2ºDAW

Resources for Web Developers

BEATRIZ MOREIRA CORDAZZO

Content

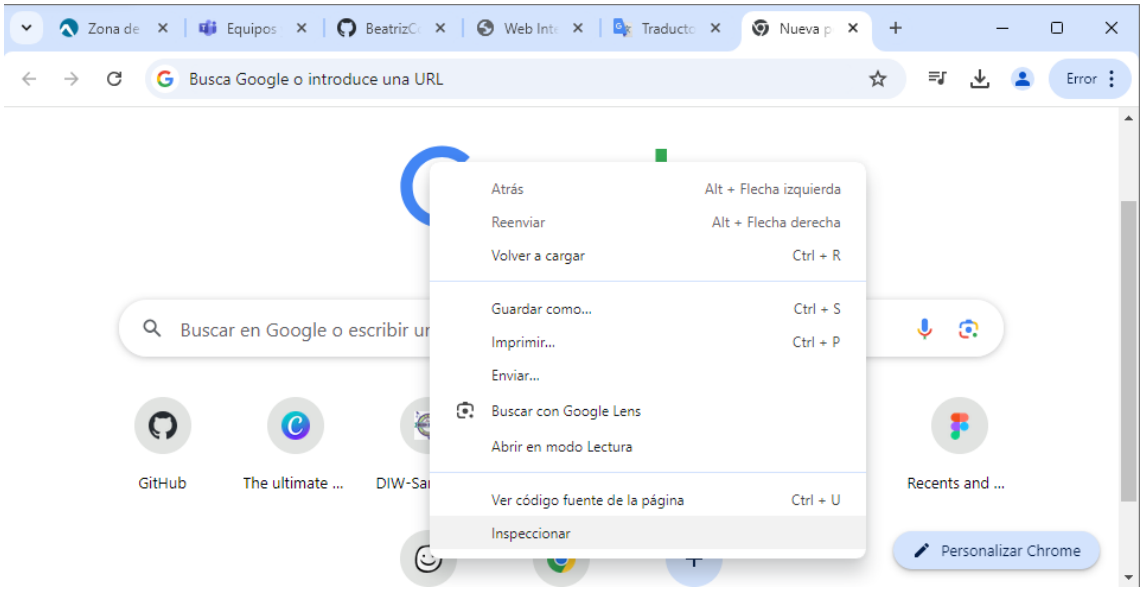
Dev Tools

Developer Tools are built-in tools in web browser that help developers inspect and debug websites. They allow you to see and edit a website's HTML, CSS, and JavaScript, test how it works on different devices, track network activity, and check performance issues. Dev Tools help developers understand, test, and fix how a website looks and behaves without changing the live site.

Chrome

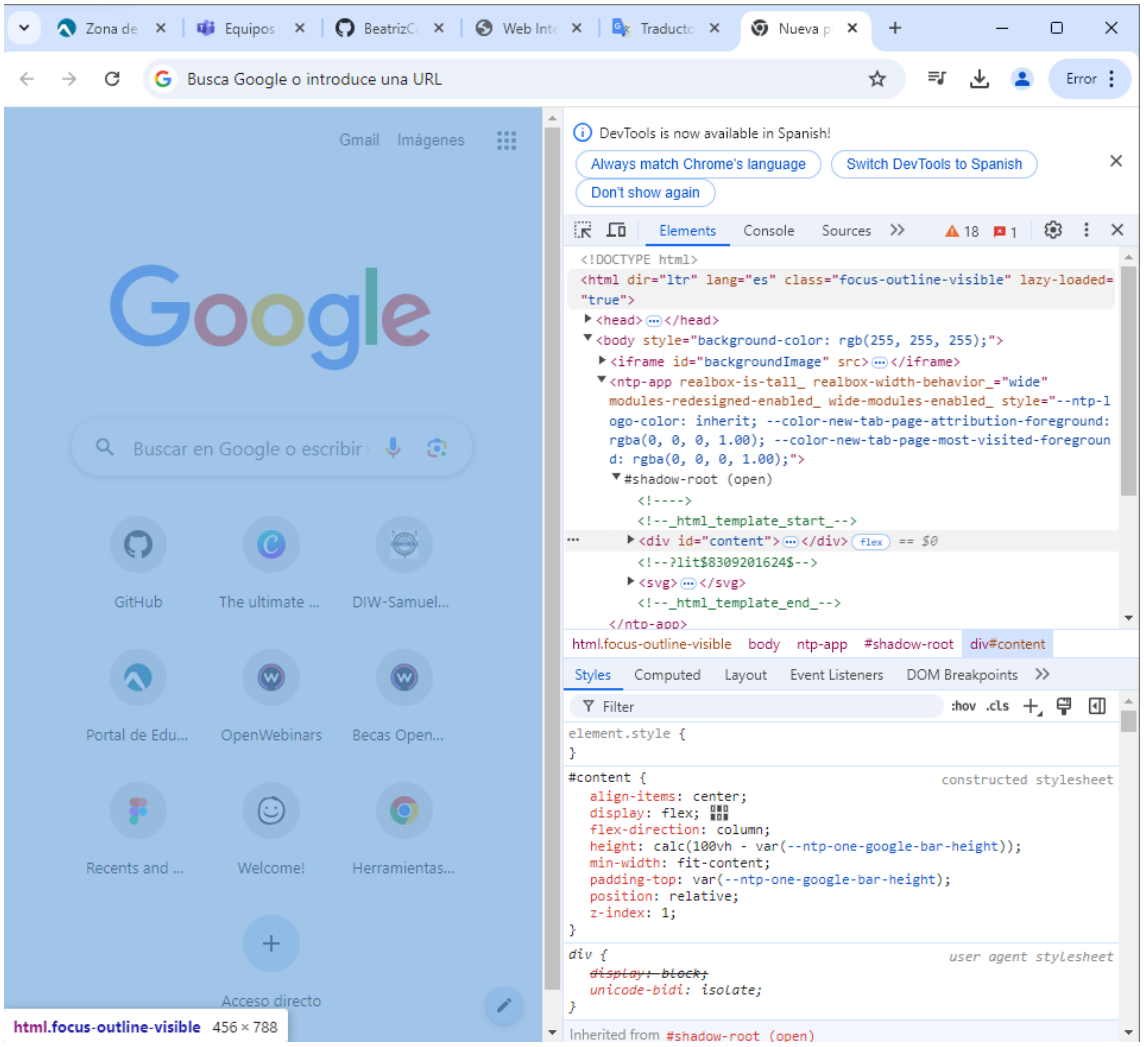
Opening Dev Tools on Chrome

To work with the DOM or CSS, right-click an element on the page and select the inspect option.



Elements

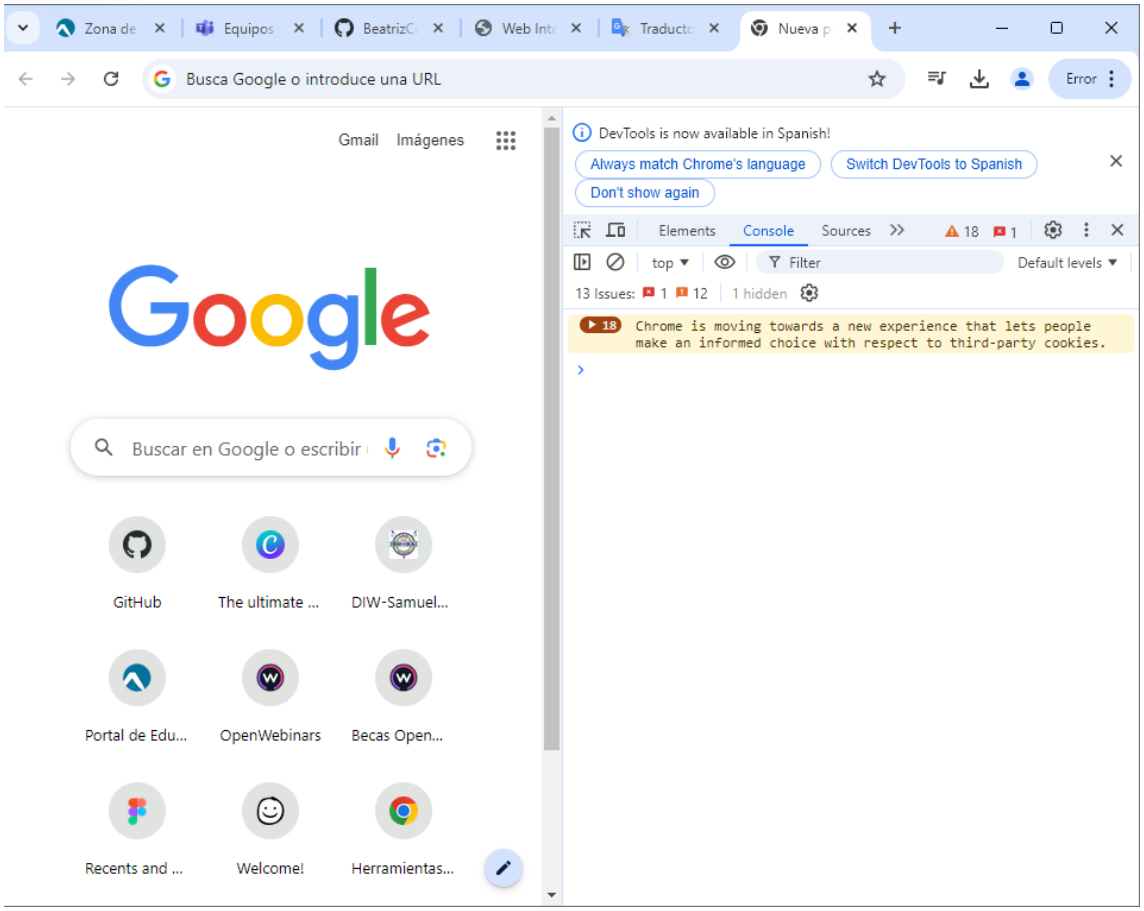
With elements we can see the web page HTML structure:



Console

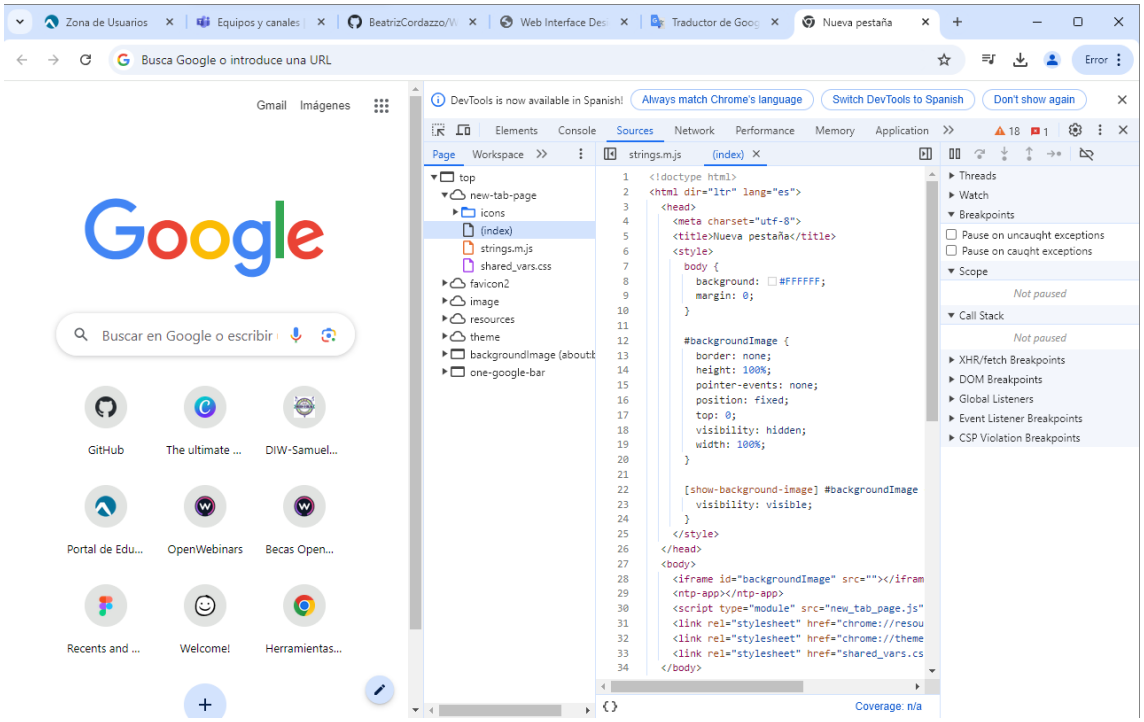
This tool allows us to debug JavaScript code, execute the code in real time, inspect objects and variables, manipulate the DOM, monitor events, check for network errors.

Overall, it is a versatile tool that makes it easy to debug, test, and manipulate your JavaScript code and the DOM.



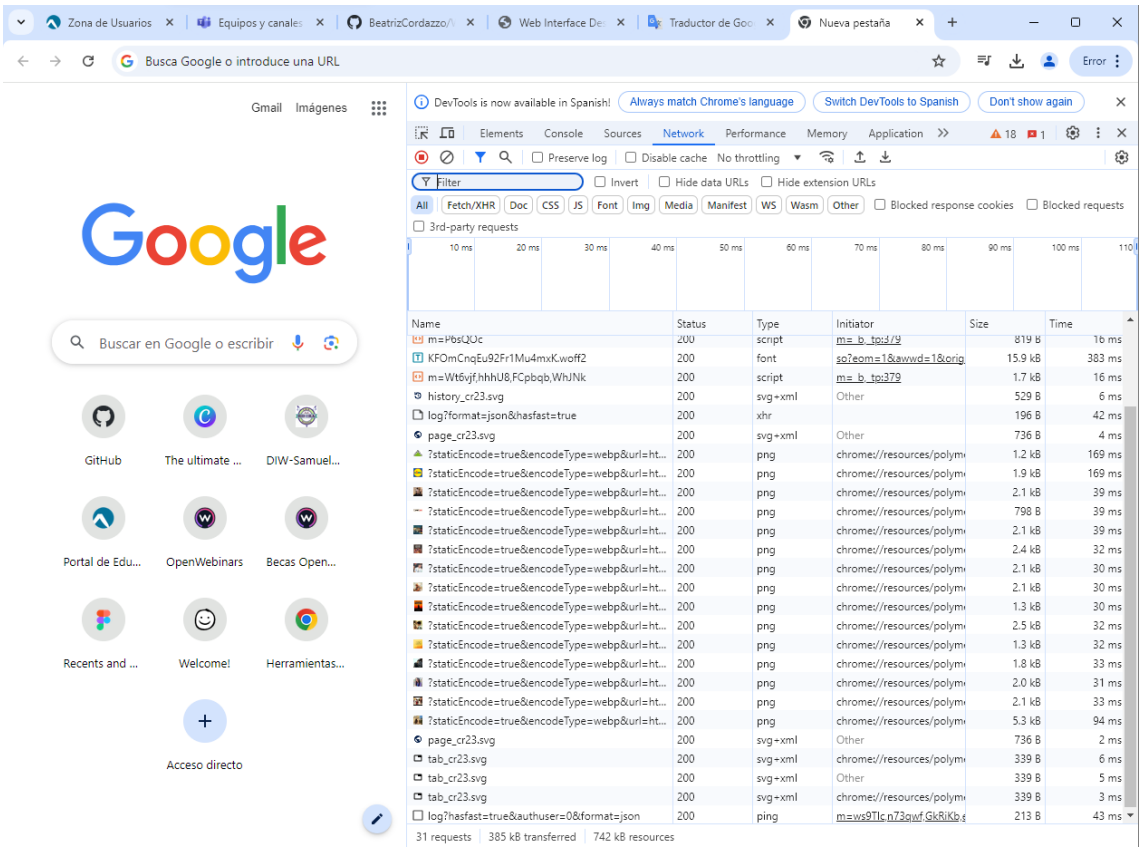
Sources

This section is used to navigate files, edit their code, and also allows you to place break points, inspect the call stack of this, you can look at the call stack that helps you understand how the current state of the break point was reached, look at or monitor specific values as the code progresses, it is used for device emulation and support for modules:



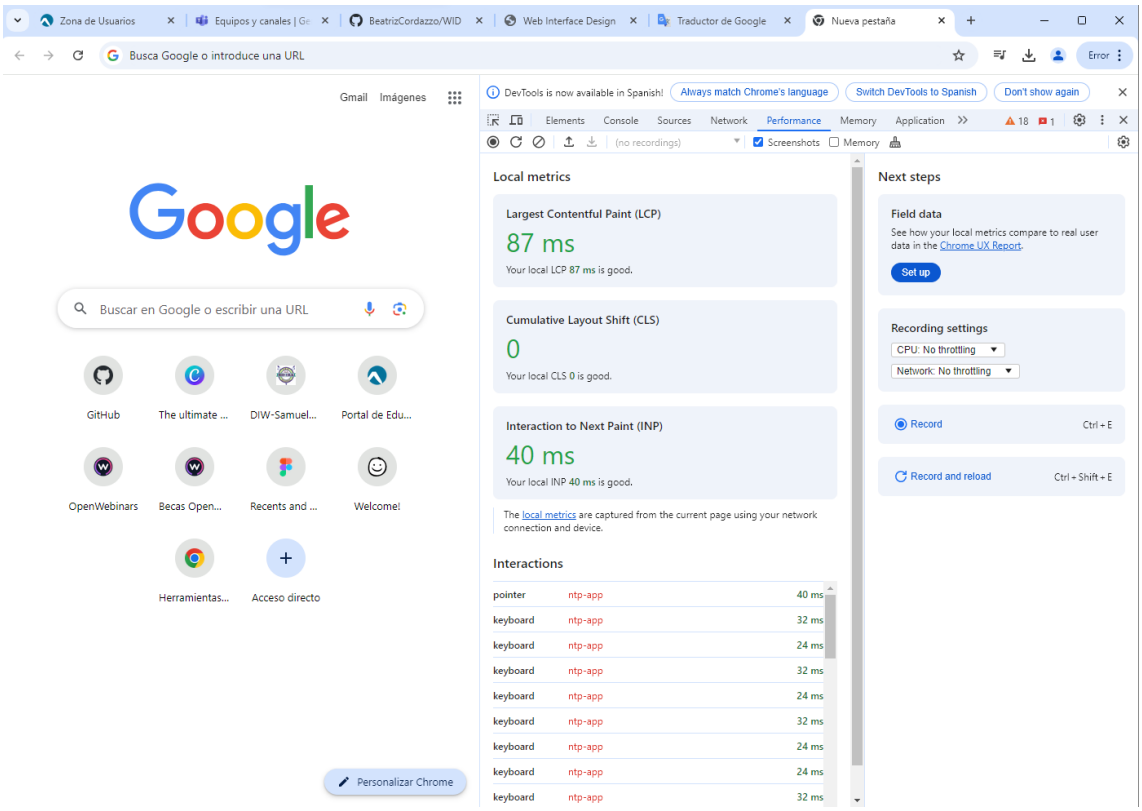
Network

In the network section we can monitor HTTP/HTTPS requests, view responses and completed requests, debug AJAX or Fetch requests, measure web page loading times, observe the cache and its behavior, filter by resource types, inspect cookies, monitor the WebSocket:



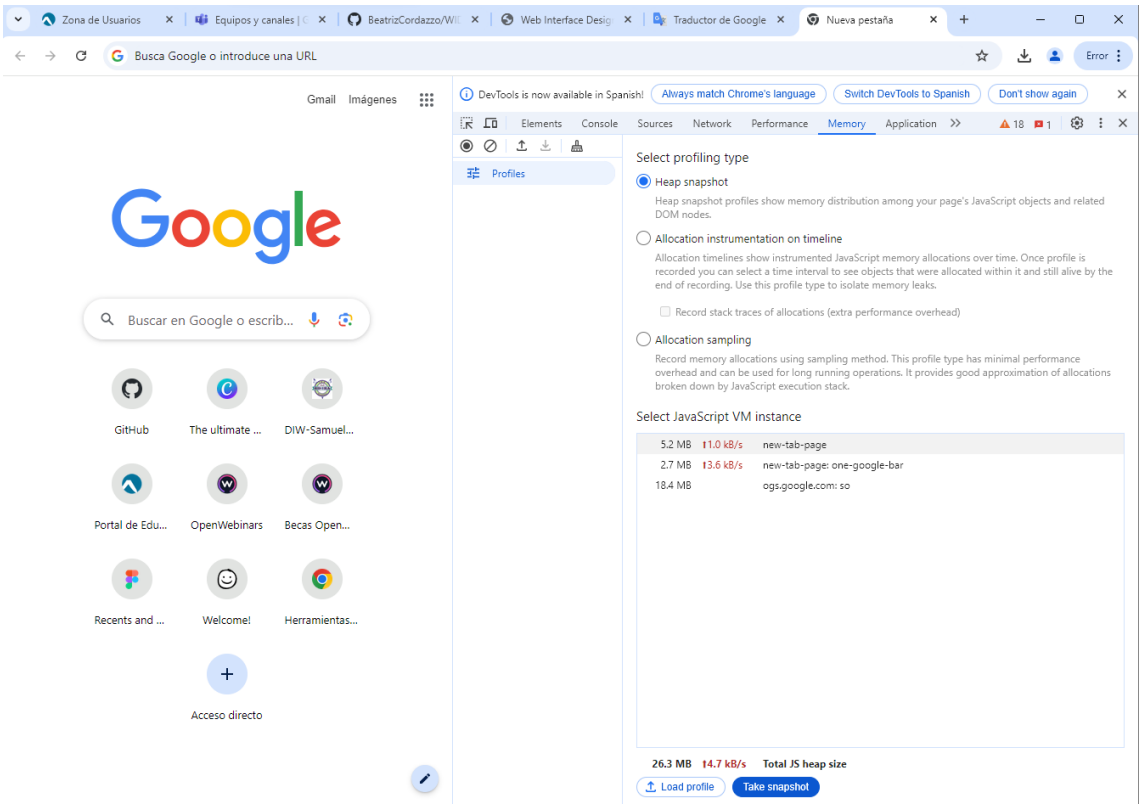
Performance

This section helps us with performance recording, load time analysis, bandwidth performance bottleneck analysis, rendering metrics, memory and JavaScript diagnosis, measuring key performance events, call stack capture, interaction with layout and CSS, animation optimization, debugging interactivity and response times.



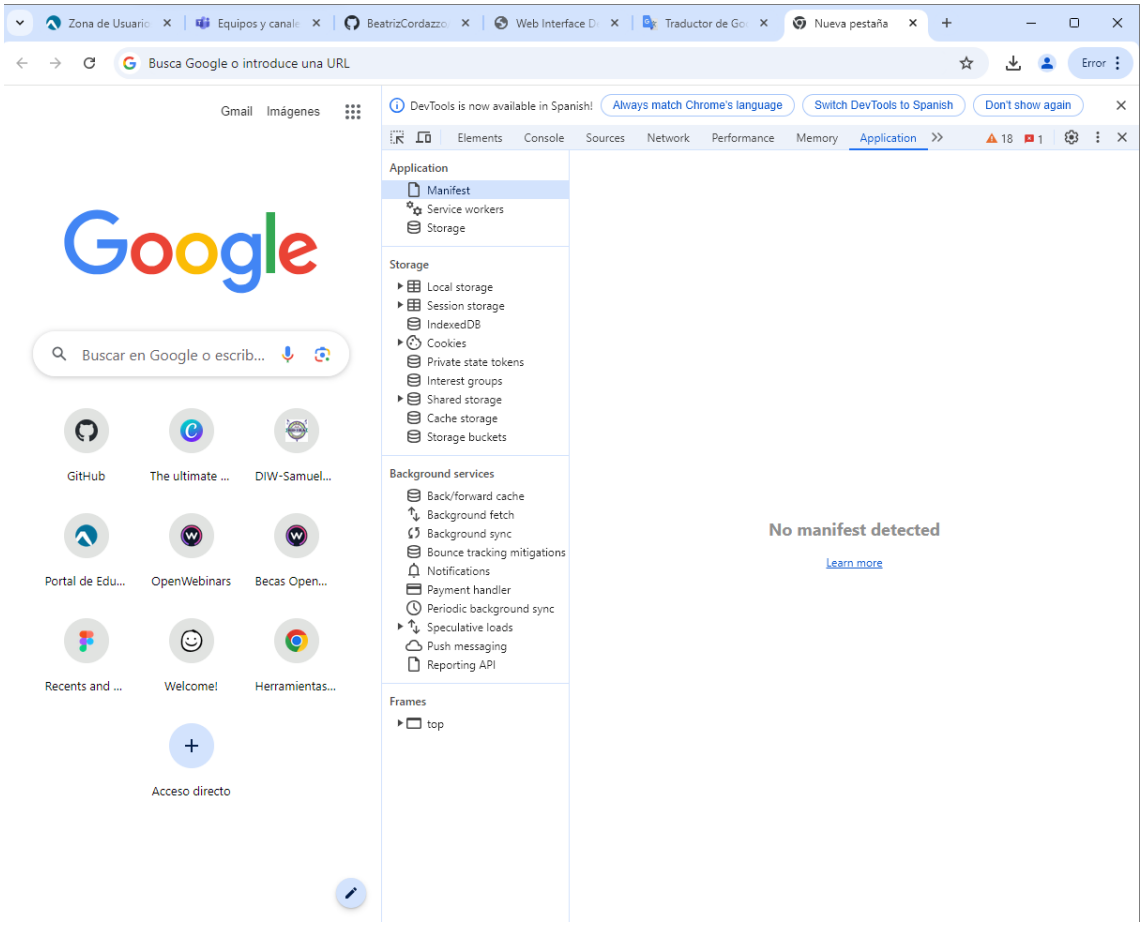
Memory

The memory section will help us detect memory leaks, perform Memory Snapshots, perform Garbage Collection, view the Dominators Tree, and is generally used to analyze and optimize memory usage through snapshots and profiles. It also improves the overall performance of an application by reducing unnecessary resource consumption.



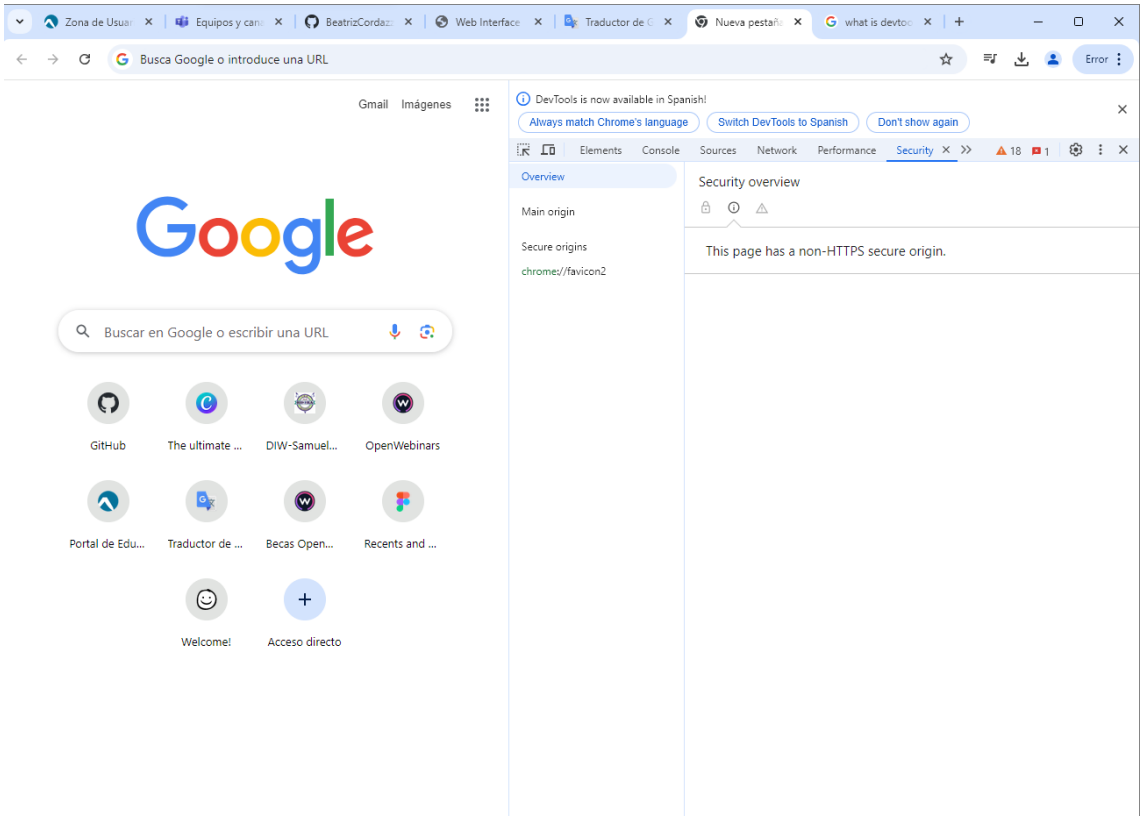
Application

The Application section is designed to manage and debug issues related to local storage, progressive web apps (PWAs), cache, cookies, and other key resources that a web application uses to interact with the browser and deliver a rich experience, even when offline. This section is especially useful for optimizing performance and ensuring proper functionality of applications that rely on local storage and advanced technologies such as Service Workers.



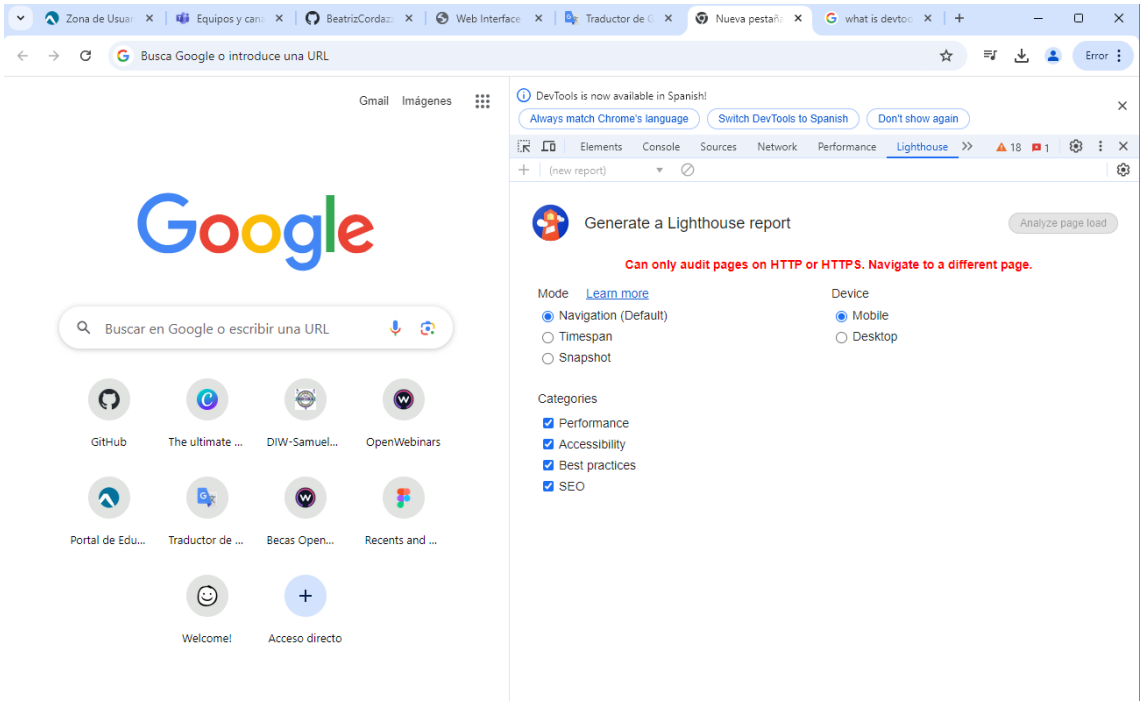
Security

This section is essential to help developers identify and fix security-related issues in their web applications. Some of the features include certificate inspection, content security policies, security alerts, cookie analysis, and security audits.



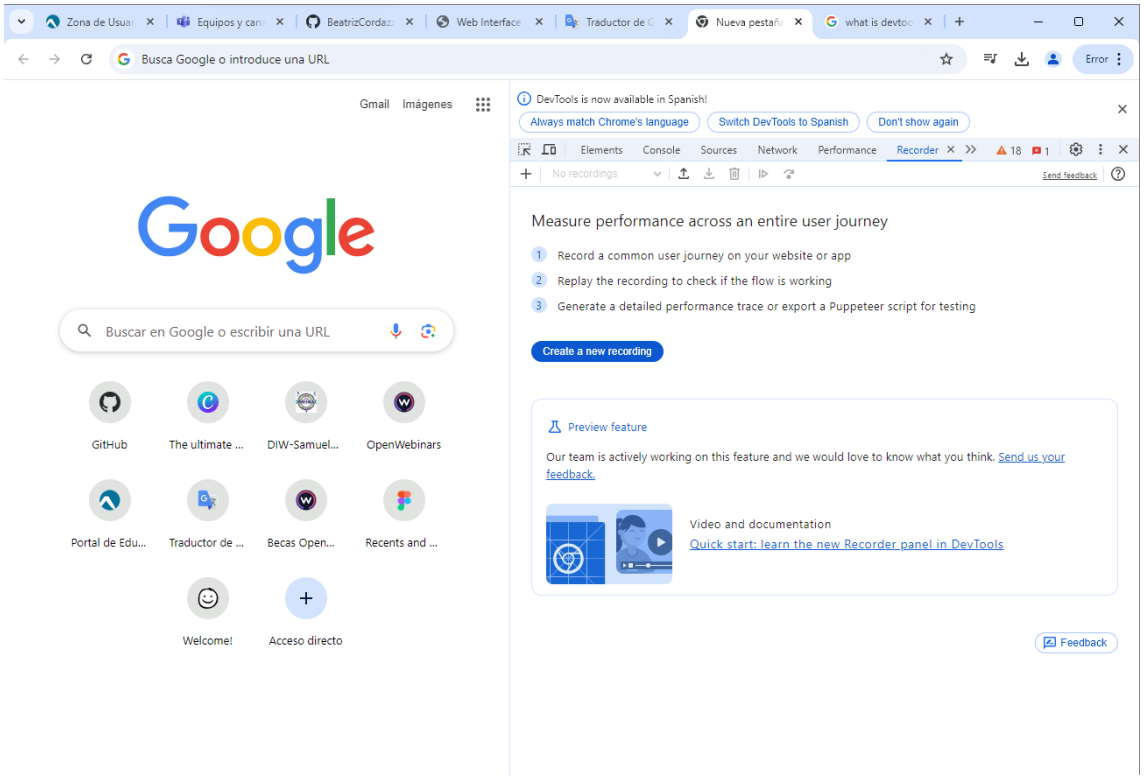
Lighthouse

This tool is used to evaluate the quality of web pages. It also allows us to audit performance, perform SEO (search engine optimization), development practices and PWA (progressive web applications).



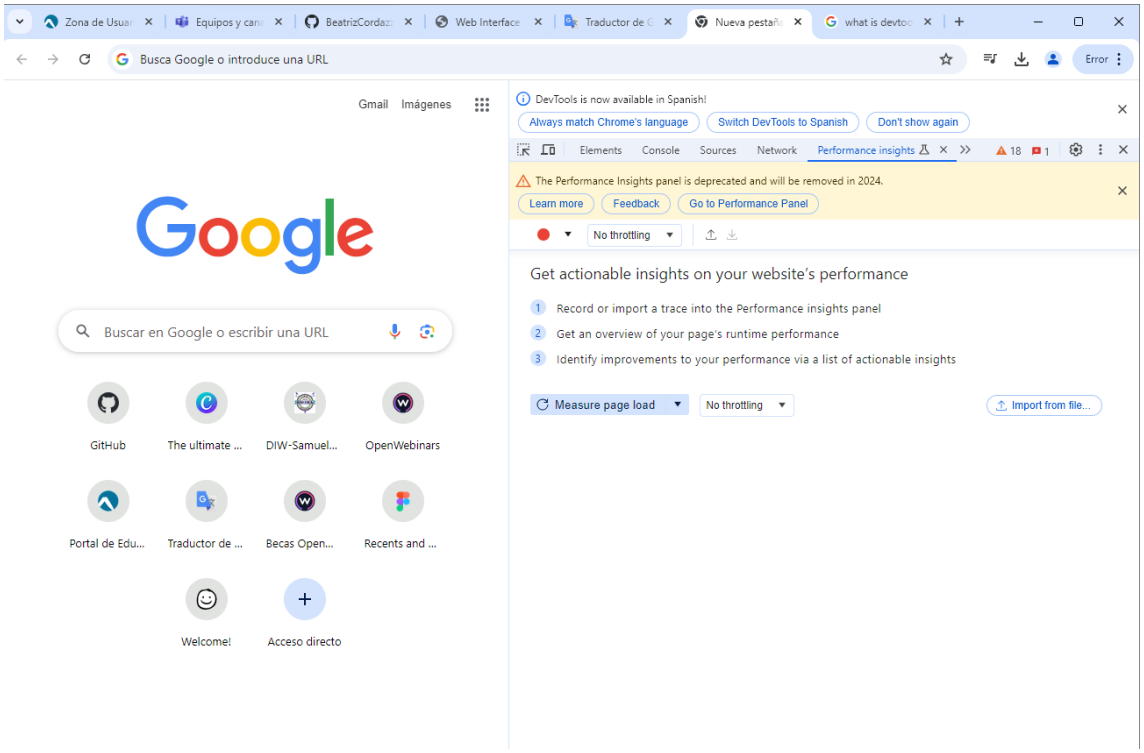
Recorder

This section is used to record and reproduce interactions with a web page. It is used to reproduce scenarios such as the behavior of certain events, and it also facilitates performance analysis.



Performance insights

This section goes further into helping pin-point the problem, and recommend some solutions. For example, the right-hand side lays out a timeline of what's happened during the recording. Each section in the Insights panel provides a link to get more details.

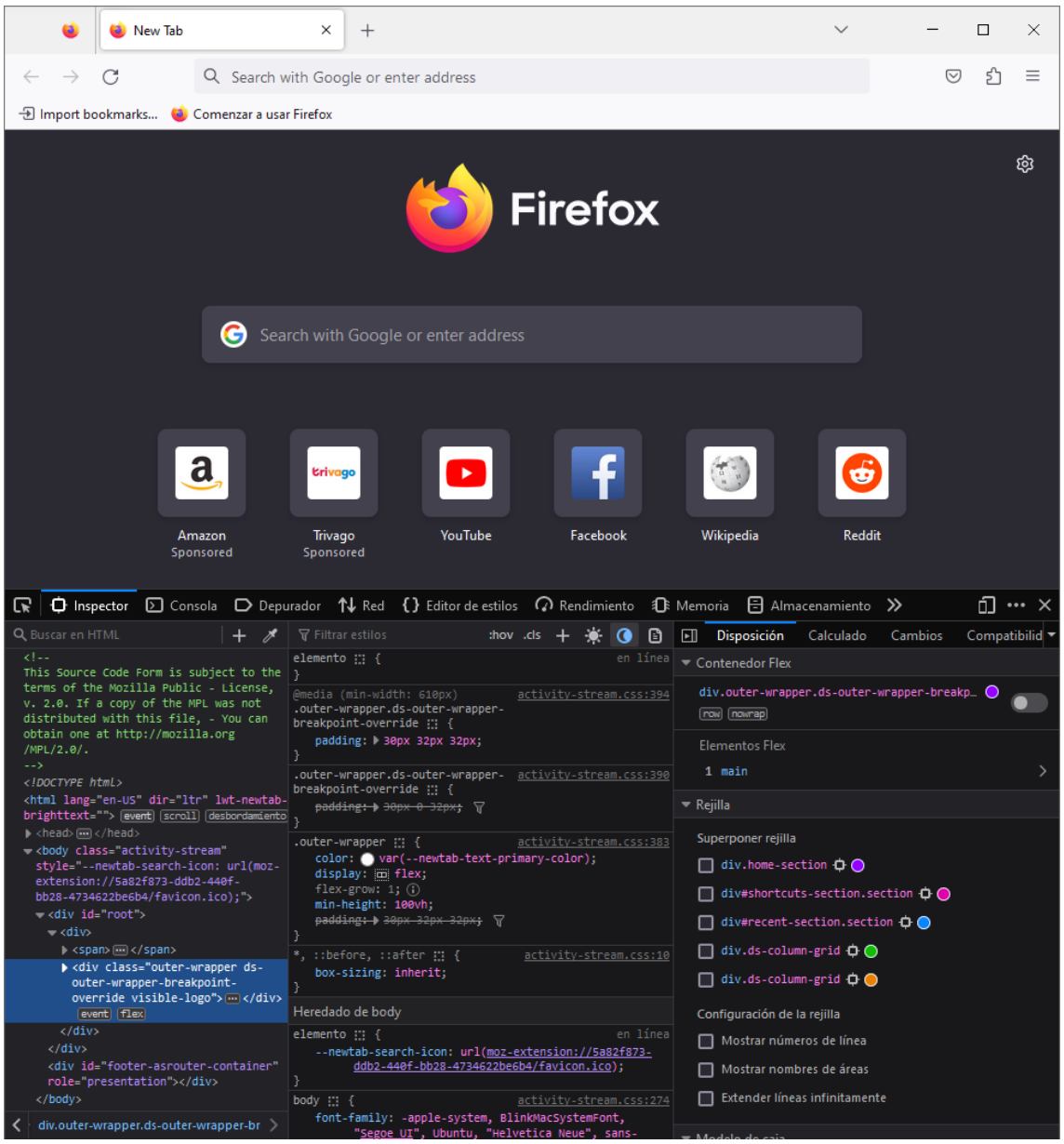


Firefox

The Firefox Developer Tools is a set of web developer tools built into Firefox. With this tool we can examine, debug and edit HTML, CSS and JavaScript.

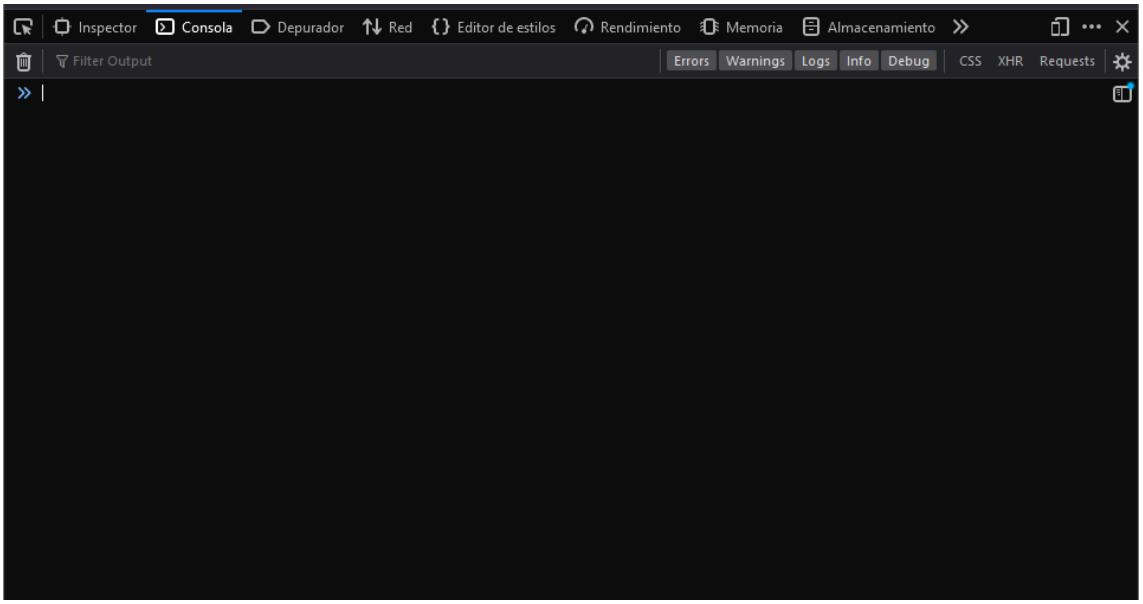
Inspector

The page inspector can view and edit the layout and the content of the page.



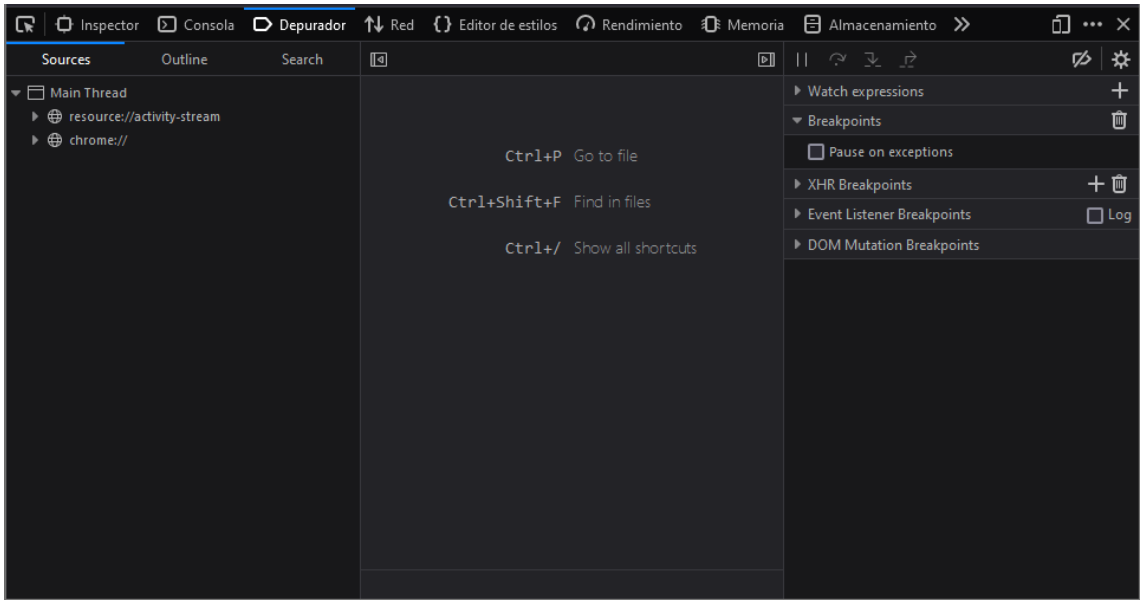
Console

The console allows us to interact with the page using JavaScript.



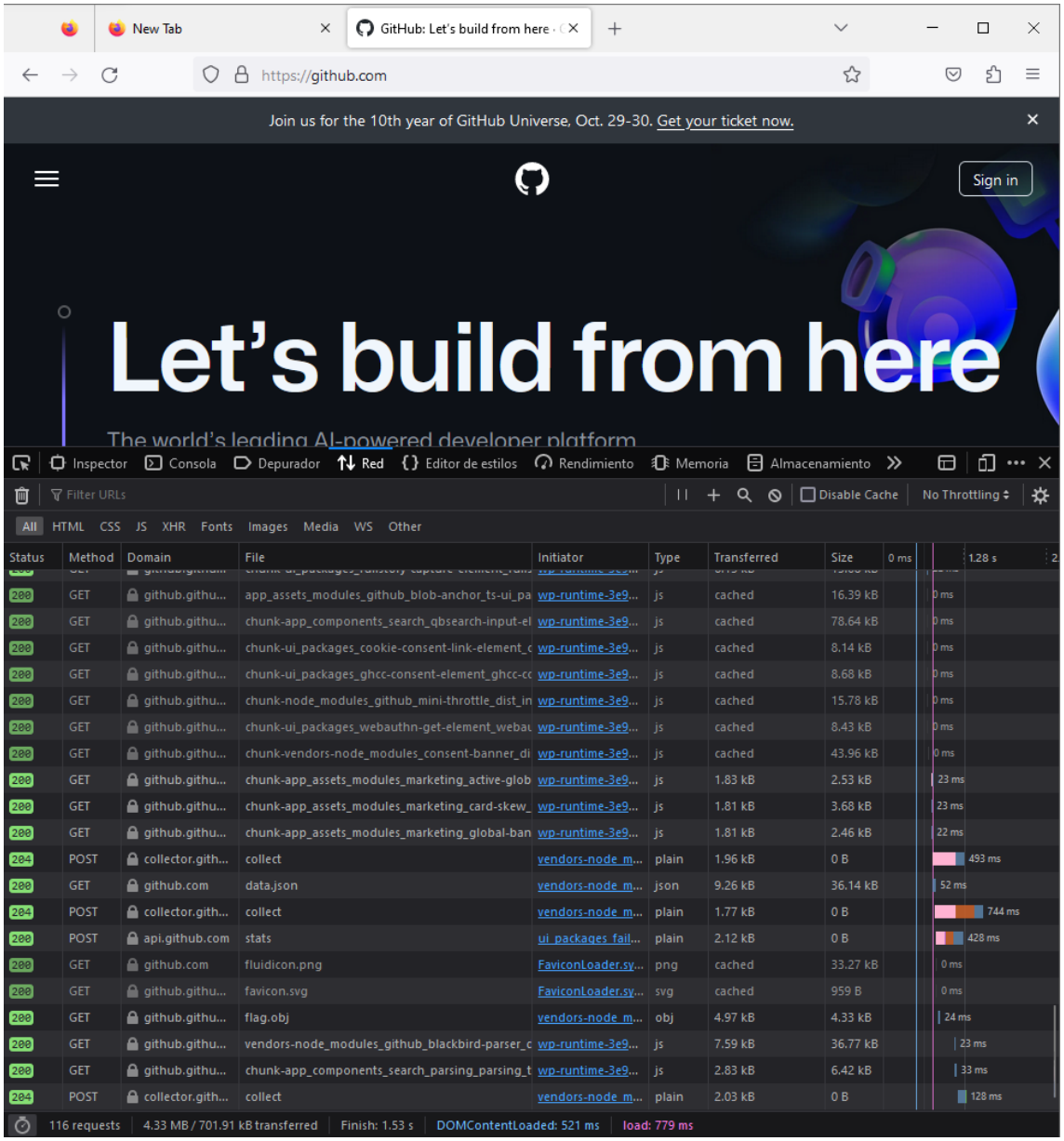
Debugger

The debugger examines the JavaScript running on a page.



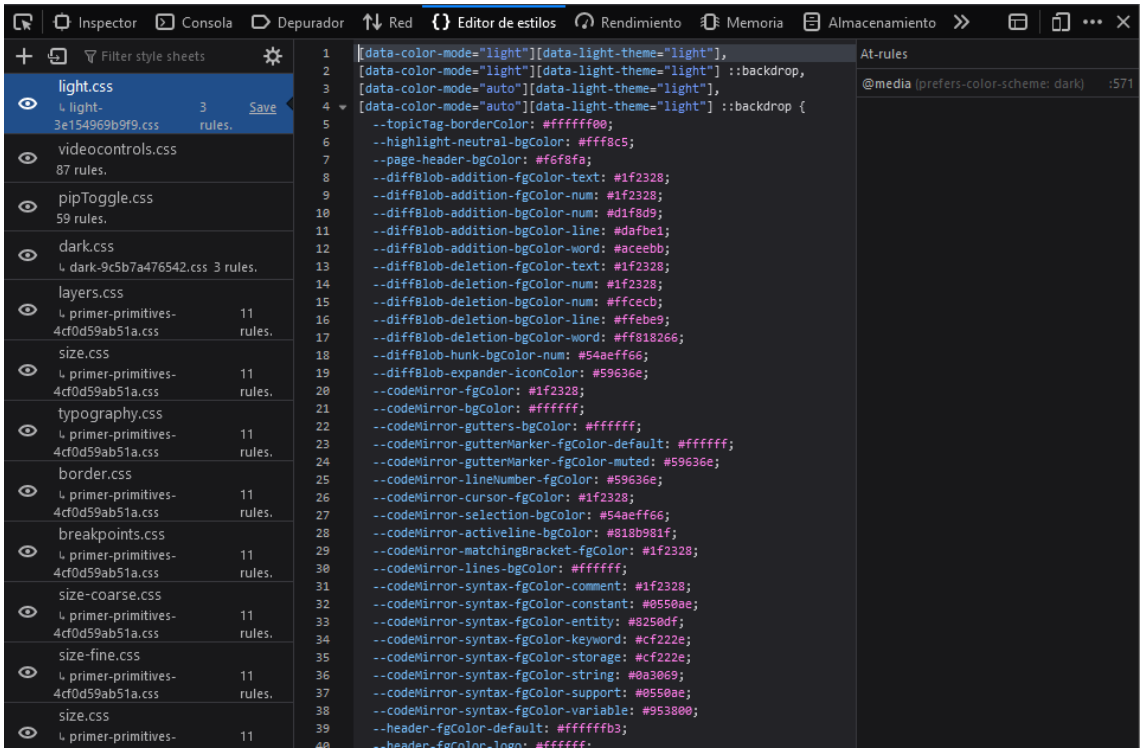
Network

The Network Monitor shows you all the HTTP requests Firefox makes, how long each request takes, and details of each request.



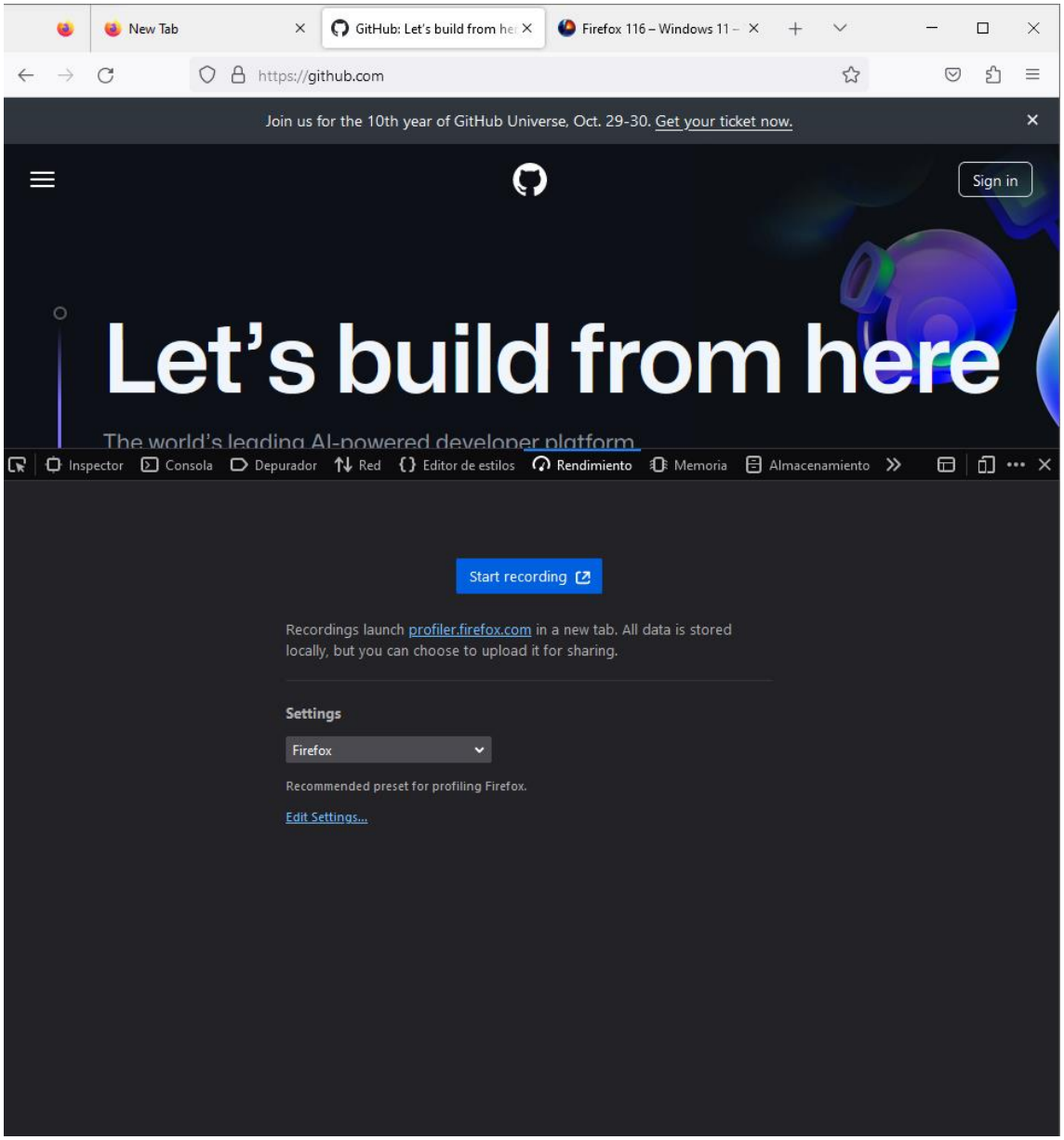
Style Editor

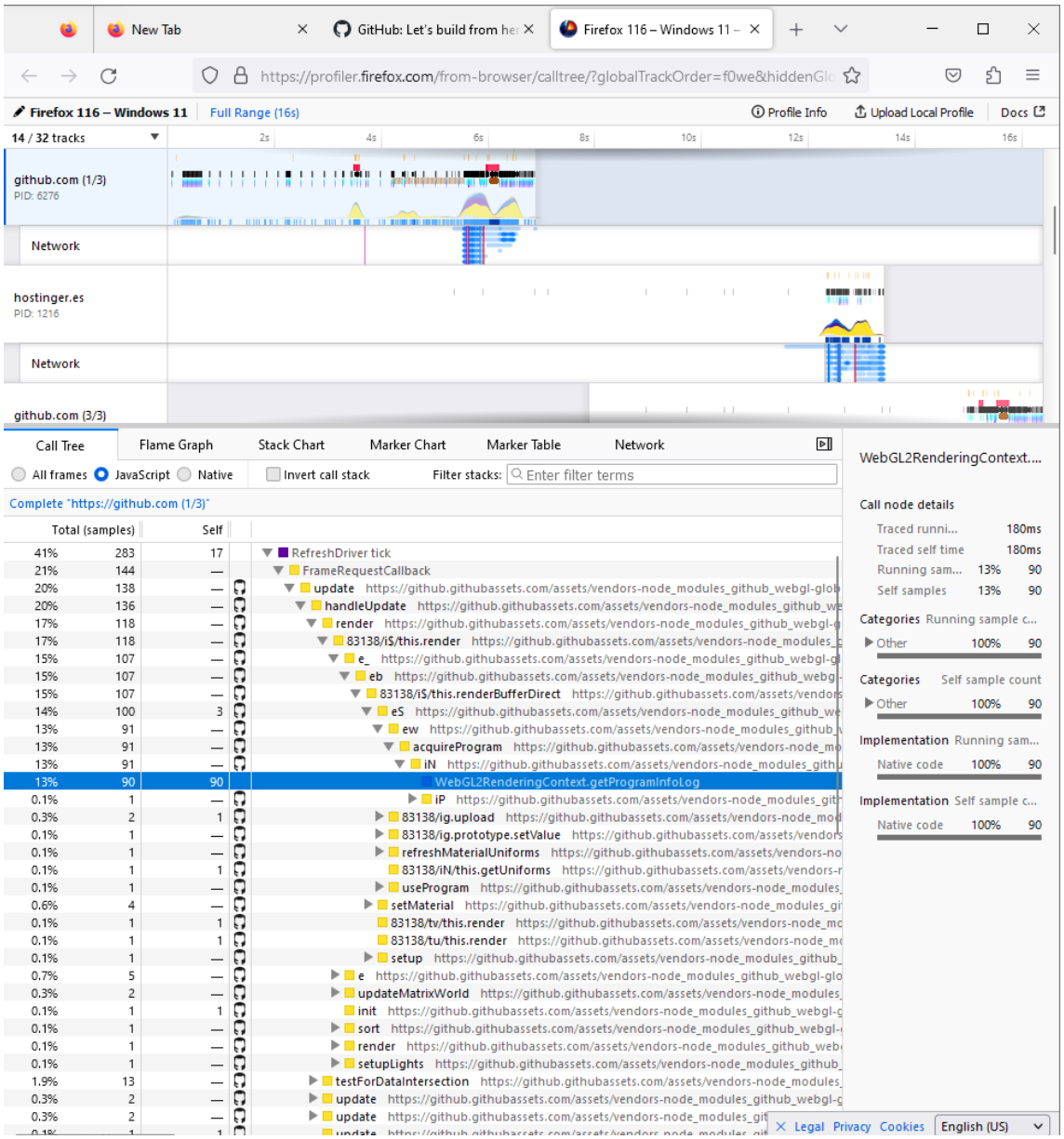
With the style editor you can edit and view all the stylesheets that are associated with a page, create new ones from scratch and apply them to the page, and you can also import existing stylesheets and apply them to the page.



Performance

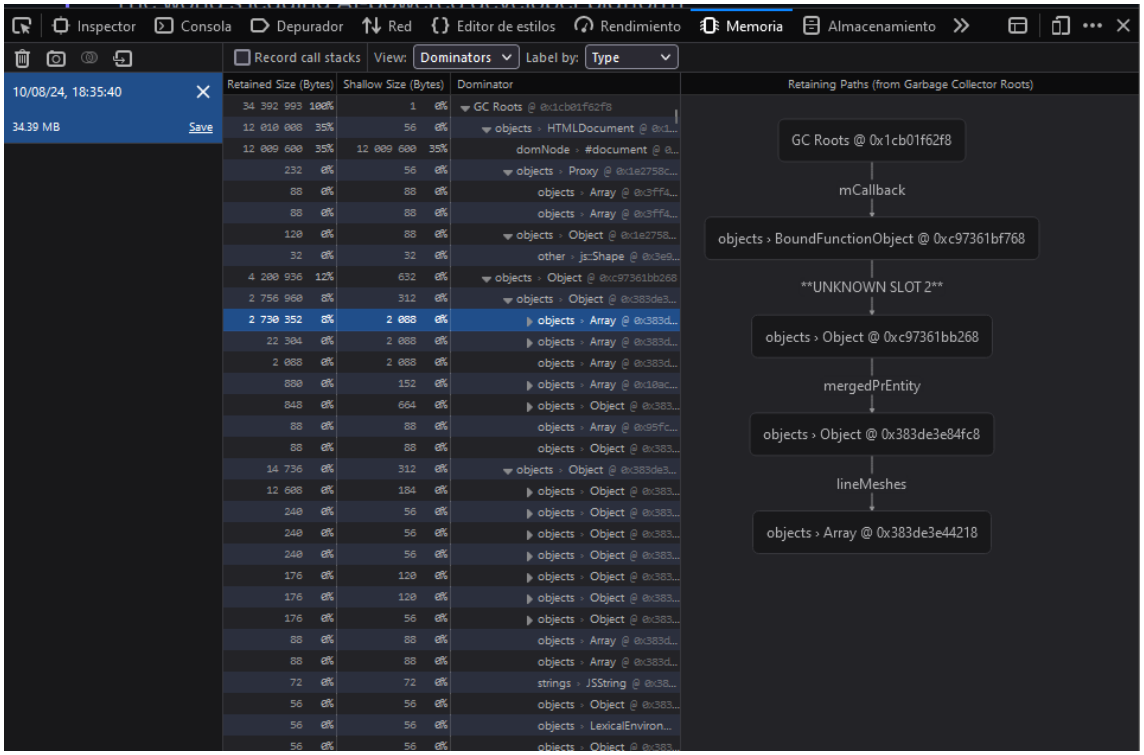
With the performance monitor you can analyze your site's general responsiveness, JavaScript and layout performance.




New Tab
GitHub: Let's build from her
Firefox 116 – Windows 11 –
https://profiler.firefox.com/from-browser/calltree/?globalTrackOrder=f0we&hiddenGlo
Firefox 116 – Windows 11
Full Range (16s)
Profile Info
Upload Local Profile
Docs
14 / 32 tracks
github.com (1/3)
PID: 6276
Network
hostinger.es
PID: 1216
Network
github.com (3/3)
Call Tree
Flame Graph
Stack Chart
Marker Chart
Marker Table
Network
All frames
JavaScript
Native
Invert call stack
Filter stacks:
Enter filter terms
Complete "https://github.com (1/3)"
Total (samples)
Self
RefreshDriver tick
FrameRequestCallback
update
handleUpdate
render
83138/iS/this.renderBufferDirect
eS
ew
acquireProgram
iN
WebGL2RenderingContext.getProgramInfoLog
83138/ig.upload
83138/ig.prototype.setValue
refreshMaterialUniforms
83138/iN/this.getUniforms
useProgram
setMaterial
83138/tv/this.render
83138/tu/this.render
setup
updateMatrixWorld
init
sort
setupLights
testForDataIntersection
WebGL2RenderingContext....
Call node details
Traced runni... 180ms
Traced self time 180ms
Running sam... 13% 90
Self samples 13% 90
Categories Running sample c...
Other 100% 90
Categories Self sample count
Other 100% 90
Implementation Running sam...
Native code 100% 90
Implementation Self sample c...
Native code 100% 90
Legal
Privacy
Cookies
English (US)

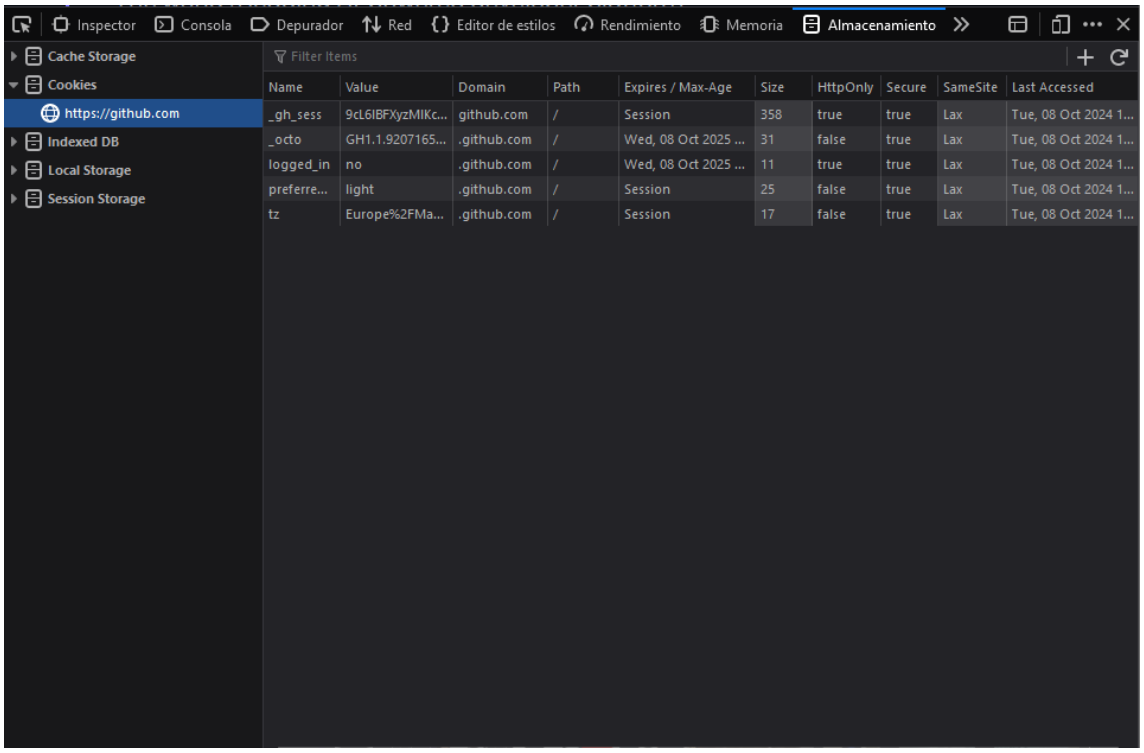
Memory

With this monitor, you can figure out which objects are keeping memory in use.



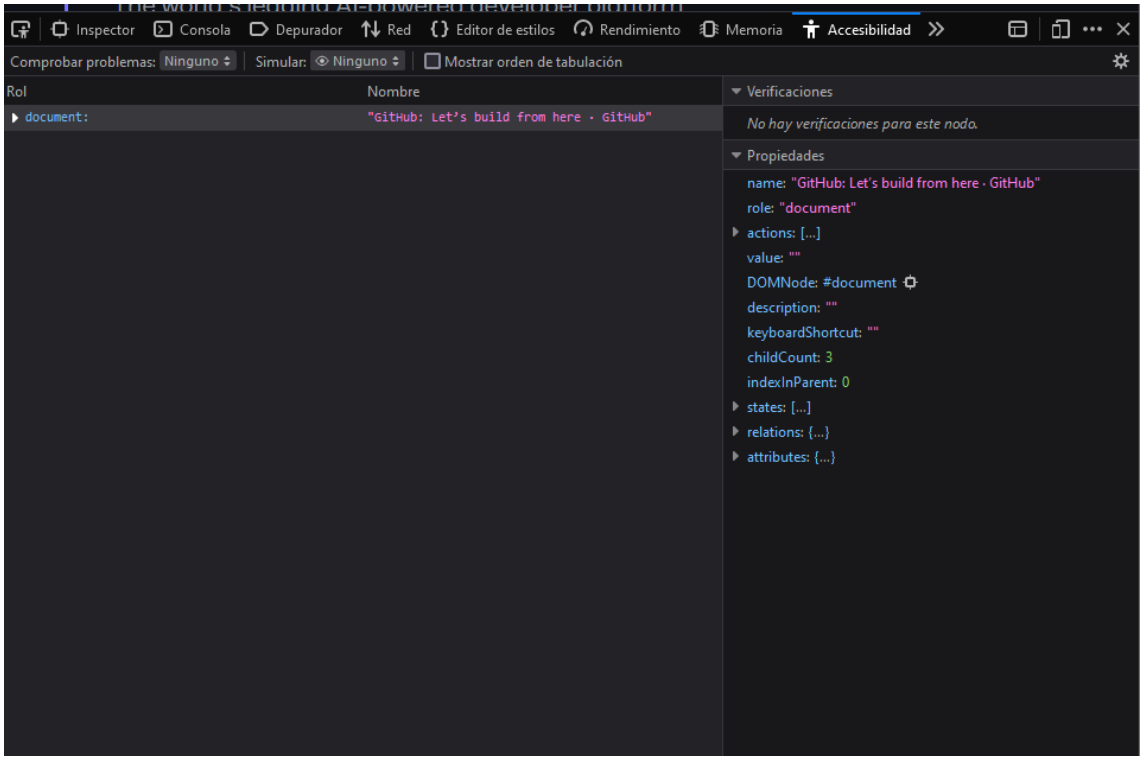
Storage

The Storage Inspector enables you to inspect various types of storage that a web page can use. Currently it can be used to inspect the following storage types: Cache Storage — any DOM caches created using the Cache API. Cookies — All the cookies created by the page or any iframes inside of the page.



Accessibility

The Accessibility Inspector provides a means to access important information exposed to assistive technologies on the current page via the accessibility tree, allowing you to check what's missing or otherwise needs attention.



Application

The application monitor provides tools for inspecting and debugging modern web apps (also known as Progressive Web Apps).