

# Práctica 2

Navegadores y motores de renderizado



## **Ejercicio 01- Navegadores**

### **1. Escoge tres navegadores que no hayas utilizado nunca o los hayas utilizado muy poco.**

Los Navegadores que he escogido, son Opera, Falkon y Midori.

### **2. Instalarlos en tu PC o máquina virtual.**

Los he instalado en un sistema Ubuntu Mate corriendo en una máquina virtual.

### **3. Detalla por cada navegador la siguiente información:**

- 1. Datos interesantes: nombre, historia del nombre, año de creación...**
- 2. Motor de renderizado.**
- 3. Estado actual.**

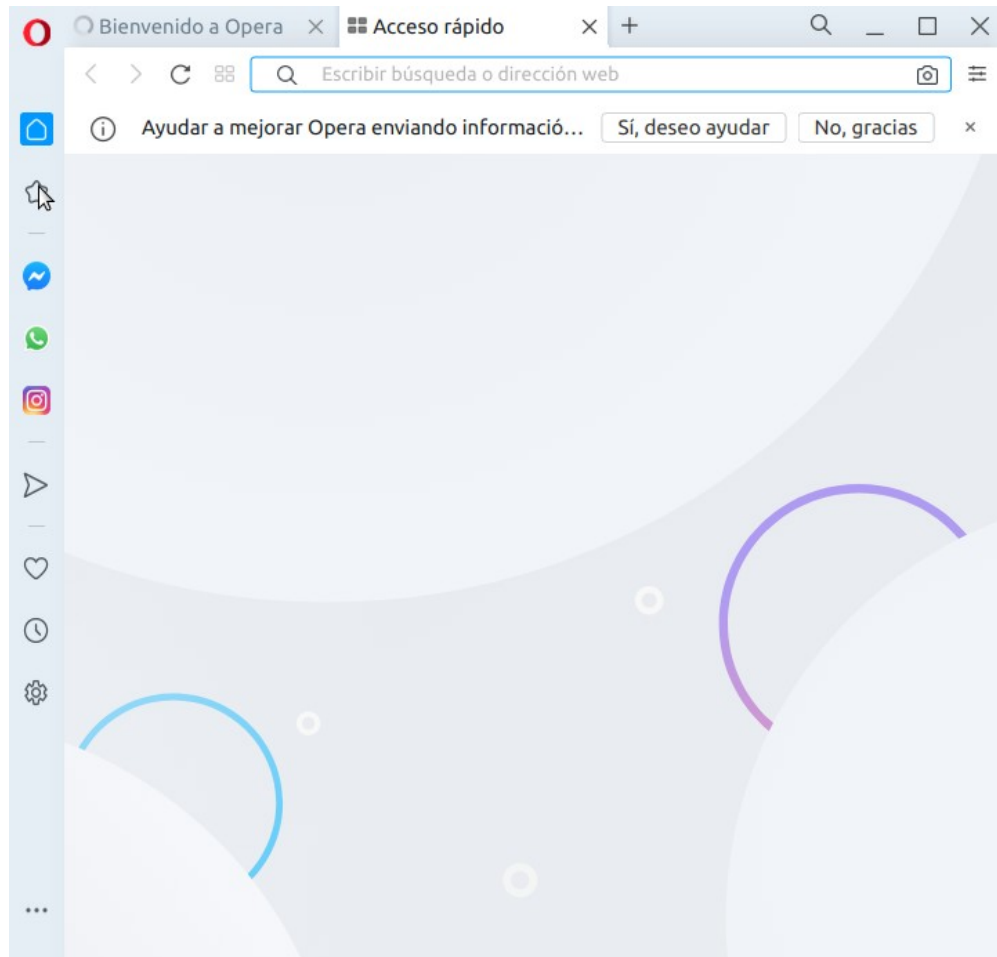
#### **Opera:**

Nació en 1994 como un proyecto de investigación en Telnor, la compañía más grande de telecomunicaciones de Noruega. Fue liberado por primera vez en publico en 1996 en su versión 2.0, era de pago hasta su versión 5.0 liberada en el 2000 aunque mostraba muchos anuncios, pero en el 2005 con la versión 8.5, fue liberado completamente .

Utiliza el motor de renderizado Chromium desarrollado por Google, este motor introduce mejores búsquedas, organización y complementos. En anteriores versiones utilizaba el motor Webkit desarrollado por Apple.

Actualmente el navegador esta a las bajas, desde que cambió a Webkit se temía lo peor, ya que había abandonado su propio motor, pero consiguió seguir adelante. Hoy en día no es muy utilizado ya que el mercado esta dominado por Firefox y Chrome.

## Interfaz de Opera:



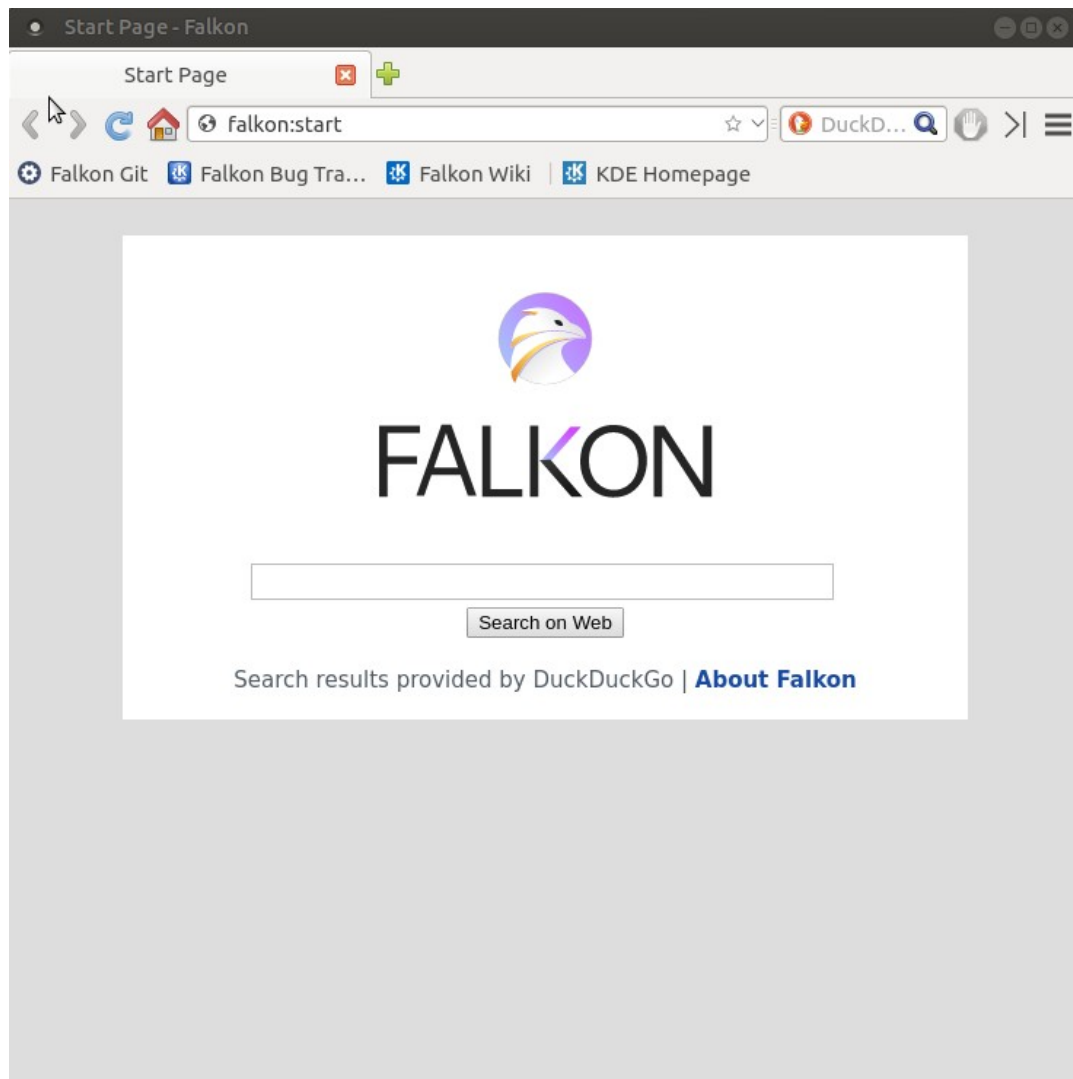
## Falkon:

Creado por David Rosca y actualmente desarrollado por KDE. Anteriormente conocido como QupZilla, nació como un proyecto de investigación en 2010, estaba escrito en Python, pero en 2011 se volvió a escribir en C++. Es muy parecido a Chrome, pero tiene muchas diferencias como Adblock preinstalado, Download Manager y otros trucos.

Utiliza el motor de renderizado QtWebEngine, una implementación de Blink que integra Chromium en los ambientes QT (Framework multiplataforma orientado a Objetos).

Actualmente no es muy utilizado, he buscado información al respecto y no aparece nada más allá del 2018 donde su repercusión no fue alta.

## Interfaz de Falkon:



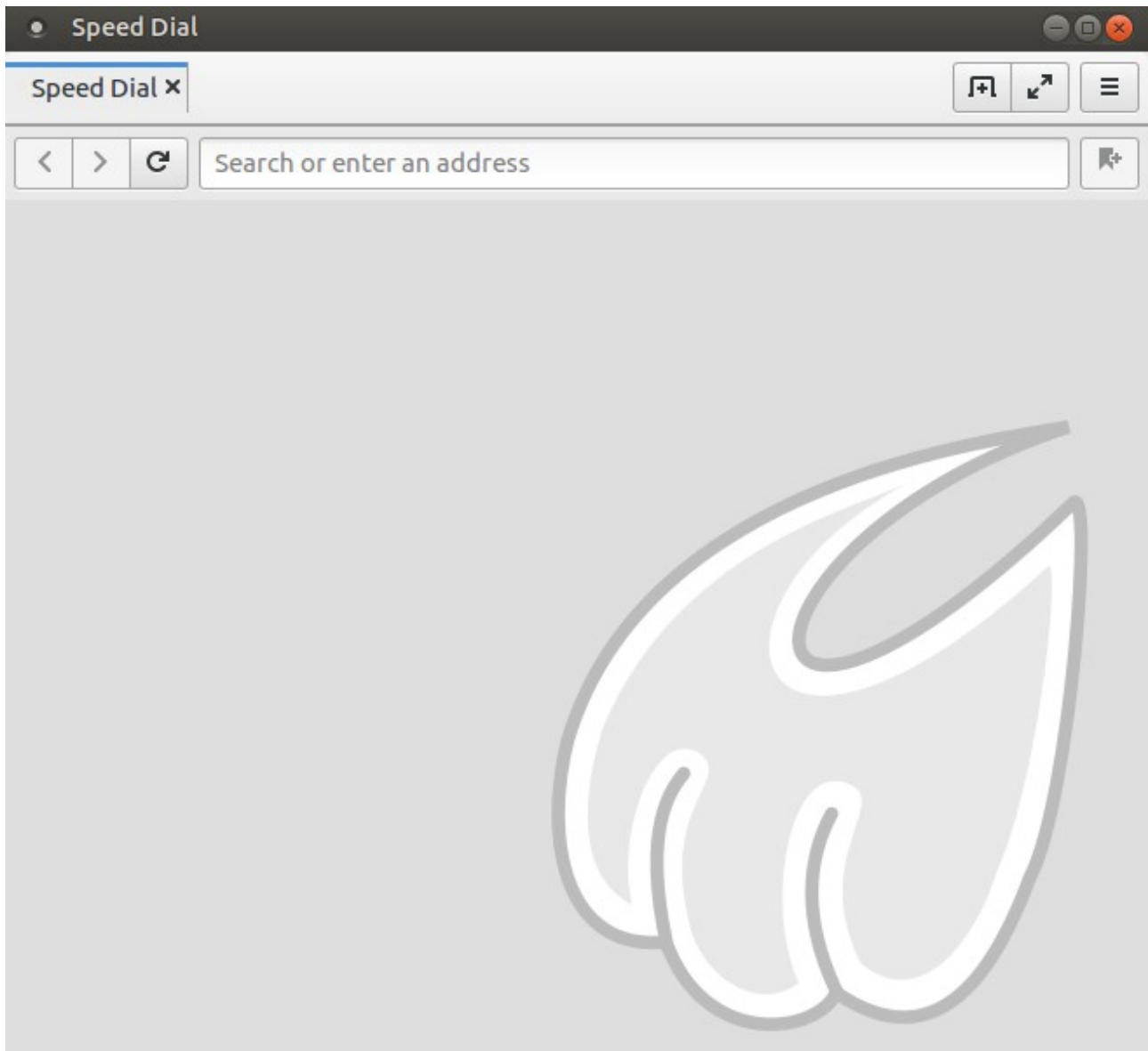
## Midori:

Creado por Christian Dywan, Nancy Runge y Astian Foundation en 2007, se caracteriza por ser muy liviano. Es open source y su núcleo está escrito en Vala que es un lenguaje que intenta acercar a C las características de los lenguajes modernos sin requisitos adicionales en tiempo de ejecución.

Utiliza el motor de renderizado WebKit conocido por su eficacia en el navegador Safari de Mac OS X. Logra un 100/100 en la prueba acid3 que es un sitio hecho por WaSP que pone a prueba los navegadores con los estándares web, especialmente los de DOM y ECMAScript.

Actualmente carece de mantenimiento en Windows y Mac OS, pero han incorporado soporte para Android y han continuado su mantenimiento en Linux. Su versión actual es la 8.

### Interfaz de Midori:



## **Ejercicio 02- Motores de renderizado**

### **1. Escoge dos motores de renderizado que no conozcas.**

Los motores que he elegido son KHTML y Servo.

### **2. Detallar para cada uno la siguiente información:**

- 1. Datos interesantes: nombre, historia del nombre, año de creación...**
- 2. En que navegadores se esta utilizando o se ha utilizado.**
- 3. Estado actual.**

#### **KHTML:**

Fue desarrollado por KDE en 2014, esta escrito en C++ y distribuido bajo licencia GNU.

Inicialmente utilizado por Konqueror aunque a partir de 2016, los motores descendientes de KHTML como WebKit son utilizados por los navegadores Google Chrome, Safari y Opera entre otros.

En la actualidad, se encuentra en estado discontinuado.

#### **Servo:**

Lanzado en 2012 por la Fundación Mozilla bajo el apoyo de Samsung, está escrito en Rust. Se enfoca en una mayor seguridad y fluidez en arquitecturas de alta gama, es de código abierto.

En 2016 se empezó a incorporar junto con Gecko, el anterior motor de Firefox, pero en la actualidad ya lo utiliza plenamente. Solo Firefox utiliza este motor.

Actualmente se cuenta en estado activo.

### **3. Realiza un estudio sobre los diferentes motores de renderizado utilizados por Google Chrome a lo largo de su historia.**

Chrome ha utilizado los motores WebKit y Blink.

#### **WebKit:**

Lo utilizó desde su lanzamiento en 2008, desarrollado por Apple y esta basado en KHTML. Este motor es utilizado en un sin fin de navegadores tales como Safari, Epiphany, Midori, QupZilla, Arora y un largo etcétera.

## Blink:

Desarrollado por Google, anunciado en abril de 2013, nace de una bifurcación del proyecto WebKit y a partir de su lanzamiento es utilizado por Chrome, Chromium, Opera, Brave, Vivaldi, Maxthon y Microsoft Edge. Implementa mejoras en la velocidad de cargar contenido en el DOM en un 90%, incorpora nuevas notificaciones enriquecidas con imágenes, listado de texto y respuesta directa.

## Ejercicio 03- Comparativa entre navegadores

**Localiza en Internet un artículo en el que se realice una comparación de los navegadores actuales. Para seleccionar el artículo considera utilizar los parámetros de búsqueda de Google, donde podrás seleccionar que los resultados los clasifique por orden de aparición cronológicamente. Comenta que te ha parecido el artículo.**

Me voy a basar en este artículo:








<https://www.mozilla.org/es-CL/firefox/browsers/compare/>

Aunque el artículo este escrito en la página de Mozilla, me ha gustado especialmente porque analiza objetivamente los aspectos mas relevantes de los navegadores y en base a eso determina cual es el más completo.

La comparativa esta echa entre Chrome, Firefox, Safari, Opera, Brave (Me encanta su versión móvil) , Internet Explorer y Edge.








El primer aspecto que se compara es la seguridad y privacidad, considero que es uno de los puntos más importantes que cualquier usuario/a tendría que considerar.

Como se puede ver en la siguiente gráfica Firefox, Brave y Edge son los que ganan la partida ofreciendo bloqueo a los scripts de los criptomineros y los rastreadores sociales. Esto se traduce en una mayor velocidad.

| Seguridad y privacidad                     |  |  |  |  |  |  |  |
|--|---|---|--|---|---|---|---|
| Modo de navegación privada                 | ✓   | ✓   | ✓  | ✓   | ✓   | ✓   | ✓   |
| Bloquea las cookies de rastreo de terceros | ✓   | ✓   | ✓  | ✓   | ✓   | ✓   | ✓   |
| Bloquea los scripts de criptomineros       | ✓   | —   | ✓  | —   | ✓   | ✓   | —   |
| Bloquea los rastreadores sociales          | ✓   | —   | ✓  | ✓   | —   | ✓   | —   |








En segundo lugar se compara la utilidad, que en resumen son las funcionalidades que puede hacer cada navegador.

En este apartado, el ganador claramente es Firefox, que realiza la totalidad de funcionalidades que puede ofrecer un navegador.

| Utilidad   |  |  |  |  |  |  |  |
|--|---|---|--|---|---|---|---|
| Bloqueo de reproducción automática                 | ✓   | —   | —  | ✓   | —   | ✓   | —   |
| Navegación con pestañas                            | ✓   | ✓   | ✓  | ✓   | ✓   | ✓   | ✓   |
| Administrador de marcadores                        | ✓   | ✓   | ✓  | ✓   | ✓   | ✓   | ✓   |
| Llena formularios automáticamente                  | ✓   | ✓   | ✓  | ✓   | ✓   | ✓   | ✓   |
| Opciones de motor de búsqueda                      | ✓   | ✓   | ✓  | ✓   | ✓   | ✓   | ✓   |
| Texto a voz  | ✓   | —   | ✓  | ✓   | —   | —   | ✓   |
| Modo de lectura                                    | ✓   | ✓   | ✓  | ✓   | —   | ✓   | ✓   |
| Corrección ortográfica                             | ✓   | ✓   | ✓  | ✓   | ✓   | ✓   | ✓   |
| Extensiones web/Complementos                       | ✓   | ✓   | ✓  | ✓   | ✓   | ✓   | ✓   |
| Herramienta de captura de pantalla en el navegador | ✓   | ✓   | —  | —   | ✓   | —   | —   |

Y por último realiza una comparativa basándose en la portabilidad que es otro de los aspectos mas demandados hoy en día.

En esta ocasión, son los navegadores Firefox y Opera los que salen vencedores.

| Portabilidad                                    |  |  |  |  |  |  |  |
|---|---|---|--|---|---|---|---|
| Disponibilidad para sistemas operativos         | ✓   | ✓   | —  | —   | ✓   | ✓   | —   |
| Disponibilidad para sistemas operativos móviles | ✓   | ✓   | ✓  | —   | ✓   | ✓   | —   |
| Sincronización con el móvil                     | ✓   | ✓   | ✓  | ✓   | ✓   | ✓   | —   |
| Administración de contraseñas                   | ✓   | ✓   | ✓  | ✓   | ✓   | ✓   | ✓   |
| Contraseña maestra                              | ✓   | —   | ✓  | —   | ✓   | —   | —   |

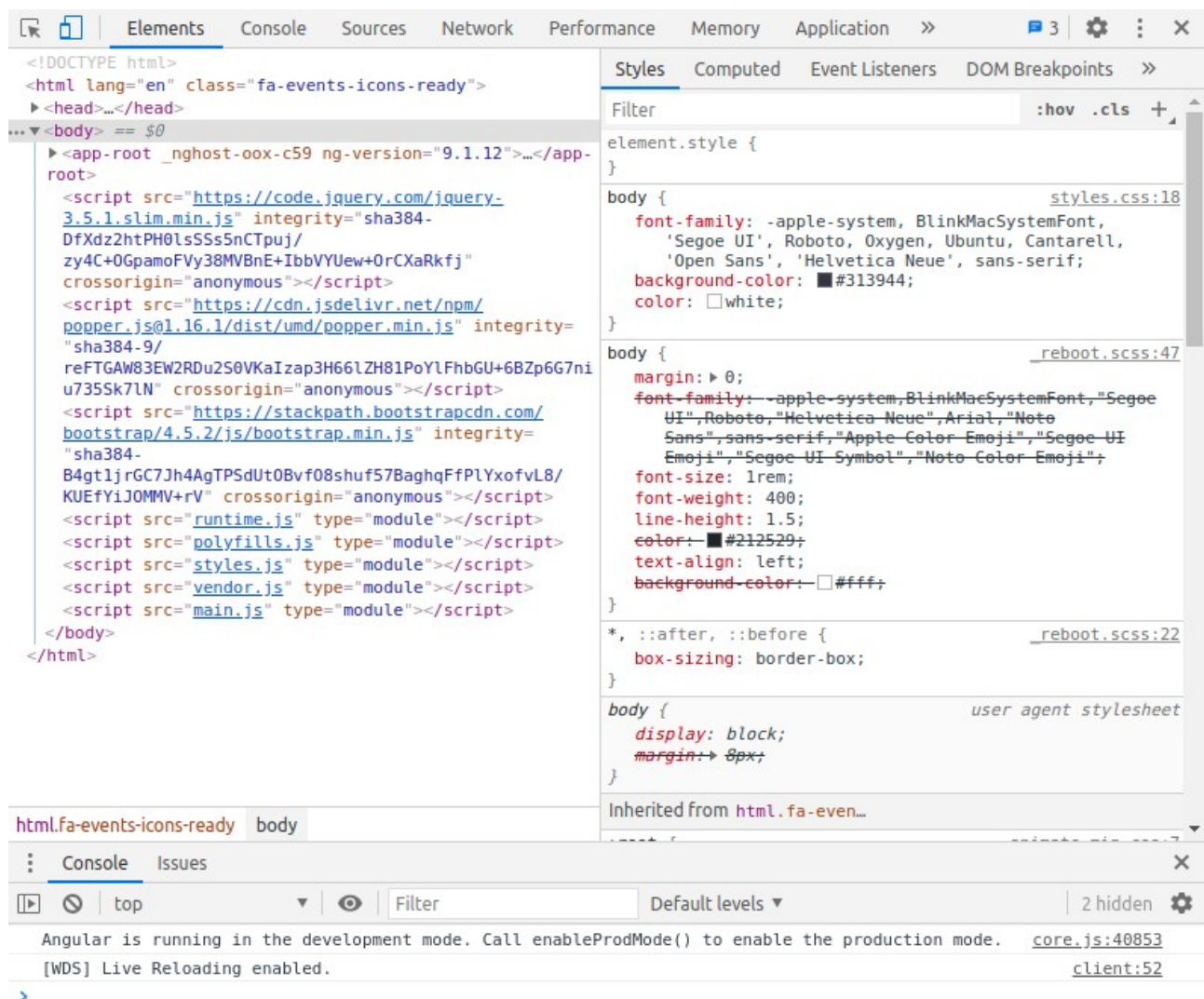


El artículo determina que Firefox es el navegador con mayores funcionalidades, seguridad y portabilidad. En mi opinión personal Brave me ha gustado especialmente en móviles, aunque actualmente esta experimentando algunos bugs.

## Ejercicio 04- Consola de Google Chrome

Elabora una pequeña guía de uso de las diferentes opciones de la consola de Google Chrome.

La consola de los navegadores es una herramienta imprescindible que todo desarrollador web debe conocer.



En ella se encuentran muchas utilidades para ayudar a los desarrolladores a la hora de programar y localizar errores en el código.

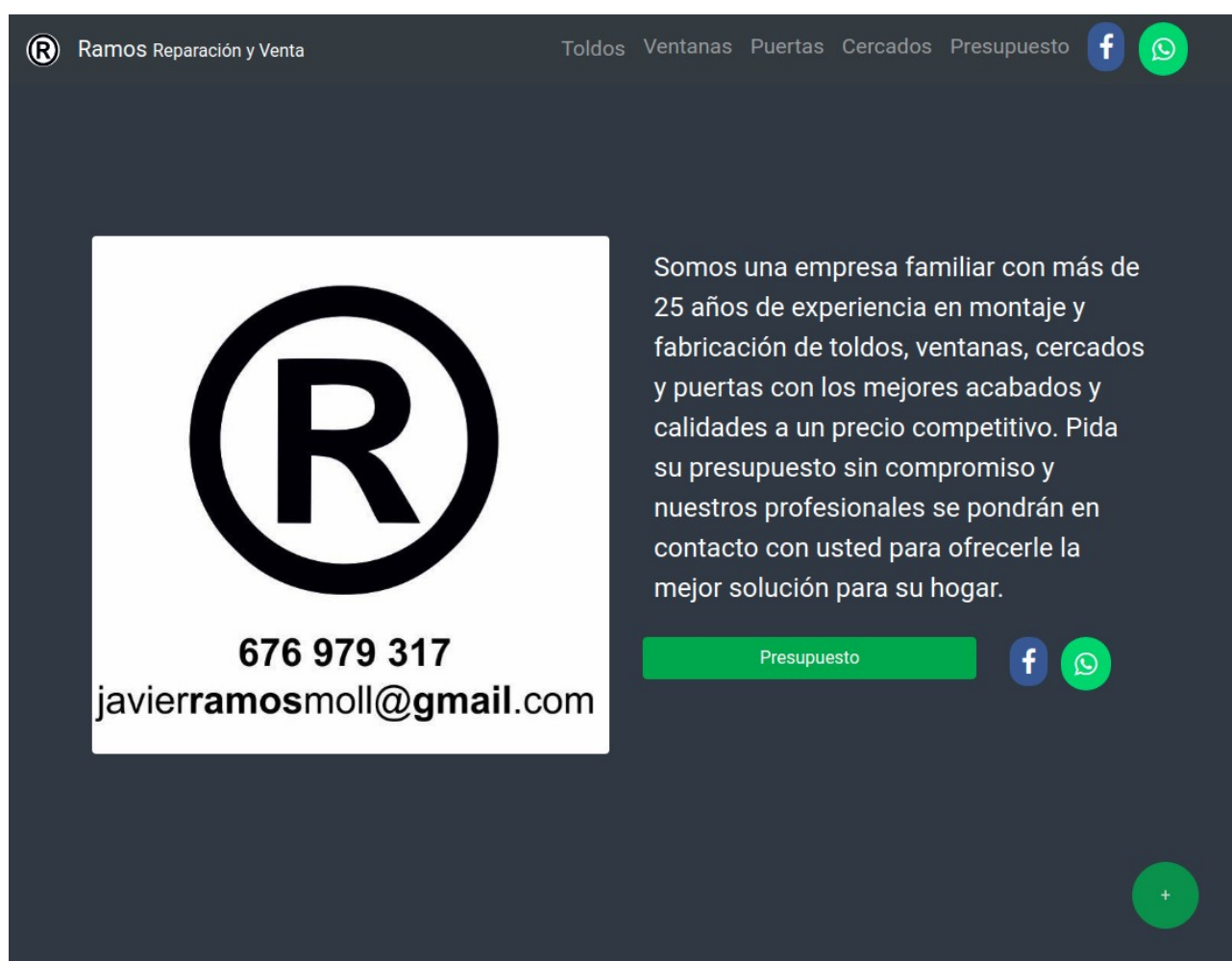
**Inspector de elementos:** 

Al pulsarlo, si posicionamos el ratón encima de cualquier elemento de la página, podremos ver la información (html, css) de ese componente en concreto.

**Toogle Divece:** 

Esta opción sirve para alternar entre la vista de escritorio o de dispositivo móvil. Es muy utilizado para ver el comportamiento responsive de la página.

Vista Escritorio:



Vista Móvil:



### Elements:

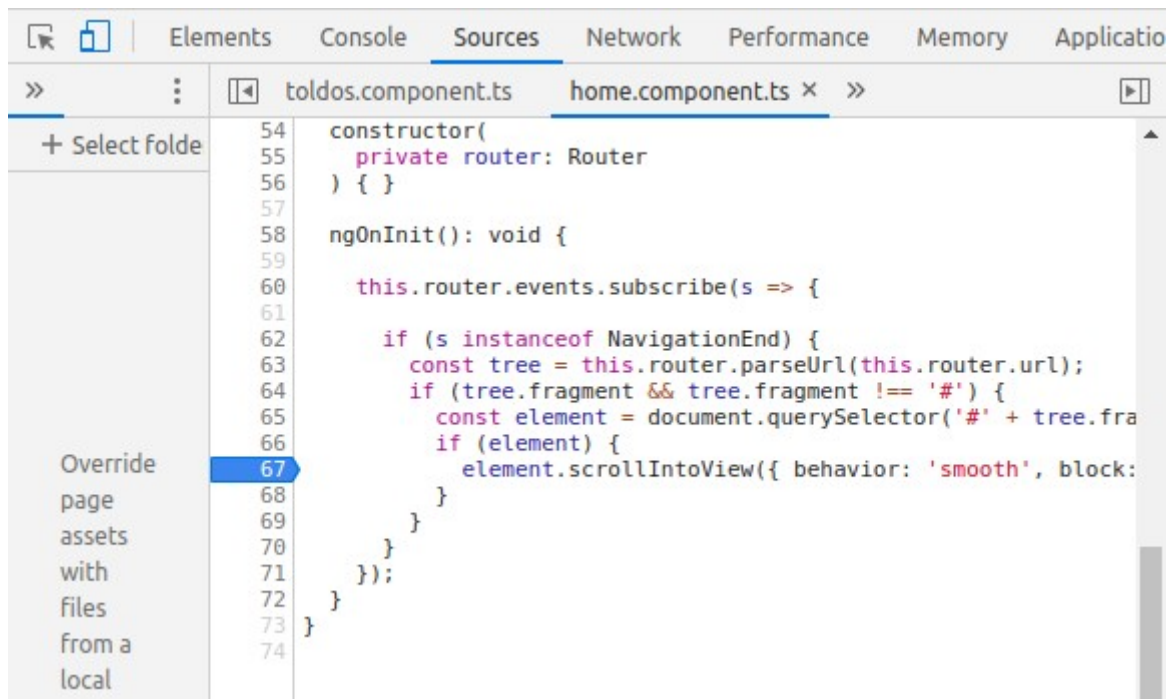
Permite la visualización del código html de la página.

### Console:

Permite escribir código JavaScript directamente en ella, se utiliza para ver mensajes de error y mensajes "Log" que son escritos por el programador desde el código fuente y ayuda en la depuración y detección de errores.

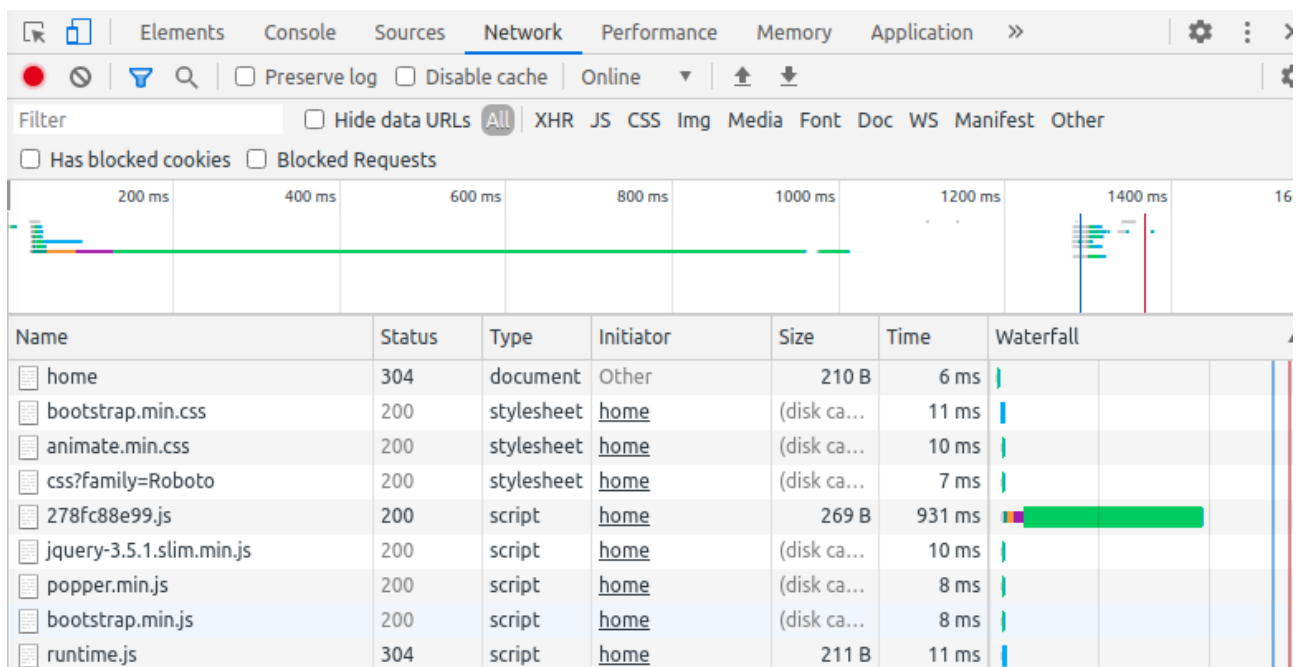
## Sources:

Permite la visualización del código fuente y realizar “Break Points” que nos ayudan a la hora de detectar errores en el código.



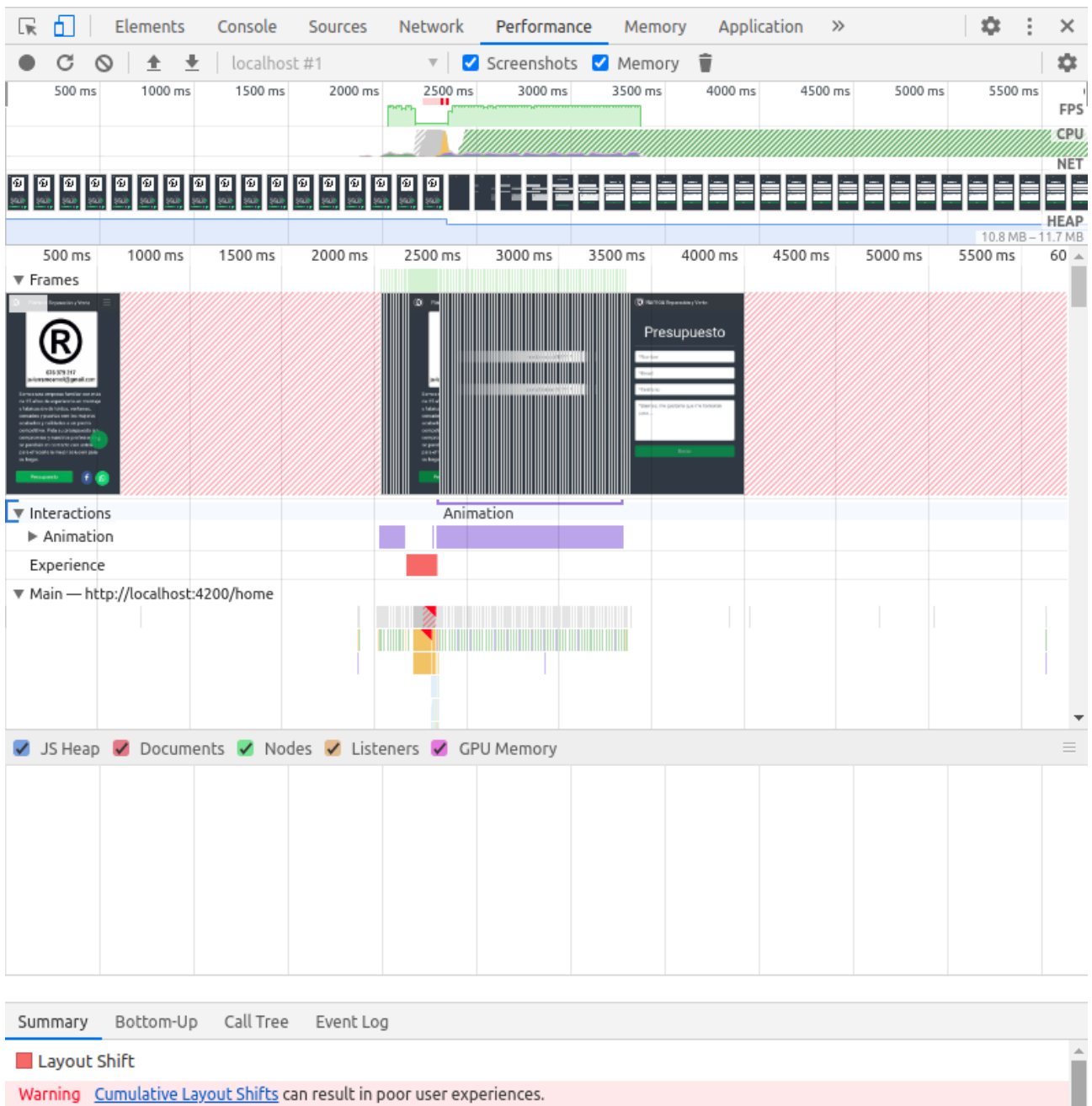
## Network:

Nos indica el uso de red y el tiempo que utilizado para cargarse los elementos.



## Performance:

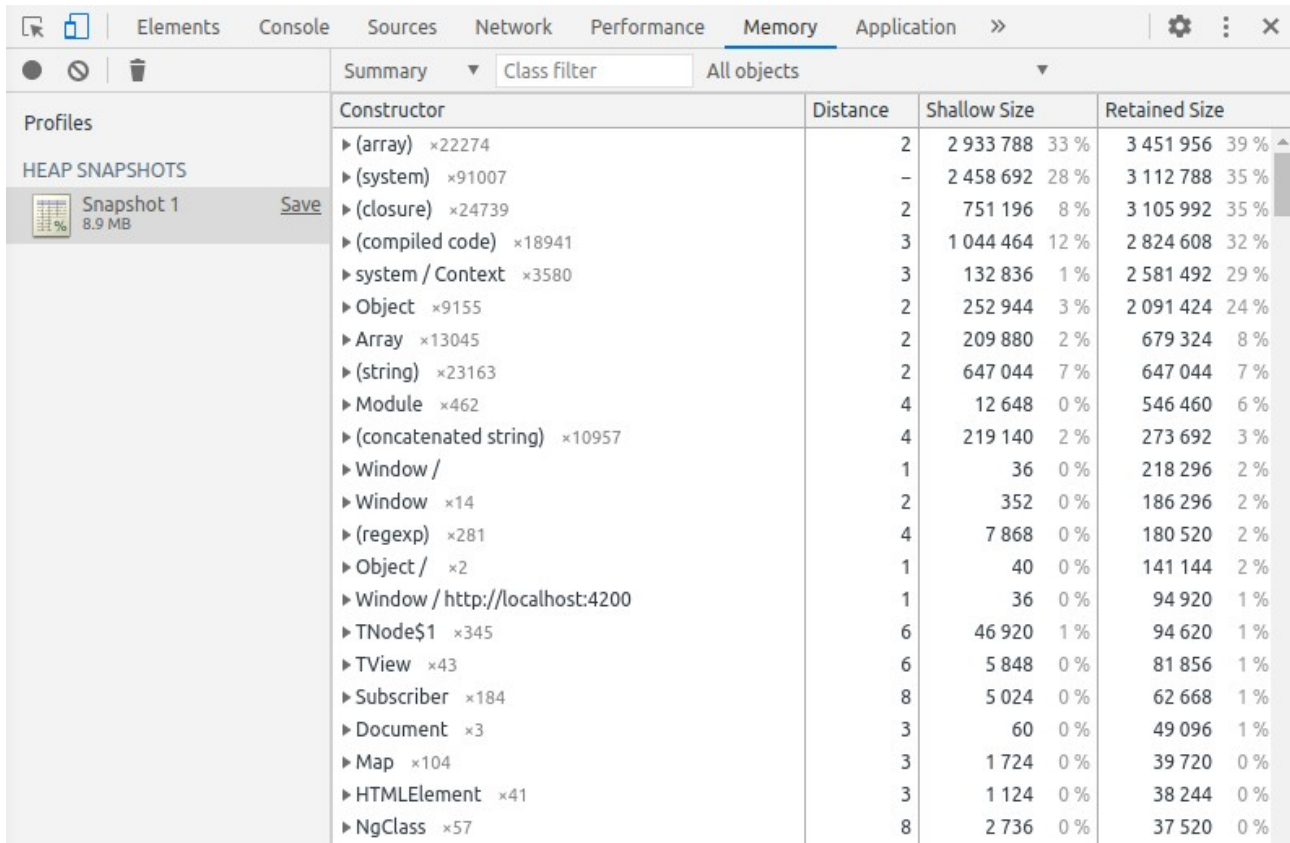
Realiza un test del rendimiento de nuestra aplicación y nos indica si la experiencia de usuario es correcta.





## Memory:

Nos indica el uso de la memoria de los elementos de la aplicación.

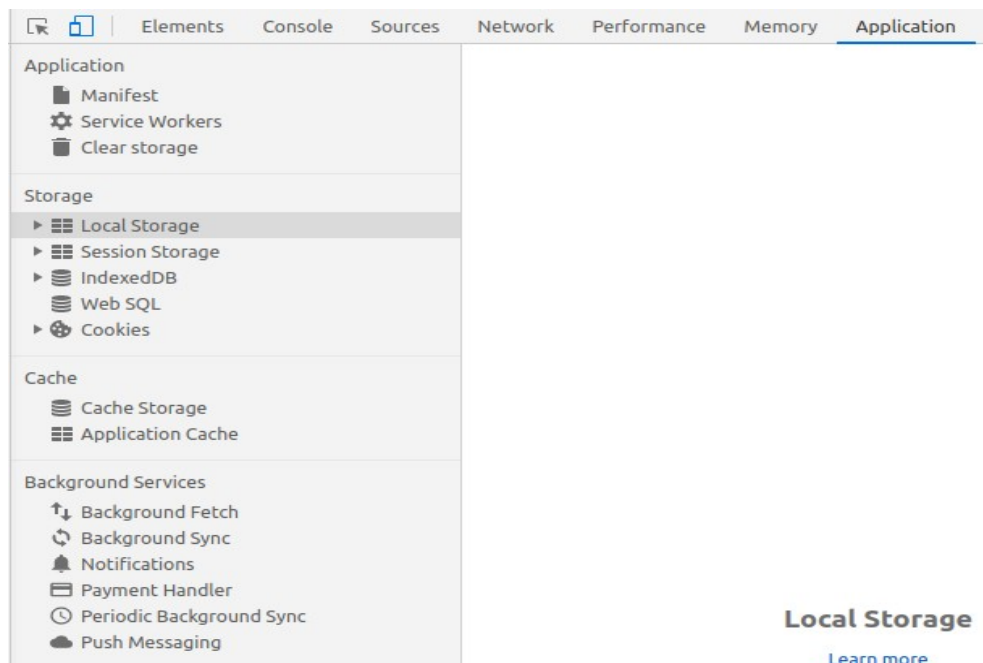


The screenshot shows the Chrome DevTools Memory tab. The left sidebar displays 'HEAP SNAPSHOTS' with 'Snapshot 1' (8.9 MB) selected. The main panel shows a table of memory objects.

| Constructor                      | Distance | Shallow Size   | Retained Size  |
|----------------------------------|----------|----------------|----------------|
| ▶ (array) ×22274                 | 2        | 2 933 788 33 % | 3 451 956 39 % |
| ▶ (system) ×91007                | -        | 2 458 692 28 % | 3 112 788 35 % |
| ▶ (closure) ×24739               | 2        | 751 196 8 %    | 3 105 992 35 % |
| ▶ (compiled code) ×18941         | 3        | 1 044 464 12 % | 2 824 608 32 % |
| ▶ system / Context ×3580         | 3        | 132 836 1 %    | 2 581 492 29 % |
| ▶ Object ×9155                   | 2        | 252 944 3 %    | 2 091 424 24 % |
| ▶ Array ×13045                   | 2        | 209 880 2 %    | 679 324 8 %    |
| ▶ (string) ×23163                | 2        | 647 044 7 %    | 647 044 7 %    |
| ▶ Module ×462                    | 4        | 12 648 0 %     | 546 460 6 %    |
| ▶ (concatenated string) ×10957   | 4        | 219 140 2 %    | 273 692 3 %    |
| ▶ Window /                       | 1        | 36 0 %         | 218 296 2 %    |
| ▶ Window ×14                     | 2        | 352 0 %        | 186 296 2 %    |
| ▶ (regexp) ×281                  | 4        | 7 868 0 %      | 180 520 2 %    |
| ▶ Object / ×2                    | 1        | 40 0 %         | 141 144 2 %    |
| ▶ Window / http://localhost:4200 | 1        | 36 0 %         | 94 920 1 %     |
| ▶ TNode\$1 ×345                  | 6        | 46 920 1 %     | 94 620 1 %     |
| ▶ TView ×43                      | 6        | 5 848 0 %      | 81 856 1 %     |
| ▶ Subscriber ×184                | 8        | 5 024 0 %      | 62 668 1 %     |
| ▶ Document ×3                    | 3        | 60 0 %         | 49 096 1 %     |
| ▶ Map ×104                       | 3        | 1 724 0 %      | 39 720 0 %     |
| ▶ HTMLElement ×41                | 3        | 1 124 0 %      | 38 244 0 %     |
| ▶ NgClass ×57                    | 8        | 2 736 0 %      | 37 520 0 %     |

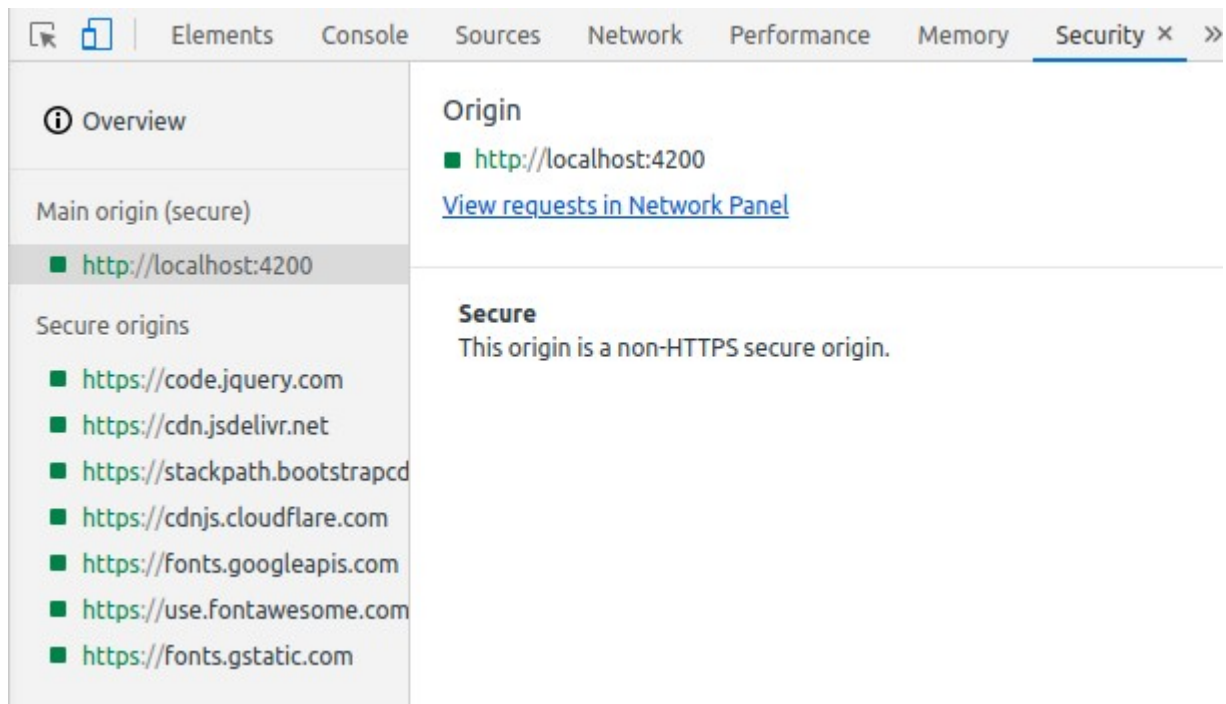
## Application:

Permite ver el Local Storage, las Cookies, el cache, el application cache y los Background Services.



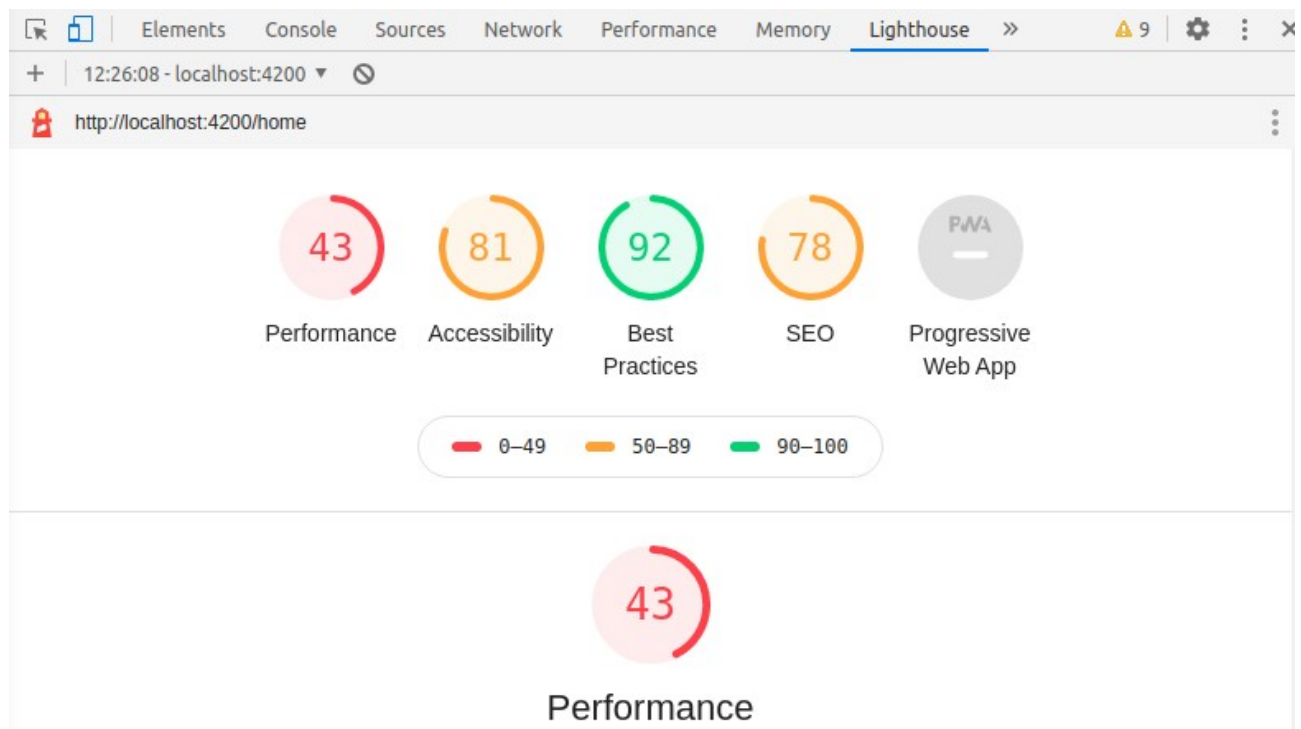
## Security:

Nos indica el tipo de conexión y el certificado de seguridad. Evalúa si el origen es seguro.



## Lighthouse:

Realiza un test que mide la Performance, la Accesibilidad, las Buenas prácticas y el SEO.



## Metrics



|                            |       |                           |       |
|----------------------------|-------|---------------------------|-------|
| ▲ First Contentful Paint   | 4.3 s | ■ Time to Interactive     | 4.3 s |
| ▲ Speed Index              | 4.3 s | ● Total Blocking Time     | 0 ms  |
| ▲ Largest Contentful Paint | 4.5 s | ● Cumulative Layout Shift | 0.015 |

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

[View Trace](#)



**Opportunities** — These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

| Opportunity                | Estimated Savings    |
|----------------------------|----------------------|
| ▲ Enable text compression  | <div></div> 3.52 s ▼ |
| ▲ Minify JavaScript        | <div></div> 1.68 s ▼ |
| ▲ Remove unused JavaScript | <div></div> 1.32 s ▼ |