

# Velocidad de Carga y JavaScript

## Técnicas + *Resumability*

PediaSure®

ROSA CLARÁ

SUNTORY  
GLOBAL SPIRITS

BELLA AURORA  
pharma

DYC

ABADIA RETUERTA  
CRAFTED SPANISH WINES

## Laura de Vicente

- SEO Estratégico y Técnico
- SEO Manager en Why Ads Media
- Profesora y divulgadora SEO

 Laura de Vicente Bonilla

 @lauradevicente\_

 curiososeo.substack.com

ANDALU-SEO



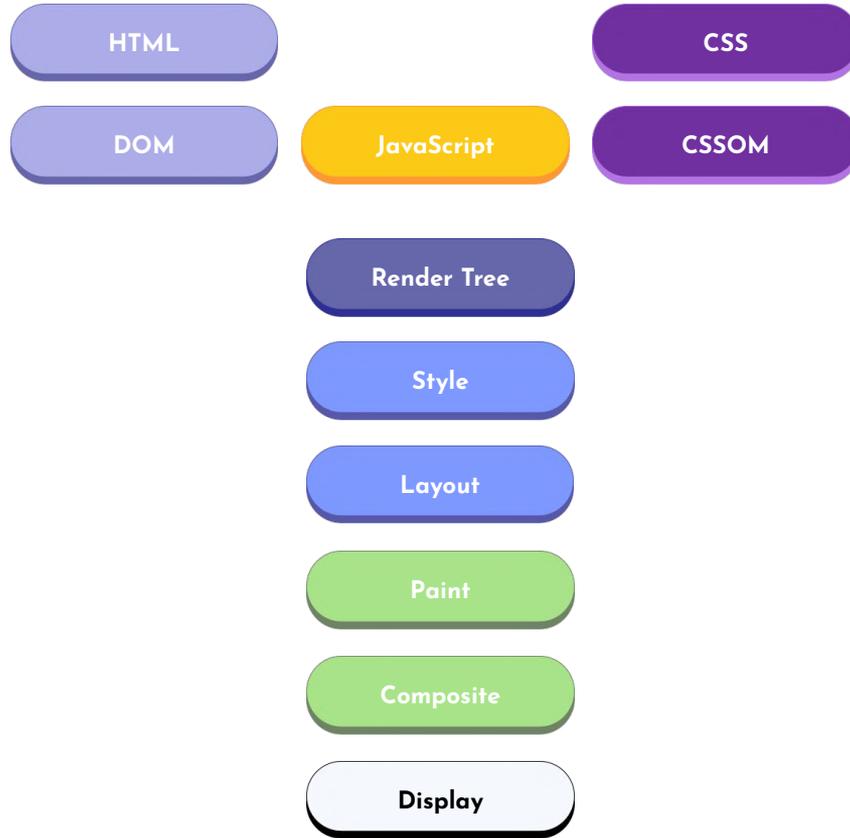
El rendimiento web consiste en garantizar que el contenido se **cargue rápidamente** y pueda responder a la **interacción del usuario**

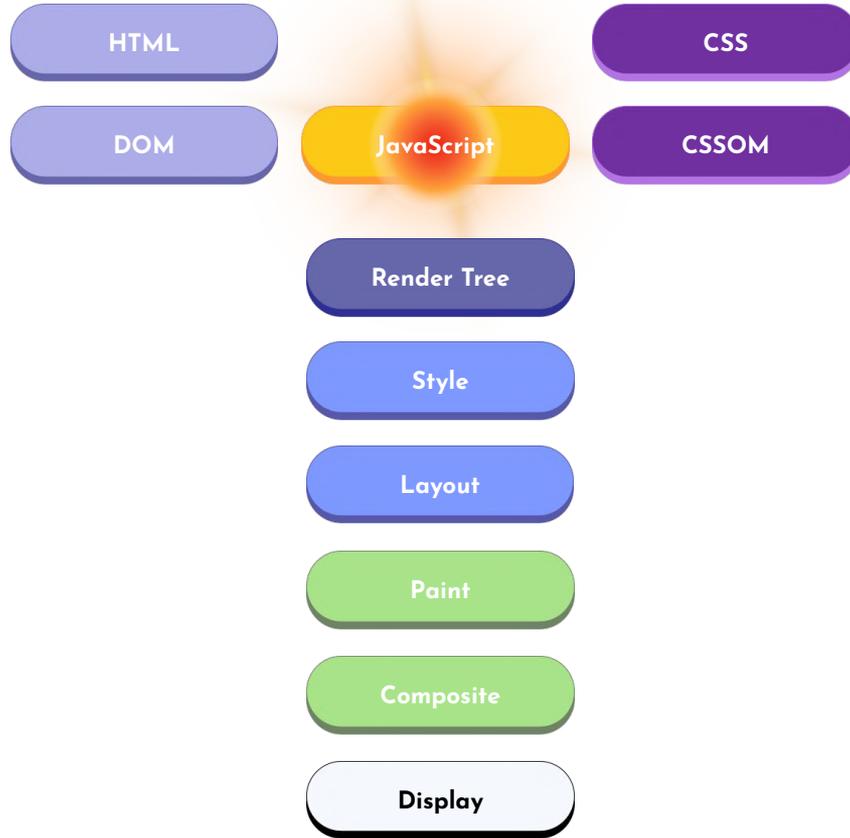
En el mundo web,  
cada **milisegundo** cuenta

En el mundo web,  
cada **milisegundo** cuenta

Google encontró que el **53%** de los sitios web visitados a través de mobile eran abandonados si tardaban **+ 3s en cargar** Fuente: Google Consumer Insights

La velocidad de carga es un aspecto fundamental en una buena **experiencia de usuario**





3 principales problemas que afectan y  
están relacionados con **JavaScript**



**Tareas largas**



**Paquetes de gran tamaño**



**Problemas de Hidratación**

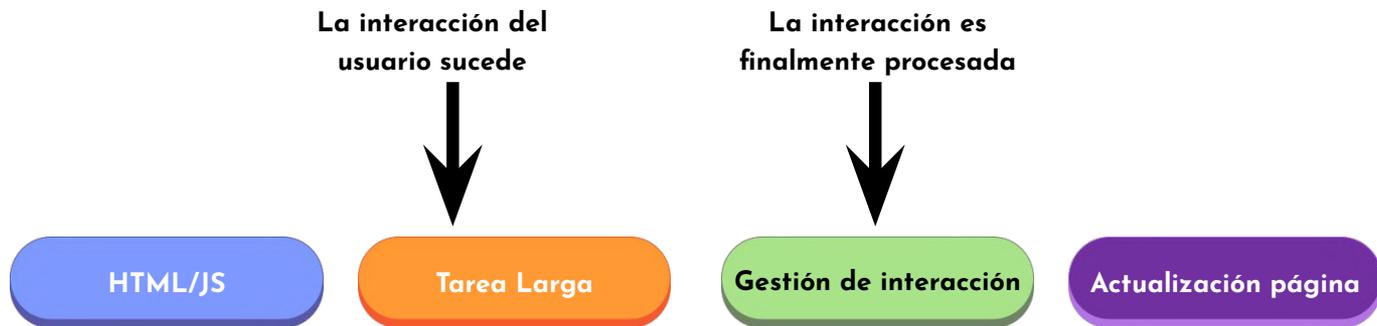


## Tareas largas

- Cuando una actividad constante del hilo principal **bloquea el navegador** durante **50ms** o más.
- Si está ocupado, no puede **gestionar las interacciones** del usuario ni realizar tareas de **renderizado** importantes.

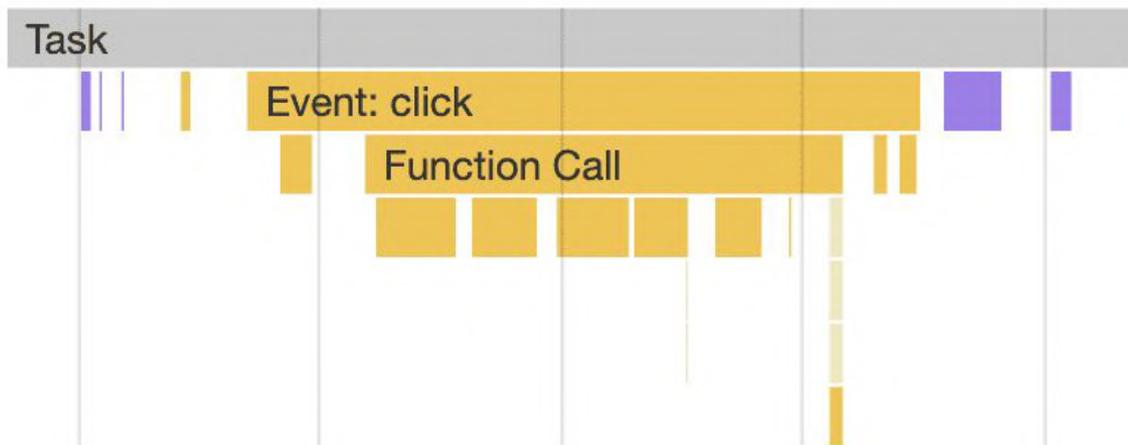


## Tareas largas





## Tareas largas





# Tareas largas

## Web Workers

- Script que se ejecuta en segundo plano, **sin afectar o bloquear** el desempeño de nuestra página web.
- Dos clases de *workers*: dedicados y compartidos.

```
Ejemplo web workers

const items = new Array(100).fill(null);

const workerScript = `
  function waitSync(milliseconds) {
    const start = Date.now();
    while (Date.now() - start < milliseconds) {}
  }

  self.onmessage = function(e) {
    waitSync(50);
    self.postMessage('Process complete!');
  }
`;

const blob = new Blob([workerScript], { type: "text/javascript" });
const worker = new Worker(window.URL.createObjectURL(blob));

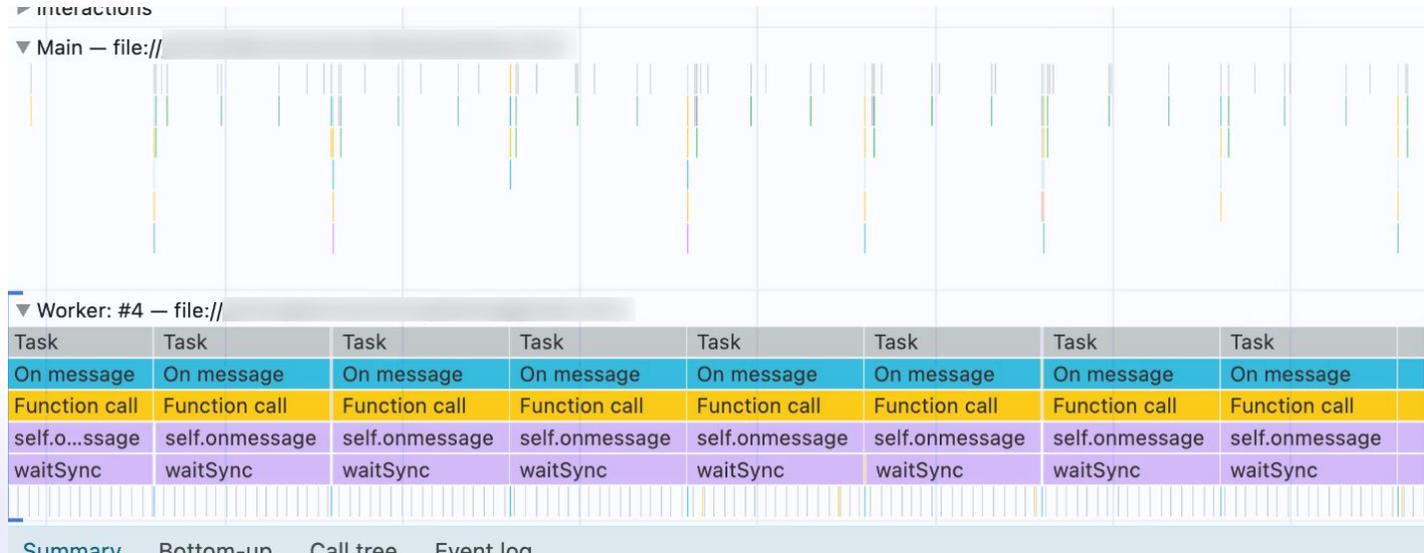
for (const i of items) {
  worker.postMessage(items);

  await new Promise((resolve) => {
    worker.onmessage = function (e) {
      LoopCount.innerText = Number(LoopCount.innerText) + 1;
      resolve();
    };
  });
};
```



# Tareas largas

## Web Workers





## Tareas largas

### `Scheduler.Yield()` al hilo principal

- **Cede al subproceso principal**, lo que permite que el navegador **ejecute** cualquier trabajo pendiente de alta prioridad y, luego, continúe la ejecución donde se detuvo.
- Mantiene una página más **responsive** y, a su vez, ayuda a mejorar el **INP**.
- Las funciones como `setTimeout` o `scheduler.postTask` también se pueden usar para dividir tareas, pero esas continuaciones suelen ejecutarse **después**.



# Tareas largas

## Scheduler.Yield() al hilo principal

```
Ejemplo scheduler.yield

const items = new Array(100).fill(null);

for (const i of items) {
  loopCount.innerText = Number(loopCount.innerText) + 1;

  await scheduler.yield();

  waitSync(50);
}
```



# Tareas largas

## Scheduler.Yield() al hilo principal

Interactions

Main — file://...

| T...  | Task           | Task        |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-------------|
| R...  | Run microtasks | Run ...asks |
| (...) | (anonymous)    | (ano...ous) |
| w...  | waitSync       | waitSync    |



## Tareas largas

### `Scheduler.postTask()`

- Se utiliza para añadir tareas que se **programarán según su prioridad**.
- Permite a los usuarios especificar opcionalmente un **retraso mínimo** antes de que la tarea se ejecute, además de una prioridad para la tarea, y una señal que se puede utilizar para modificar la prioridad de la tarea y/o abortarla.



# Tareas largas

## Scheduler.postTask()

```
Ejemplo scheduler.postTask()

const items = new Array(100).fill(null);

for (const i of items) {
  loopCount.innerText = Number(loopCount.innerText) + 1;

  await new Promise((resolve) => scheduler.postTask(resolve));

  waitSync(50);
}
```



# Tareas largas

## Scheduler.postTask()

► Interactions

▼ Main — file:///l

Task	Task						
Fire postTask	Fire postTask						
Run mi...tasks	Run m...tasks						
(anonymous)	(anonymous)						
waitSync	waitSync	waitSync	waitSync	waitSync	wa...c	w...c	waitSync

# Antes



# Después



**Minimizar** el trabajo y actividad del hilo principal 🕒



# Paquetes de gran tamaño

- Un “bundle” de gran tamaño implica la **cantidad total de código** JavaScript que debe **descargarse**, **analizarse** y **ejecutarse** cuando un usuario visita una web.
- Si son muy grandes provocan
  - **Menor** porcentaje de aciertos en la **caché**
  - Tiempos de descarga **más lentos**
  - **Mayor** tiempo de **análisis y ejecución**





# Paquetes de gran tamaño Eliminar el código no utilizado

The screenshot shows the Chrome DevTools interface. The top part displays the source code of a JavaScript file, with a coverage overlay indicating which parts of the code are executed. The bottom part shows a 'Coverage' report table with the following data:

URL	Type	Total Bytes	Unused Bytes	Usage Visualization
https://www.nike.com/static/uxf/web-shell-client/v1/bundle.umd.js	JS (per func...	32,277	12,439 38.5%	
https://www.nike.com/static/uxf/web-shell-client/2.33.0/549.chunk.cc0d72.js	JS (per func...	223,715	187,244 83.7%	
https://www.nike.com/static/privacy-core/public/privacy-core.js	JS (per func...	57,837	33,220 57.4%	
https://www.nike.com/static/experience/experimentation-eng/optimization-client/bundle.js	JS (per func...	91,135	76,575 84%	
https://www.nike.com/static/experience/experimentation-eng/custom-script/bundle.js	JS (per func...	49,080	33,591 68.4%	
https://www.nike.com/static/experience/experimentation-eng/bellotti/playmaker/bundle.js	JS (per func...	90,867	77,734 85.5%	
https://www.nike.com/static/analytics-client/public/config.chunk.34163a.js	JS (per func...	74,758	74,543 99.7%	
https://www.nike.com/static/analytics-client/public/analytics-client.min.js	JS (per func...	49,785	35,034 70.4%	
https://www.nike.com/static/analytics-client/public/253.chunk.bf8ccd.js	JS (per func...	21,829	18,873 86.5%	
https://www.nike.com/assets/vendor/adobe-alloy/2.21.2/alloy.js	JS (per func...	145,934	95,186 65.2%	
https://www.nike.com/assets/experience/ciclp/landing-pages-next/static/v2/915-edb07c742db/_next/static/image-loader.js	JS (per func...	7,067	6,915 97.8%	



# Paquetes de gran tamaño

## Tree Shaking

- Eliminación de **código muerto**
- Se basa en **import y export** para detectar si los módulos de código se exportan e importan para su uso entre archivos JavaScript.
- Se pueden emplear “module bundlers” como [webpack](#) o [rollup](#)





# Paquetes de gran tamaño

## Tree Shaking

```
math.js ejemplo

export function sum(x, y) {
  return x + y;
}

export function multiply(x, y) {
  return x * y;
}
```

```
index.js ejemplo

import { multiply } from "./math.js"

multiply(5,6)
```

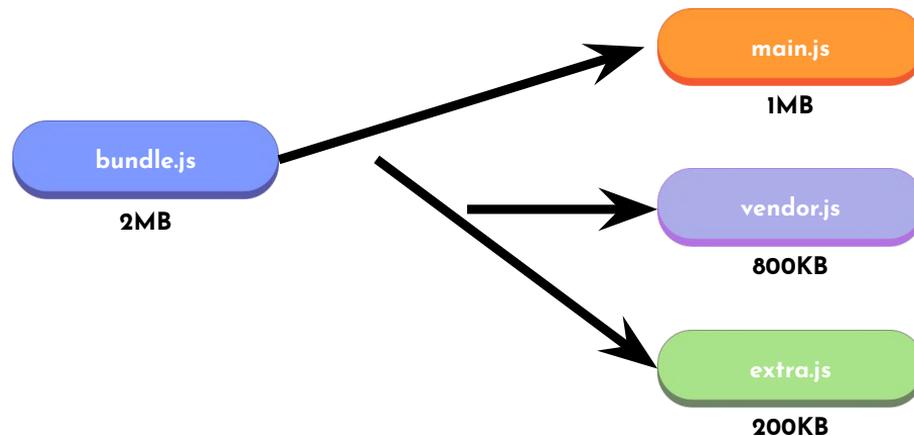
bundle.js



# Paquetes de gran tamaño

## Code Splitting

- Por página
- Below the fold
- Contenido condicional
- Caché del navegador



Hay vida más allá de `<async>` y `<defer>`

# Hidratación

Hidratación  Listo

Obtener HTML

Descargar todo JS de la  
página

Parse y ejecución JS

Enlazar Listeners

# Hidratación

- **Aumento del tamaño del documento:** algunos frameworks serializan el estado en la fuente HTML como JSON, **hinchando la carga inicial** e incluso duplicando datos.
- **Retrasos en la interactividad:** la página está visible, pero aún **no es interactiva**.  
Cuanto mayor sea la carga útil de hidratación, mayor será este retraso.
- **Reconstrucción inútil:** algunos frameworks que ofrecen SSR hacen que el DOM se construya **dos veces**, una en el servidor y otra en el cliente durante la hidratación.



# Hidratación

## Hidratación Progresiva

- Renderiza **todos los componentes**, hidratando primero los componentes **clave** y después pasando progresivamente a los demás en base a una cierta condición.





# Hidratación

## Islas de Arquitectura

- Se genera creando **islas de interactividad** separadas dentro de una página HTML renderizada **estáticamente por el servidor**, reduciendo así la **cantidad** de JavaScript enviado al cliente.
- Los scripts solo son necesarios para los componentes interactivos.





## Resumability

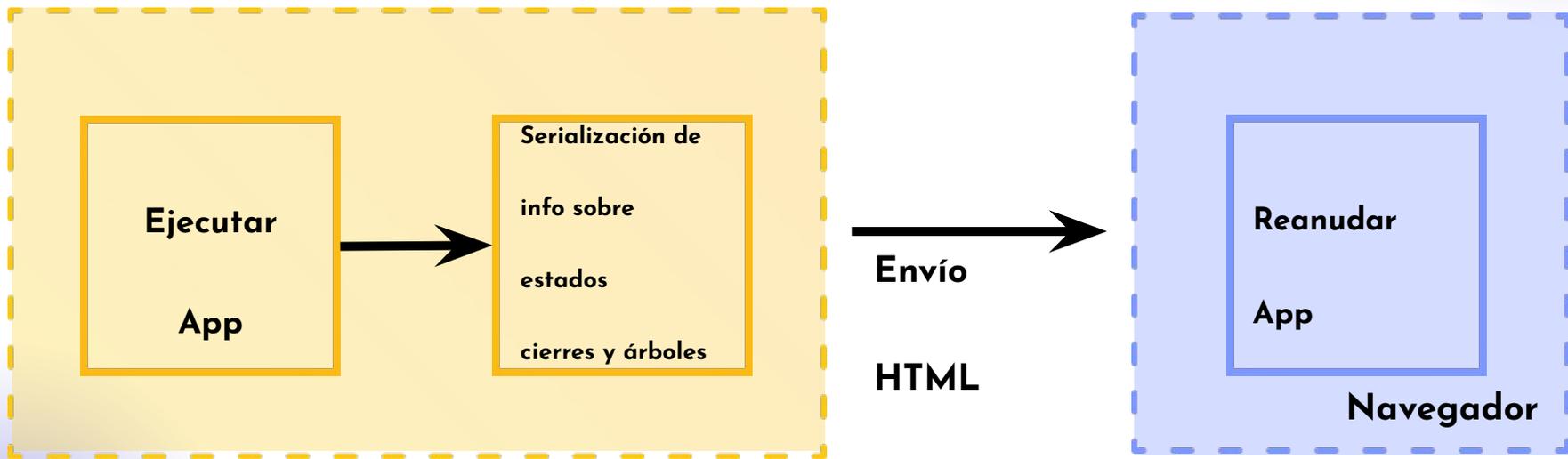
- Puede utilizar la **información** que tenía el servidor cuando **prerenderizó el HTML**, para continuar ejecutándose donde el servidor lo dejó.
- **Serializar** el estado de la página en HTML para que la página sea interactiva hace que **no sea necesario descargar** ni ejecutar ningún código de plantilla.
- Simplemente, se reanuda la página.

Resumable —  Listo

Obtener HTML



# Resumability





# Resumability o Hidratación

## Hidratación

- ✓ Dota de interactividad a la página
- ✓ Promueve el code splitting
- ✓ Reduce el tamaño de los paquetes
- ✓ Permite la carga bajo demanda de partes de la página utilizadas con poca frecuencia
- ✗ Caro
- ✗ Debe ejecutarse antes de que la aplicación sea interactiva
- ✗ Tiene que renderizar el sitio web dos veces

## Resumability

- ✓ Rápido
- ✓ Interactivo antes de cualquier ejecución de código
- ✓ Puede continuar ejecutándose donde lo dejó el servidor
- ✓ Ahorra el coste de volver a ejecutar el árbol de componentes para volverse interactivo.
- ✗ Requiere la descarga previa del código de la plantilla
- ✗ Requiere que el código del "event handler" se descargue por adelantado para poder adjuntarlo

# Performance - Visualizar en la consola

```
<iframe src="https://www.nike.com/static/uxf/guest-session-html/index.html" title="Guest Session" id="guest-session-iframe" style="display:none">
</iframe>
<div id="shared-wrapper">
<div>
<script id="react-script" type="text/javascript" src="https://www.nike.com/assets/vendor/react/17.0.2/react.production.min.js"></script>
<script id="react-dom-script" type="text/javascript" src="https://www.nike.com/assets/vendor/react-dom/17.0.2/react-dom.production.min.js"></script>
<script id="@emotion/core-script" type="text/javascript" src="https://www.nike.com/assets/vendor/@emotion/core/10.0.27/core.umd.min.js"></script>
<script id="@emotion/styled-script" type="text/javascript" src="https://www.nike.com/assets/vendor/@emotion/styled/10.0.27/styled.umd.min.js"></script>
<script id="@emotion/styled-base-script" type="text/javascript" src="https://www.nike.com/assets/vendor/@emotion/styled-base/10.0.27/styled-base.umd.min.js"></script>
</div>
</div>
</div>
```

html body

Console What's new AI assistance

received the embeddable debugging events

- VIDEJS: WARN: Using the tech directly can be dangerous. I hope you know what you're doing. See <https://github.com/videojs/video.js/issues/2617> for more info.
- GET <https://api.nike.com/buy/carts/v2/ES/NIKE/NIKECOM> 404 (Not Found)
- VIDEJS: WARN: Using the tech directly can be dangerous. I hope you know what you're doing. See <https://github.com/videojs/video.js/issues/2617> for more info.

performance

- Performance
- PerformanceMark
- PerformanceEntry
- PerformanceTiming
- PerformanceMeasure
- PerformanceObserver
- PerformanceNavigation
- PerformanceEventTiming
- PerformancePaintTiming
- PerformanceScriptTiming
- PerformanceServerTiming
- PerformanceElementTiming
- PerformanceLongTaskTiming
- PerformanceResourceTiming
- PerformanceNavigationTiming

performance

Performance {timeOrigin: 1743206705017.8, onresourcetimingbufferfull: null, timings: PerformanceTiming, navigation: PerformanceNavigation, memory: MemoryInfo, ...}

ANDALU SEO

@lauradevicente\_

# Performance - Visualizar en la consola

```
▼ 1: PerformanceLongAnimationFrameTiming
  blockingDuration: 32.307
  duration: 91.89999999850988
  entryType: "long-animation-frame"
  firstUIEventTimestamp: 0
  name: "long-animation-frame"
  renderStart: 1824.8999999985099
  ▼ scripts: Array(1)
    ▼ 0: PerformanceScriptTiming
      duration: 77
      entryType: "script"
      executionStart: 1846.6999999955297
      forcedStyleAndLayoutDuration: 6
      invoker: "https://www.nike.com/static/uxf/dotcom-nav-v3/packages/header-footer-renderer/public/commons.clientBundle.6012355e69125c1968ab.js"
      invokerType: "classic-script"
      name: "script"
      pauseDuration: 0
      sourceCharPosition: 0
      sourceFunctionName: ""
      sourceURL: "https://www.nike.com/static/uxf/dotcom-nav-v3/packages/header-footer-renderer/public/commons.clientBundle.6012355e69125c1968ab.js"
      startTime: 1846.6999999955297
    ▶ window: Window {0: Window, 1: global, 2: global, 3: global, 4: Window, 5: global, 6: global, 7: Window, 8: global, window: Window, self: Window, document: document, name: '',
      windowAttribution: "self"
  }
```

PediaSure®

ROSA CLARÁ

SUNTORY  
GLOBAL SPIRITS

BELLA AURORA  
pharma

DYC

ABADIA RETUERTA  
CRAFTED SPANISH WINES

¡Gracias!



Laura de Vicente Bonilla



lauradevicente\_



curioso.substack.com

ANDALU-SEO

