

Mejorando la carga del Javascript

Antes de avanzar, si todavía no sabés bien como medir la velocidad de tu sitio, el siguiente artículo puede ayudarte:

Midiendo la velocidad de tu sitio

Si bien mejorando la carga de las imágenes vas a aumentar bastante la velocidad, el Javascript y el CSS son dos cosas que pueden bloquear la carga del contenido que se encuentre debajo del llamado a estos archivos, ya sea HTML o cualquier cosa en el documento. En este artículo les voy a contar sobre algunas cosas que pueden hacer con Javascript.

La forma "clásica" de construir la maqueta siempre fue de la siguiente forma:

Content

Dependiendo de la importancia y requerimientos del JS, se podía cargar antes del contenido o abajo de todo justo antes de cerrar el `body`.

En el ejemplo anterior, el navegador va a comenzar a leer el archivo HTML, va a cargar primero el CSS, una vez que lo cargó busca el JS, lo carga (hasta acá no cargó nada de HTML y el usuario está esperando frente a una pantalla en blanco) y por último carga el resto del documento.

Quizás con el código anterior no se vea tan grave pero pensá qué pasa si tenés mucho CSS (tuyo y de algún framework como Bootstrap), librerías de JS (como JQuery), plugins de JS (un mundo de posibilidades... para hacer más lento el sitio). Acá empez

a a ponerse mas lento el tema, sobretodo considerando que el navegador puede tomar un tiempo en leer el Javascript que ya fue cargado.

Por esto lo primero que tenés que pensar es en **reducir al mínimo el uso de JS**, esto no quiere decir que no puedas usarlo, está buenísimo y de hecho a mi me salva usar JQuery que muy fácil de implementar, pero hay que tener la escoba lista para barrer todo lo que no sea necesario (y poder resolver vos con un poco más de esfuerzo pero gran recompensa).

Por más tentador que sea, **pensá dos veces antes de agregar un nuevo plugin, app externa o un framework**, todos son granitos que pueden atentar contra la carga del sitio. Te recomiendo usar [BundlePhobia](#) para probar el peso y el posible tiempo de carga del plugin que vayas a agregar.

Mi consejo es que si necesitás plugins busques uno que pueda atacar todas nuestras necesidades ahora y a futuro; y si es posible ver que en github tenga algo de vida o mantenimiento, algún commit reciente o soporte.

Pensá si es realmente necesario tener cargado Bootstrap JS cuando quizás lo usas solo para los popups, esto es algo que tranquilamente se puede resolver con JQuery solo.

Si no hay alternativa por lo menos pensá en cargar todo el JS que no sea crítico para los primeros segundos de la página, como asincrónico, es decir que el HTML no se verá bloqueado por la carga de este JS y el navegador lo cargará cuando lo disponga.

Por otro lado también hay que pensar que plugins son críticos y cuales no, pensando en críticos como los que muy probablemente no tengan que depender de JQuery (de esta forma podrán cargar JQuery asincrónico), como por ejemplo sucede con lazy sizes

Jquery asincrónico

Para cargar Jquery de forma asincrónica simplemente agregá el atributo `async="true"` al llamado del tag

Plugins críticos y no críticos

Como dije anteriormente, es importante dividir los plugins críticos de los que no son críticos para cargar los mas importantes primero y porque puede que los no críticos dependan de que primero esté Jquery cargado (por lo que van a demorar más en cargar).

Un ejemplo de un plugin crítico es lazy sizes (lazy load), lo necesitás cuanto antes para ir mostrando las imágenes principales, mientras que un plugin no crítico puede ser el bxSlider o el mismo JS de Bootstrap. El primero porque depende de Jquery y el segundo porque quizás no sea crítico que funcionen los popups en los primeros segundos de carga de la tienda.

En nuestras tiendas cargamos los plugins antes de cerrar el `body` usando [Twig](#) para que se inserten como código inline y evitamos hacerle el pedido de un archivo más al navegador.

Dividimos los plugins en los archivos:

- **external-no-dependencies.js** : Este contiene los plugins críticos, cuanto menos mejor.
- **external.js.tpl** : Este carga el resto de los plugins, englobados por la función

```
LS.ready.then
```

La forma en la que los cargamos es:

```
{{ "js/external-no-dependencies.js" | static_url | script_tag }}
```

¿Usar plugins asincrónicos quiere decir que pueden cargar cualquier cosa con tal de que sea "async"? ¡**Para nada!** Por más que pongas todo como asincrónico esto va a afectar las métricas de velocidad sobre todo el "Tiempo hasta que la página está interactiva", que es el momento en el cual el navegador dice "ok, ya terminé de cargar el documento".

Así que por más que tengas JS que no bloquee la carga, hay que usarlo a conciencia.

¿Qué sigue?

Si ya pudiste optimizar todo lo que tiene que ver con Javascript te recomiendo seguir leyendo los cambios que aplicamos sobre los otros dos frentes para mejorar la velocidad:

[Mejorando la carga de imágenes y SVGs](#)

[Mejorando la carga del CSS](#)