



4.3 Análisis de uso de Memoria

¡BIENVENIDOS!

Soy Juan Diego Pérez Jiménez

Profesor de Informática en FP



<https://www.linkedin.com/in/juandiegoperez>



@pekechis

ÍNDICE

- Problemas a detectar.
- Herramientas para el análisis.
- Administrador de Tareas de Chrome.
- El panel Memory. Profiling.
- Conceptos sobre Memoria.
- Visualización de problemas.

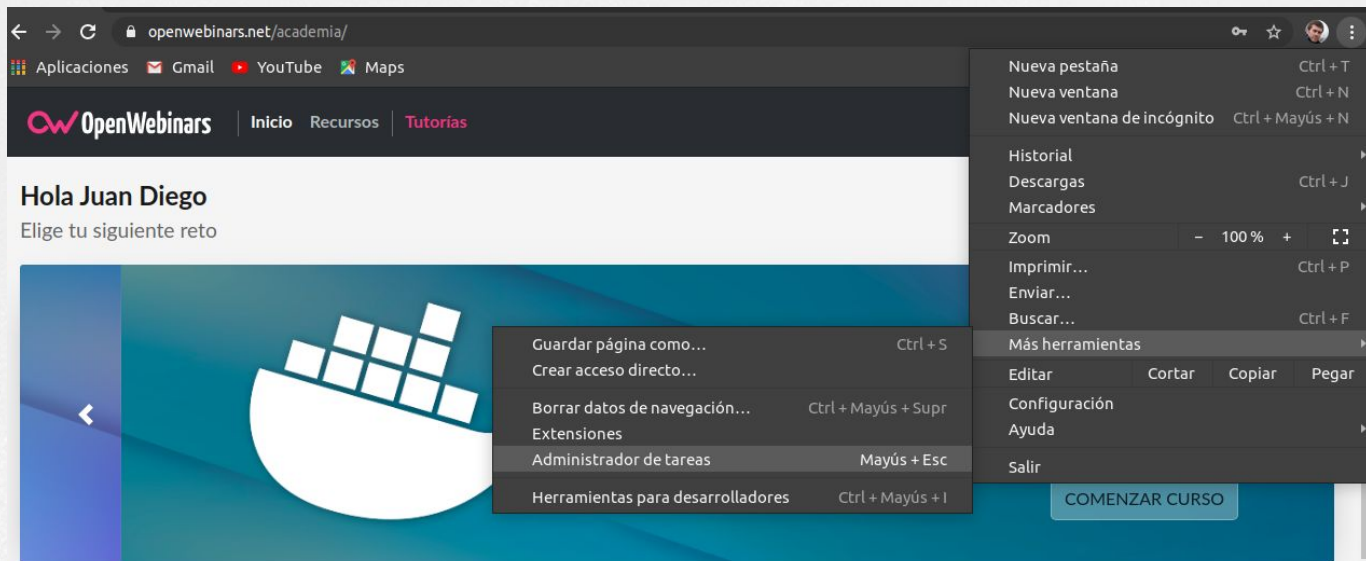
Problemas a detectar

- Páginas que consumen “MUCHA” memoria (Administrador de Tareas)
- Fugas de memoria con el (DOM,JS Head).
- Pausas “injustificadas” de mi página. El GC actúa frecuentemente.
-

Herramientas para el análisis

- **Administrador de Tareas** del navegador.
- El panel **Performance/Memory**.
- El panel **Memory** de las Chrome Dev Tools.

Administrador de Tareas



Punto de partida para detectar problemas de memoria

Administrador de Tareas

Administrador de tareas de Google Chrome

Tarea	Uso de memoria ▼	CPU	Red	ID de proceso	Caché de imagen
Aplicación: Recibidos (2) - pekechis...	418.636 K	1.0	0	146970	1.401 K
Service Worker: https://mail.google...					
Pestaña: DevTools - twitter.com/hc...	361.240 K	0.0	0	151991	186 K
Worker dedicado:					
Worker dedicado:					
• Navegador	252.672 K	18.0	0	129878	—
• Proceso de GPU	124.792 K	13.0	0	129915	—
• Pestaña: DevTools - mail.google.co...	117.400 K	1.0	0	152159	72 K
• Pestaña: Cursos online de Program...	115.956 K	0.0	0	130898	7.320 K
• Pestaña: AS.com - Diario online dep...	111.732 K	1.0	0	166424	1.639 K
• Service Worker: https://twitter.com...	111.728 K	0.0	0	151304	—
• Pestaña: Acelerar la ejecución de J...	101.604 K	0.0	0	130991	5.375 K
• Pestaña: DevTools - www.juntadea...	60.312 K	0.0	0	156304	78 K

Finalizar proceso

- ✓ Tarea
- ✓ Perfil
- ✓ **Uso de memoria**
- ✓ CPU
- ✓ Red
- ✓ ID de proceso
- ✓ Caché de imagen
- Caché de secuencia de comandos
- Caché de CSS
- Memoria de la GPU
- Memoria de SQLite
- Puerto de depuración NaCl
- Memoria de JavaScript**
- Activación inactiva
- Descriptores de archivo
- Prioridad del proceso
- Número de actividades en curso

Administrador de Tareas

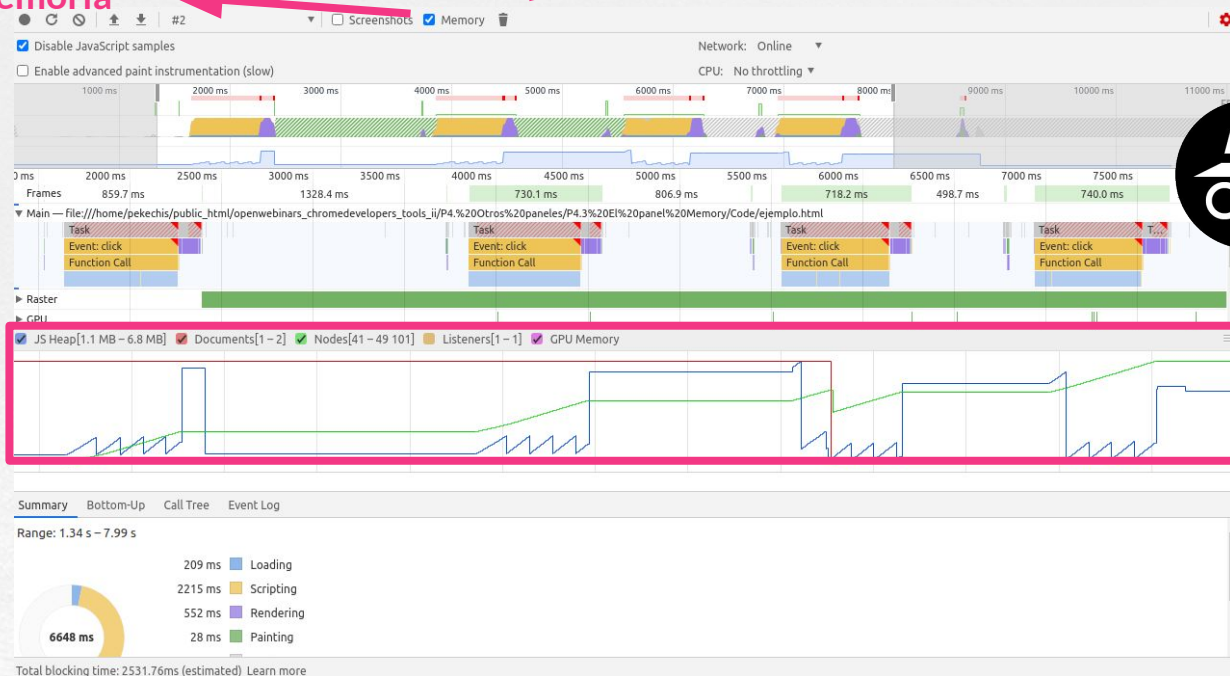
- **Uso de memoria:** Nodos del DOM que están en memoria.
- **Memoria JavaScript:** JS Heap. La memoria que está ocupando las variables/objetos “alcanzables” de mi página.

Memory TimeLine

Vista de memoria

Recolector de basura

Resultados



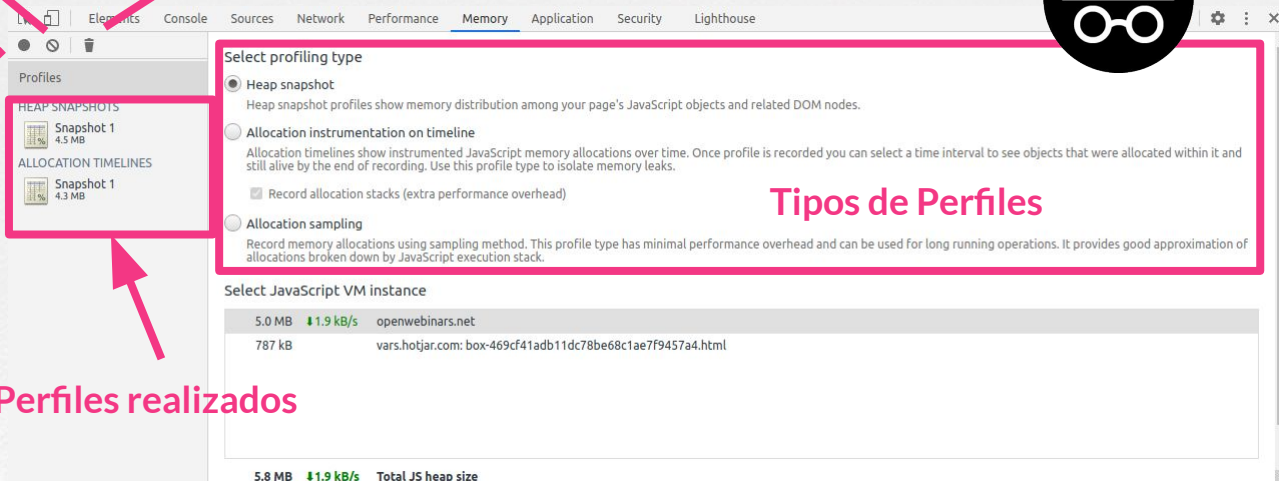
El panel Memory

Borrado

Recolector de basura

Grabación

Perfiles realizados



The screenshot shows the Chrome DevTools Memory panel. A red box highlights the 'Profiles' sidebar on the left, which contains a list of heap snapshots and allocation timelines. A red arrow points from the 'Borrado' (Delete) label to the trash icon in the sidebar. Another red arrow points from the 'Recolector de basura' (Garbage Collector) label to the 'Heap snapshot' option in the 'Select profiling type' section. A third red arrow points from the 'Grabación' (Recording) label to the 'Snapshot 1' entry in the 'HEAP SNAPSHOTS' list. A fourth red arrow points from the 'Perfiles realizados' (Profiles performed) label to the 'Snapshot 1' entry. A red box also highlights the 'Select profiling type' section, which includes options for 'Heap snapshot', 'Allocation instrumentation on timeline', and 'Allocation sampling'. A red arrow points from the 'Tipos de Perfiles' (Profile Types) label to this section. A black circular icon with a white hat and glasses is positioned in the top right corner of the panel.

Profiles

HEAP SNAPSHOTS

- Snapshot 1
4.5 MB

ALLOCATION TIMELINES

- Snapshot 1
4.3 MB

Select profiling type

- ☒ Heap snapshot
Heap snapshot profiles show memory distribution among your page's JavaScript objects and related DOM nodes.
- ☐ Allocation instrumentation on timeline
Allocation timelines show instrumented JavaScript memory allocations over time. Once profile is recorded you can select a time interval to see objects that were allocated within it and still alive by the end of recording. Use this profile type to isolate memory leaks.
 - ☐ Record allocation stacks (extra performance overhead)
- ☐ Allocation sampling
Record memory allocations using sampling method. This profile type has minimal performance overhead and can be used for long running operations. It provides good approximation of allocations broken down by JavaScript execution stack.

Select JavaScript VM instance

5.0 MB	1.9 kB/s	openwebinars.net
787 kB		vars.hotjar.com: box-469cf41adb11dc78be68c1ae7f9457a4.html

5.8 MB 1.9 kB/s Total JS heap size

Tipos de Perfiles

El panel Memory

(profiling type ¿Version?)

- **Heap snapshot:** Distribución en memoria de los objetos Js de tu página y de los nodos DOM relacionados
- **Allocation instrumentation on timeline:** Muestra cómo Js realiza la reserva en memoria de objetos a lo largo del tiempo. Nos permite localizar los memory leaks (fugas de memoria).
- **Allocation sampling:** Igual que lo anterior pero con menos “overhead”. Ideal para análisis de largos periodos de tiempo.

El panel Memory

(Heap Snapshot)

Vista seleccionada

Summary

Class filter

All objects

Distance	Shallow Size	Retained Size
5	52 0 %	372 0 %
5	28 0 %	368 0 %
5	20 0 %	340 0 %
5	28 0 %	336 0 %
5	0 0 %	336 0 %
5	20 0 %	284 0 %
5	48 0 %	264 0 %
5	20 0 %	260 0 %
5	104 0 %	144 0 %
5	0 0 %	116 0 %
5	28 0 %	28 0 %
5	0 0 %	0 0 %
5	0 0 %	0 0 %
5	0 0 %	0 0 %
5	0 0 %	0 0 %
5	0 0 %	0 0 %
5	0 0 %	0 0 %
5	0 0 %	0 0 %
5	0 0 %	0 0 %
4	15 176 0 %	572 080 13 %
4	1 236 0 %	23 516 1 %
4	21 160 0 %	21 348 0 %
4	868 0 %	19 876 0 %
4	2 868 0 %	18 360 0 %

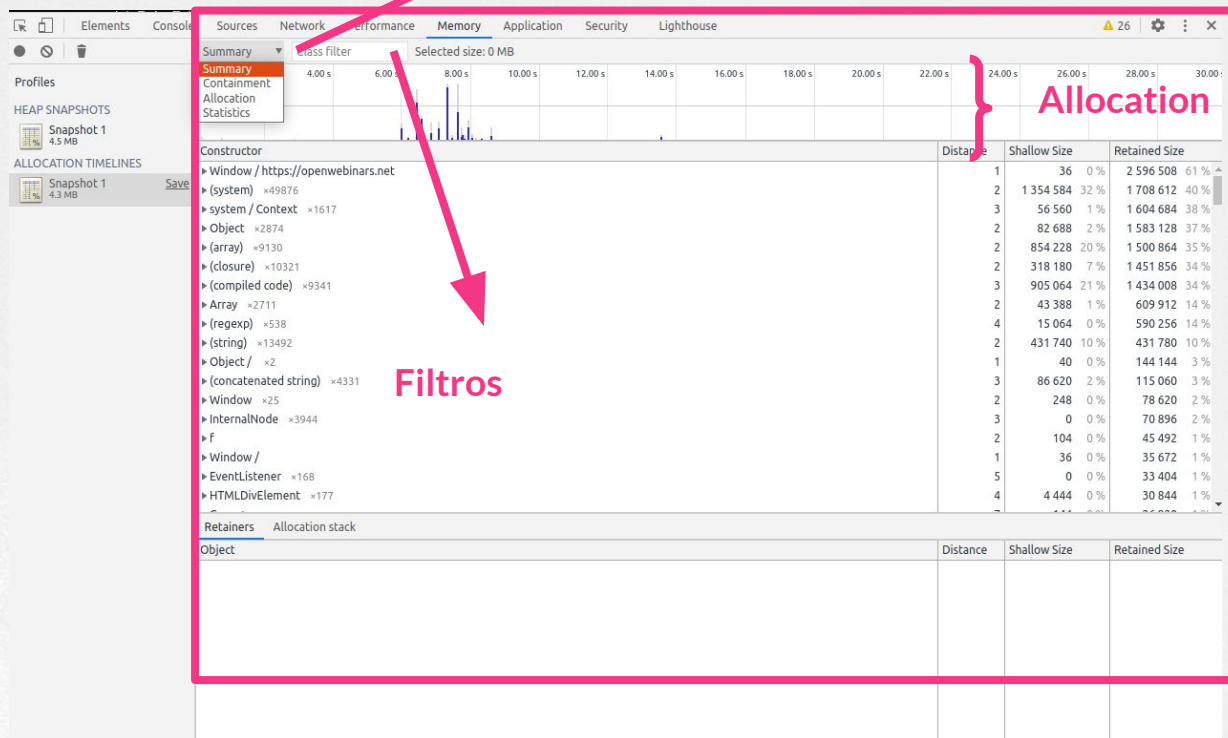
Object

Distance	Shallow Size	Retained Size
----------	--------------	---------------

Filtros y comparaciones

El panel Memory (Allocation)

Vista seleccionada



Conceptos de Memoria



Este es un **panel complejo y aunque a veces los problemas se “vean “ para un mejor uso debemos tener claros ciertos **conceptos** relacionados con la **gestión de memoria.****



Conceptos de Memoria

<https://developers.google.com/web/tools/chrome-devtools/memory-problems/memory-101>

Visualización de problemas

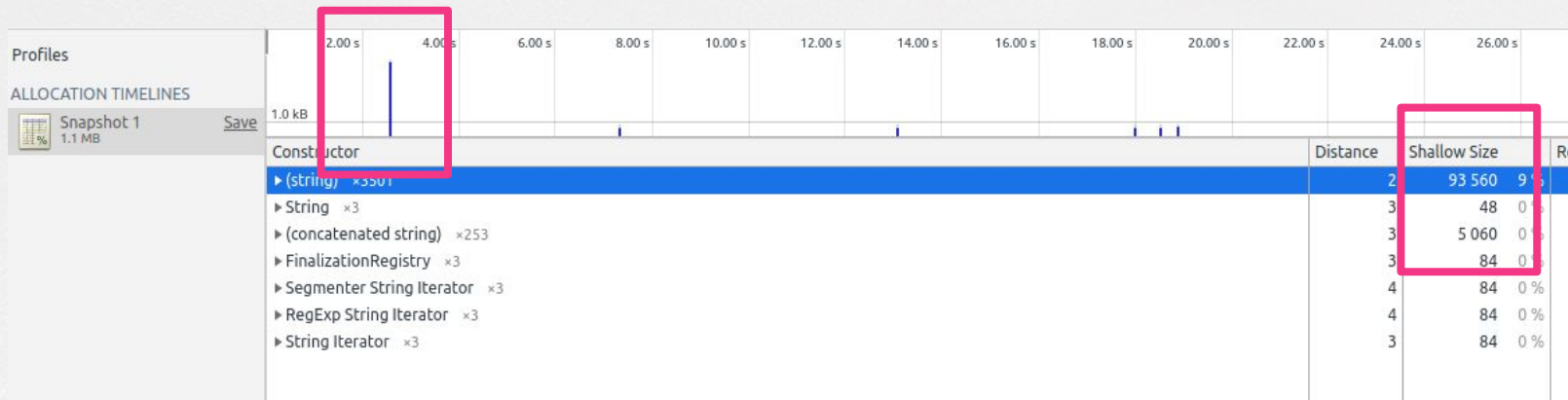
(detached DOM memory leak)

Summary ▾ Detached ✕ Objects allocated before Snapshot 1 ▾

Constructor	Distance ▾	Shallow Size	Retained Size
▶ Detached HTMLCollection ×2	10	56 0 %	
▶ Detached DOMStringMap ×2	8	56 0 %	
▶ Detached CSSStyleDeclaration ×2	8	56 0 %	
▶ Detached Text	6	28 0 %	
▶ Detached HTMLDivElement ×4	4	112 0 %	
▶ Detached HTMLAnchorElement ×2	4	56 0 %	

Visualización de problemas

(JS Heap memory leak)



Visualización de problemas

(Páginas lentas. GBC Actúa frecuentemente)



THANKS!

Any questions?

You can find me at @pekechis &
<https://github.com/pekechis>