f ()
  ▶ blur: f ()
  ▶ btoa: f btoa()
  ▶ caches: CacheStorage {}
  ▶ cancelAnimationFrame: f cancelAnimationFrame()
  ▶ cancelIdleCallback: f cancelIdleCallback()
  ▶ captureEvents: f captureEvents()
  ▶ chrome: {loadTimes: f, csi: f}
  ▶ clearInterval: f clearInterval()
  ▶ clearTimeout: f clearTimeout()
  ▶ clientInformation: Navigator {vendorSub: "", productSub: "20030107", vendor: "Google Inc...
  ▶ close: f ()
  ▶ closed: false
  ▶ confirm: f confirm()
  ▶ createImageBitmap: f createImageBitmap()
  ▶ crypto: Crypto {subtle: SubtleCrypto}
  ▶ customElements: CustomElementRegistry {}
  ▶ defaultStatus: ""
  ▶ defaultstatus: ""
```



# Depuración en navegador web



The screenshot displays the Google Chrome DevTools interface. At the top, a notification states "DevTools is now available in Spanish!" with buttons for "Always match Chrome's language", "Switch DevTools to Spanish", and "Don't show again". The main toolbar includes tabs for "Page", "Elements", "Console", "Sources", "Network", and "Memory". The "Sources" panel is active, showing a file tree on the left with "drive.mojom-webui.js" selected. The right pane displays the source code of this file, which includes comments and JavaScript imports. Below the code editor, the "Scope" and "Watch" tabs are visible. The "Scope" tab shows a list of variables: "Main", "one-google-bar", "app", and "account". The "Watch" tab is currently empty. The bottom of the interface features a "Dashboard" section with a "IP" button and a "Global Listeners" section.

DevTools is now available in Spanish!

Always match Chrome's language Switch DevTools to Spanish Don't show again

Page Elements Console Sources Network Memory

drive.mojom-webui.js

```
1 // mojom-webui/chrome/browser/new_tab_page/modules/dr
2 // Copyright 2020 The Chromium Authors. All rights re
3 // Use of this source code is governed by a BSD-style
4 // found in the LICENSE file.
5 import { mojo } from '../resources/mojom/mojom/public/js
6 import { UrlSpec as url_mojom_UrlSpec } from 'chrome:
7 /**
8  * @implements {mojom.internal.interfaceSupport.Pendin
```

Group files by Authored/Deployed Send feedback Learn More

icons (index) chrome\_cart.mojom-webui.js drive.mojom-webui.js

Scope Watch

Breakpoints No breakpoints

Threads Main one-google-bar app account

Call Stack Not paused

XHR/fetch Breakpoints DOM Breakpoints Global Listeners

Coverage: n/a

Dashboard IP



# Depuración en navegador web

- **Depurador** (pestaña *sources*).
  - El depurador se divide en tres secciones.
  - La posición de dichas secciones puede variar en función de la configuración específica de cada navegador.



# Depuración en navegador web

- **Depurador** (pestaña *sources*).
  - Sección 1: selección del fichero cuyo código se desea depurar, en el que se pueden colocar puntos de ruptura (breakpoint) en cualquier línea, tal como veremos más adelante.
  - Sección 2: inspección del código correspondiente al fichero que se ha seleccionado en la sección anterior.





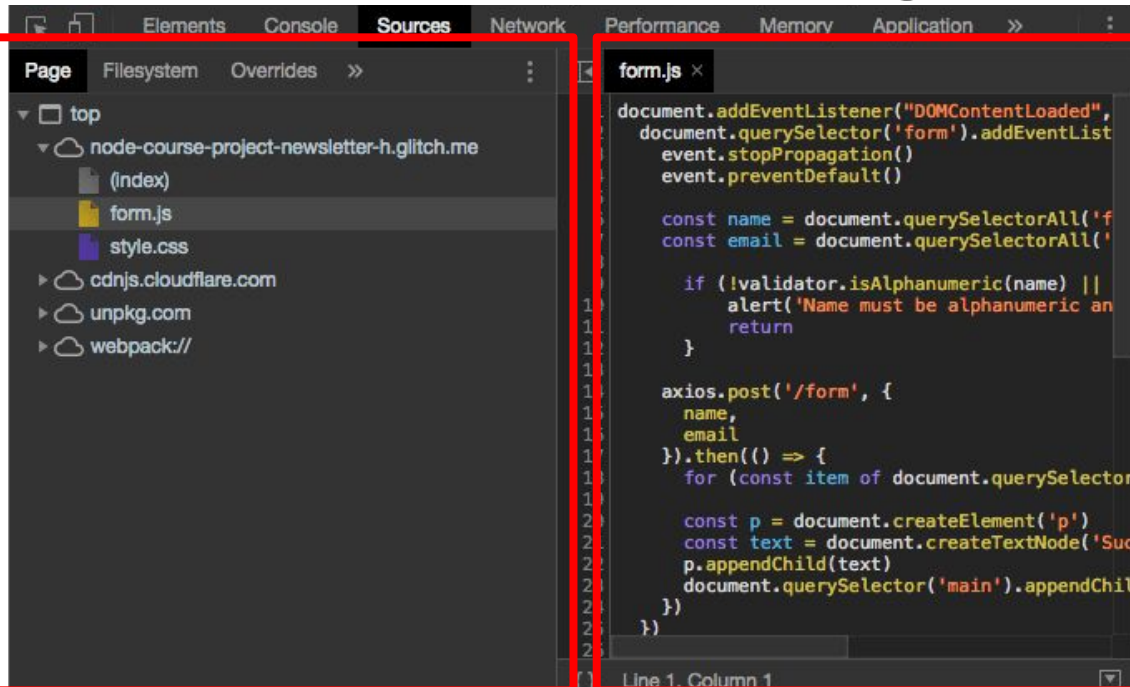
# Depuración en navegador web

- **Depurador** (pestaña *sources*).
  - Sección 3: vista del propio depurador, con la posibilidad de inspeccionar variables y controlar la ejecución paso a paso.



# Depuración en navegador web

SECCIÓN 1



SECCIÓN 2



SECCIÓN 3



# Depuración en navegador web

- Puntos de ruptura.
  - Cuando se carga una página, el código JS se ejecuta hasta que se encuentre un punto de ruptura.
  - En ese momento, la ejecución se detiene y se puede analizar el estado del programa, inspeccionar las variables del mismo y continuar la ejecución.

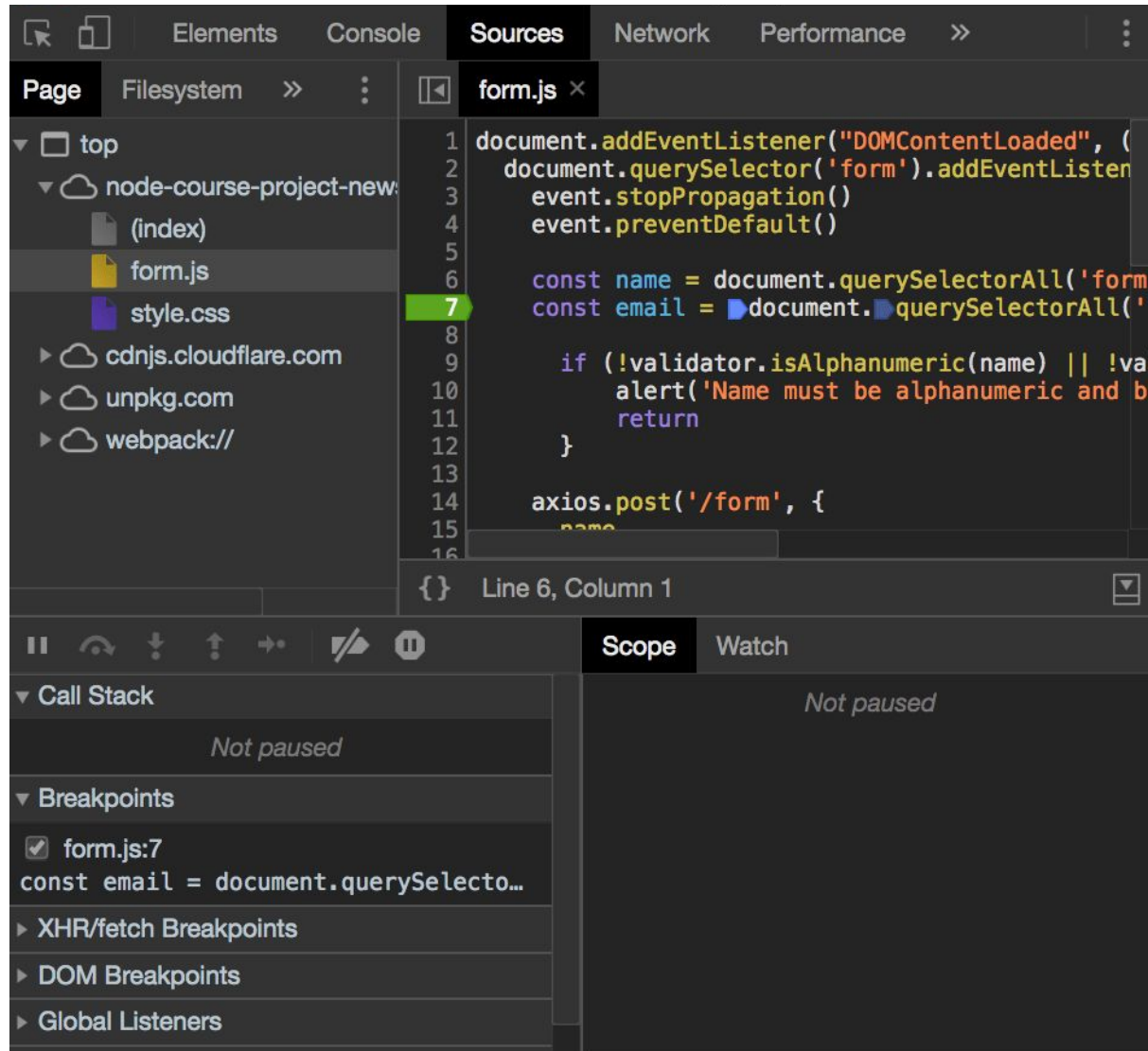


# Depuración en navegador web

- *Puntos de ruptura.*
  - Los puntos de ruptura se insertan simplemente haciendo clic con el ratón en la línea correspondiente.
  - Si se vuelve a hacer clic con el ratón, el punto de ruptura desaparecerá.



# Depuración en navegador web



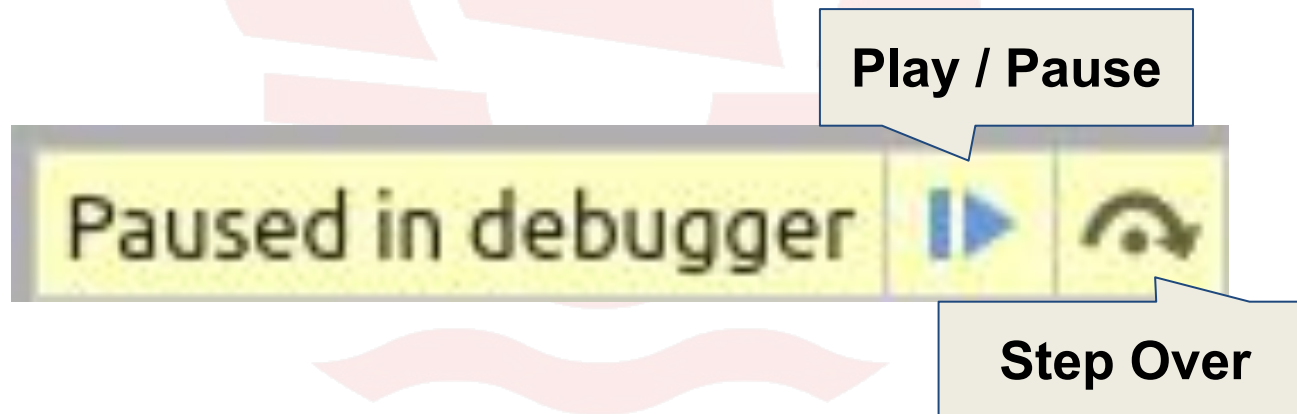
# Depuración en navegador web

- *Puntos de ruptura.*
  - Después de añadir el punto de ruptura se debe recargar la página (F5) para que la ejecución se detenga en el nuevo punto de ruptura.
  - A partir de ese punto de ruptura podemos controlar la ejecución del código mediante un conjunto de botones.



# Depuración en navegador web

- Puntos de ruptura.
  - *Play/Pause*: permite reanudar la ejecución del programa.
  - *Step over*: ejecuta la siguiente línea de código y vuelve a detenerse.



# Depuración en navegador web

- Puntos de ruptura.
  - Cuando el breakpoint está ubicado dentro o fuera de una función, aparecen los siguientes botones:
    - *Step into*: se adentra en el código de la función para inspeccionarla paso a paso.
    - *Step out*: sale de la función a la que previamente se ha entrado (comportamiento contrario a Step into).





# Depuración en navegador web

- *Puntos de ruptura.*
  - A medida que se añaden puntos de ruptura, se muestran en el panel *Breakpoints*, donde también se pueden activar y desactivar temporalmente.



# Depuración en navegador web

- *Puntos de ruptura.*
  - Existen varios tipos de puntos de ruptura:
    - *Event listener breakpoints*: lanzados cuando ocurre algún evento en la página.
    - *XHR/fetch breakpoints*: lanzados cuando se envía alguna petición de red.
    - *DOM breakpoints*: lanzados cuando cambia algún elemento del DOM.



# Depuración en navegador web

- Puntos de ruptura.

- Podemos crear puntos de ruptura desde el código fuente a través del comando debugger.

`debugger; // <-- the debugger stops here`

- Este comando sólo funciona cuando el panel de herramientas de desarrollo está abierto, de otro modo el navegador lo ignora.



# Depuración en navegador web

- *Inspección de variables.*
  - En el panel *Scope* se muestran todas las variables que están dentro del ámbito (scope) actual, junto con sus respectivos valores.
  - También se pueden editar los valores haciendo doble clic sobre las mismas.

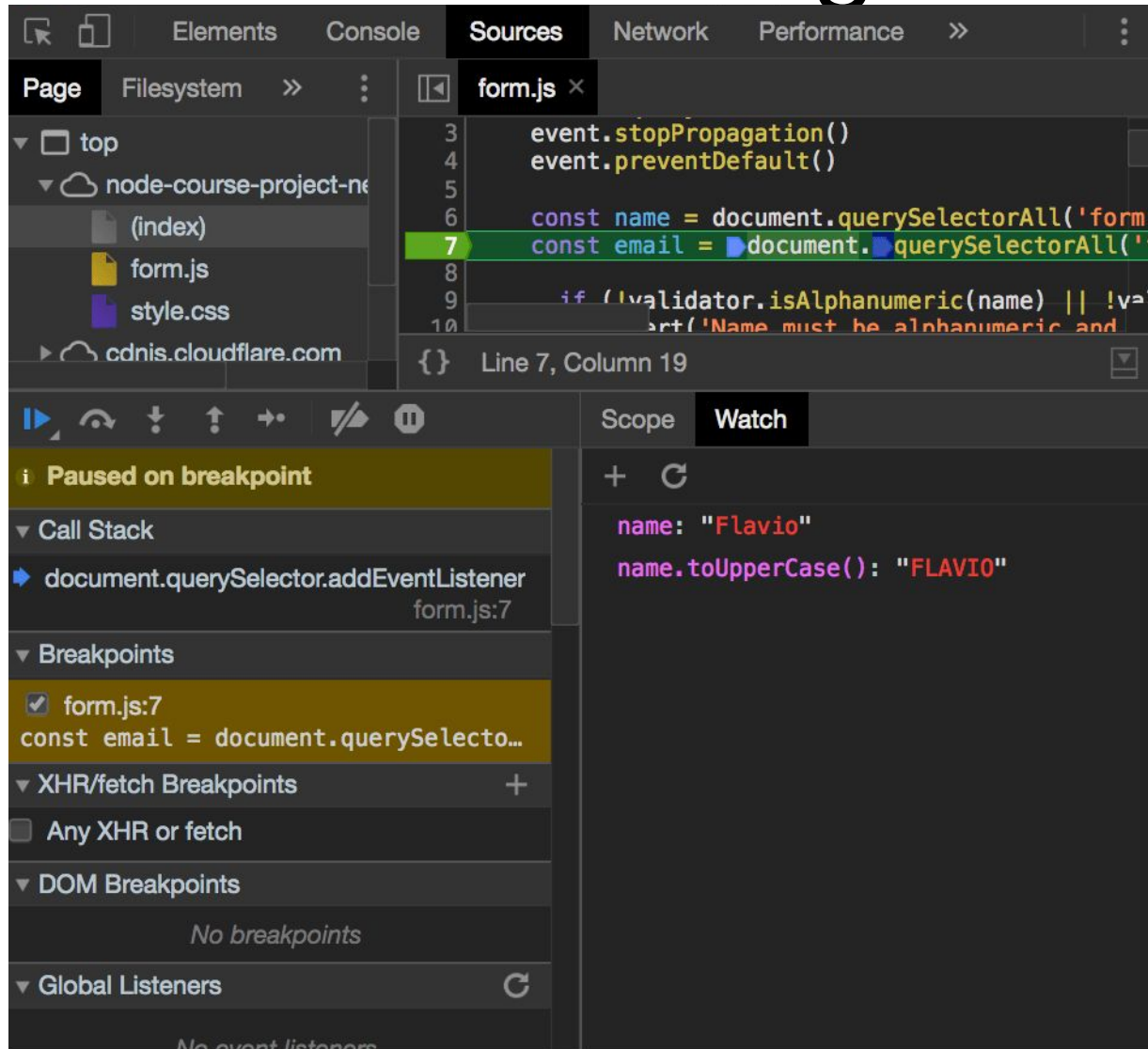


# Depuración en navegador web

- *Inspección de variables.*
  - Junto al panel *Scope* se encuentra el panel *Watch*. Cuenta con un botón **+** que permite añadir expresiones cuyo valor se desea conocer.
  - En el siguiente ejemplo se añaden `name` y `name.toUpperCase()` para conocer los valores de ambas expresiones.



# Depuración en navegador web



# Depuración en navegador web

il Imágenes

DevTools is now available in Spanish!

Always match Chrome's language Switch DevTools to Spanish Don't show again

Elements Console Sources Network Memory

Filter ☐ Invert ☐ Hide data URLs

All Fetch/XHR JS CSS Img Media Font Doc WS Wasm Manifest Other ☐ Has blocked cookies

☐ Blocked Requests ☐ 3rd-party requests

20 ms 40 ms 60 ms 80 ms 100 ms

Name	Status	Type	Initiator	Size	Time	Waterfall
4UabrENHsxJlGDuGo1...	200	font	account?o...	(me...	0 ms	
data:image/svg+xml,...	200	svg+...	account?o...	(me...	0 ms	
m=n73qwfw,ws9Tlc,e5qF...	200	script	m= b, tp,...	(disk...	26...	
m=UMu52b,udhWs,FCJ...	200	script	m= b, tp,...	(disk...	20...	
m=sOXFj,q0xTif,WNBcme	200	script	m= b, tp,...	(disk...	27...	
m=bm51tf	200	script	m= b, tp,...	(disk...	25...	
m=cro4ab	200	script	m= b, tp,...	(disk...	14...	
m=GHwlb	200	script	m= b, tp,...	(disk...	11...	
ListAccounts?listPages=...	200	xhr	m= b, tp,...	511 B	17...	
m=Wt6vjf,hhhU8,FCpbq...	200	script	m= b, tp,...	(disk...	3 ms	
log?format=json&hasfa...	200	xhr	m= b, tp,...	167 B	85...	
log?format=json&hasfa...	200	xhr	m= b, tp,...	179 B	59...	

19 requests 18.7 kB transferred 651 kB resources

Dashboard

IP



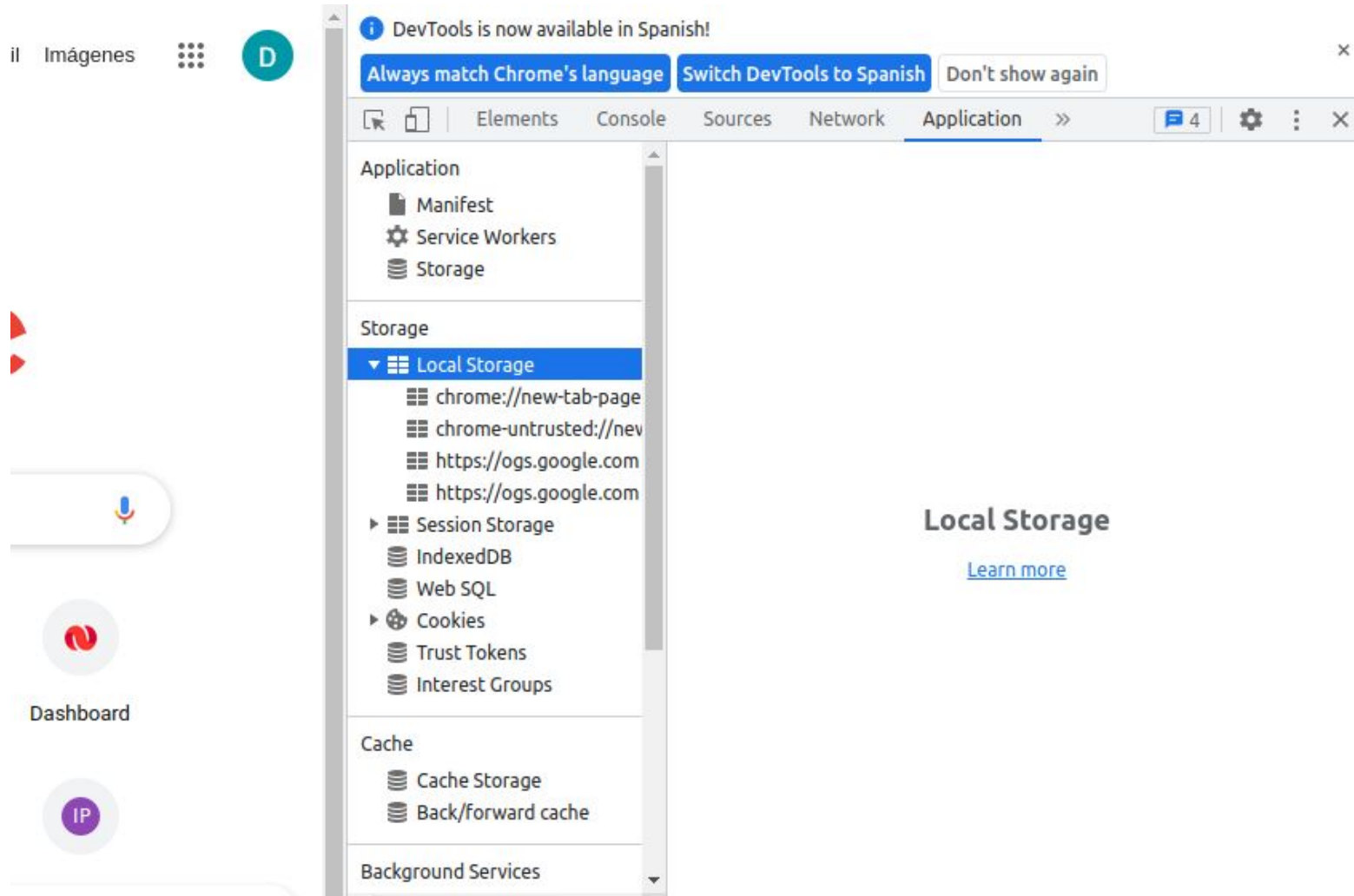
# Depuración en navegador web

- **Analizador de red** (pestaña *network*).
  - Permite comprobar el tráfico entre nuestro navegador y los servidores de Internet.
  - Hay que tener en cuenta que una web, puede contener múltiples fuentes HTML, CSS, JS y librerías y no tienen porqué estar en el mismo servidor.
  - Podemos inspeccionar peticiones GET, POST, etc. y sus respuestas.





# Depuración en navegador web



# Depuración en navegador web

- **Control de aplicación** (pestaña *application*).
  - Nos permite conocer, entre otras cosas, el contenido del almacenamiento del navegador, las cookies, etc.
  - Además también se puede editar dicho contenido.



# Tipos de errores en JS

- **Error de sintaxis.**

- Se produce cuando no seguimos la sintaxis adecuada al escribir el código fuente de un programa.
- Dicho error impedirá al navegador traducir dicha instrucción, ya que, no entenderá qué le está diciendo el programador.



# Tipos de errores en JS

- **Error lógico.**

- Los errores lógicos son errores que impiden que su programa haga lo que estaba previsto.
- Su código puede compilarse y ejecutarse sin errores de sintaxis, pero el resultado de una operación puede generar un resultado no esperado.



# Guía práctica de depuración

- **Error de sintaxis.**

- Inspección de código (requiere experiencia).
- Mensajes de la consola de las *devTools* del navegador.

- **Error lógico.**

- Control de flujo del programa con depurador (*breakpoints*) o consola (`console.log()`).
- Valor de variables con depurador o consola.



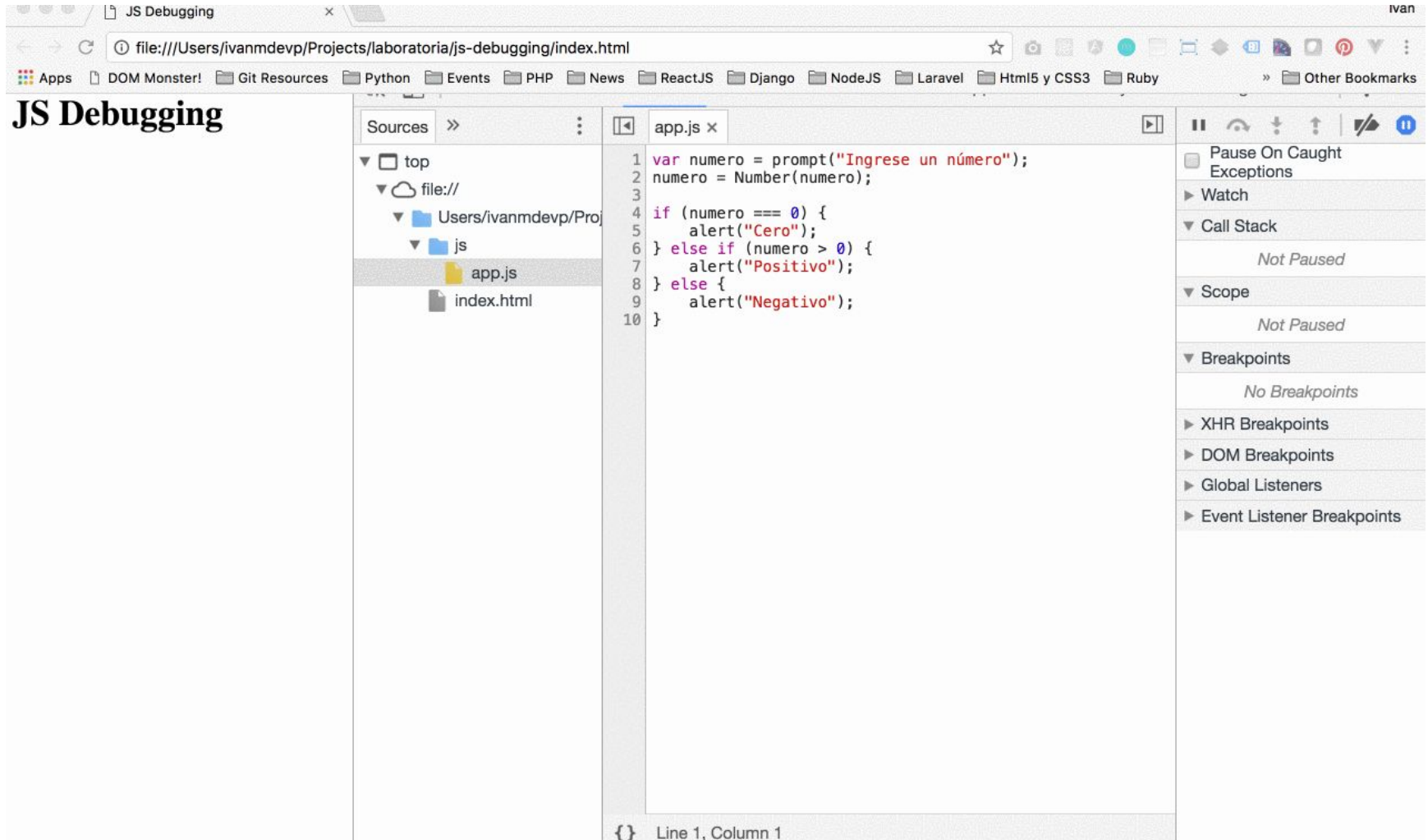
# Ejemplo de depuración

- En el siguiente ejemplo, tenemos un código que verifica si el número ingresado por un usuario es cero, positivo o negativo.

```
var numero = prompt("Ingresa un número");
numero = Number(numero);
if (numero === 0) {
    alert("Cero");
} else if (numero > 0) {
    alert("Positivo");
} else {
    alert("Negativo");
}
```



# Ejemplo de depuración



# Referencias

- [JavaScript en w3schools](#)
- [JavaScript en Mozilla Developer Network](#)







FORMACIÓN PROFESIONAL  
MONTECASTELO