Documentación

Primera tabla

Para la realización de esta tabla no se hizo ningún cambio en la página web, es la misma web que la utilizada en la PEC2. Para tomar las medidas se utilizó “Regular 3G” y todas se tomaron 7 veces, luego se hice la media eliminando el valor más alto y el más bajo.

Index.html

|  |  |
| --- | --- |
| Título | Index |
| URL |  |
| DOMContentLoaded | 275,6 ms |
| Load | 1,73 s |
| Peso total | 350,17 KB |
| Peso transferido | 136,89 KB |
| Recursos | 11 |

Presentacion.html

|  |  |
| --- | --- |
| Título | Presentación |
| URL |  |
| DOMContentLoaded | 1,22 s |
| Load | 18,06 s |
| Peso total | 4,58 MB |
| Peso transferido | 1,48 MB |
| Recursos | 44 |

Categoría.html

|  |  |
| --- | --- |
| Título | Categoria |
| URL |  |
| DOMContentLoaded | 187,6 ms |
| Load | 9,83 s |
| Peso total | 788,2 KB |
| Peso transferido | 777,6 KB |
| Recursos | 35 |

Detalle.html

|  |  |
| --- | --- |
| Título | Detalle |
| URL |  |
| DOMContentLoaded | 183,8 ms |
| Load | 2,73 s |
| Peso total | 273,16 KB |
| Peso transferido | 258,7 KB |
| Recursos | 11 |

Enlaces.html

|  |  |
| --- | --- |
| Título | Enlaces |
| URL |  |
| DOMContentLoaded | 231,2 ms |
| Load | 597 ms |
| Peso total | 26,31 KB |
| Peso transferido | 10,38 KB |
| Recursos | 7 |

Tablas después de las dos iteraciones

|  |  |
| --- | --- |
| Título | Index |
| URL |  |
| DOMContentLoaded | 262 ms |
| Load | 1,69 s |
| Peso total | 349,02 KB |
| Peso transferido | 136,64 KB |
| Recursos | 11 |

|  |  |
| --- | --- |
| Título | Presentación |
| URL |  |
| DOMContentLoaded | 226,6 ms |
| Load | 736,4 ms |
| Peso total | 120,61 KB |
| Peso transferido | 108,75 Kb |
| Recursos | 12 |

|  |  |
| --- | --- |
| Título | Categoría |
| URL |  |
| DOMContentLoaded | 184,8 ms |
| Load | 695,6 ms |
| Peso total | 322,19 KB |
| Peso transferido | 306,54 KB |
| Recursos | 21 |

|  |  |
| --- | --- |
| Título | Detalle |
| URL |  |
| DOMContentLoaded | 181,4 ms |
| Load | 665,6 ms |
| Peso total | 134,94 KB |
| Peso transferido | 121,9 KB |
| Recursos | 11 |

|  |  |
| --- | --- |
| Título | Enlaces |
| URL |  |
| DOMContentLoaded | 228,8 ms |
| Load | 561 ms |
| Peso total | 25,44 KB |
| Peso transferido | 10,37 KB |
| Recursos | 7 |

Preguntas teóricas

**¿Qué cambios detectas en las herramientas para desarrolladores al aplicar lazy loading a las imágenes de tu web?¿Cómo crees que afecta a la performance de tu página?**

Las imágenes no empiezan a descargarse y cargarse hasta que hago scroll y llego hasta ellas. En consecuencia, muchas empiezan a descargarse una vez la página ya se ha cargado. Esto es muy importante, en especial cuando nuestra página va a utilizar muchas imágenes.

Sin lazy loading el navegador las descargaría todas y esto consume mucho, para los usuarios con mala conexión puede bloquear la página, hace que tarde mucho en terminar de cargar, y a lo mejor ese usuario ni siquiera iba a ver esas imágenes. Con lazy loading se consigue que no se descarguen hasta que el usuario interactúa con ellas.

**¿Qué ocurre al aplicar carga asíncrona a los scripts de tu página?¿Qué problemas crees que puede haber si cargas asíncronamente el javascript?**

Al cargarlo asíncronamente el script deja de ser bloqueante. Cuando se llega al script, este se va cargando sin bloquear el resto de cargas, y cuando está listo se ejecuta, se haya terminado de cargar o no la página.

El problema es que no tenemos control de cuando estará listo para ejecutarse, y esto es un grave cuando tenemos más de un script con dependencias entre ellos. Puede que antes esté listo un script que dependen de otro que aun no lo está, y se producirían errores.

Para estos casos puede ser más recomendable usar defer. En estos casos, los scipts sí que terminarán en el mismo orden que están en el HTML, siempre se ejecutan cuando el DOM ya esté listo.

Para el caso de este trabajo, utilizo async simplemente porque la página es sencilla y no tendré problemas de dependencias.

No hemos hecho carga asíncrona de estilos ¿Crees que se podría hacer?¿Qué problemas podríamos tener? Razona la respuesta.

Sí, de hecho, es posible que en ciertos casos sea positivo cargarlo asíncronamente. Por ejemplo cuando se requiere renderizar cosas que requerirás tiempo y pueden bloquear la página. Pero cargar todo el CSS de forma asíncrona puede ser problemático, porque si es muy pesado tardará en cargarse y durante un tiempo el usuario verá la página sin estilos, se vería simplemente como una web que solo está hecha en HTML, sin CSS.

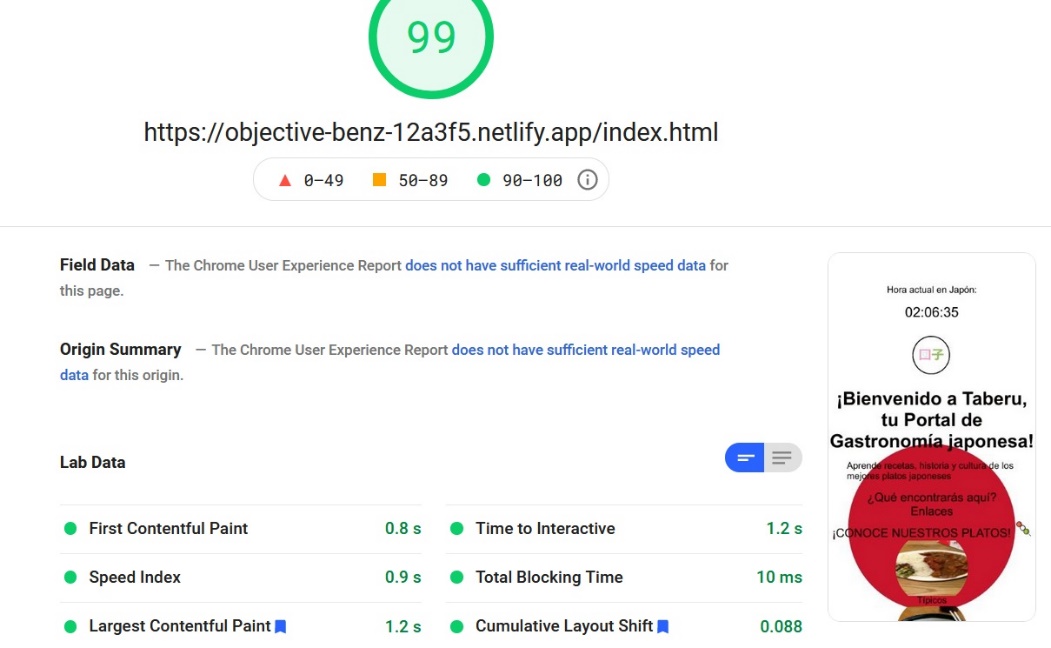
Entonces se puede cargar CSS de forma síncrona para darle estilo a la web, y luego las tareas bloqueantes se podrían cargar de forma asíncrona si no se requiere que el usuario interaccione con ellas inmediatamente.

Informe primera iteración

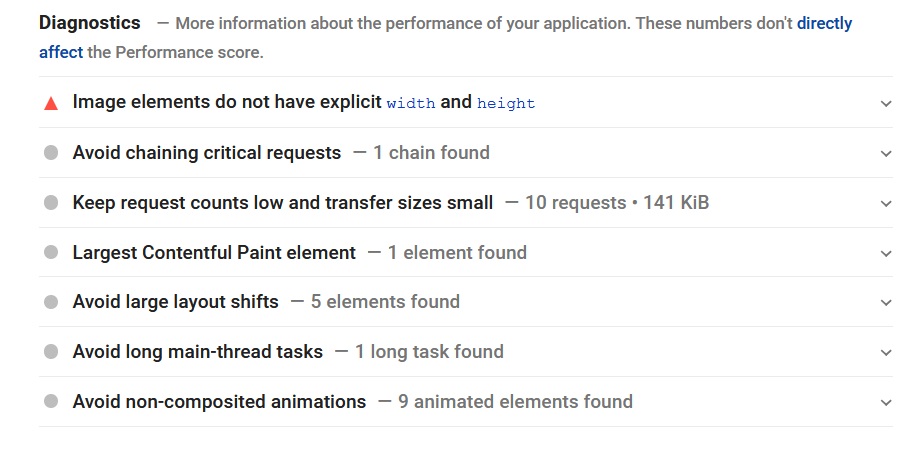
Para la primera iteración la página web es la misma que se utiliza para crear las primeras tablas, pero añadiendo lazy loading en las imágenes y en la carga asíncrona de los scripts.

Index.html

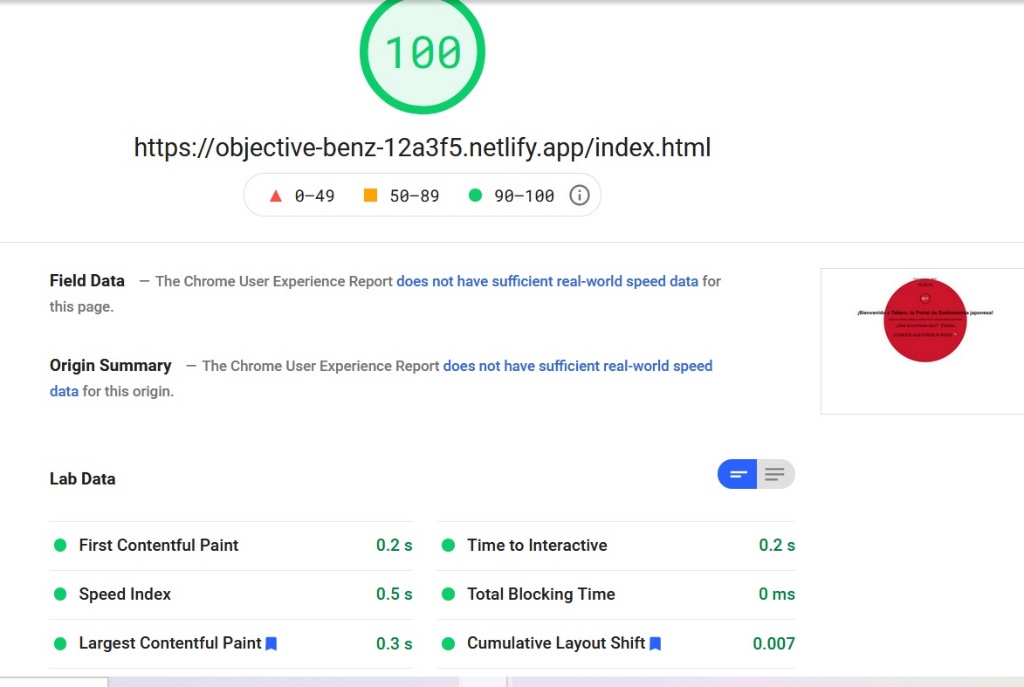
Versión móvil:



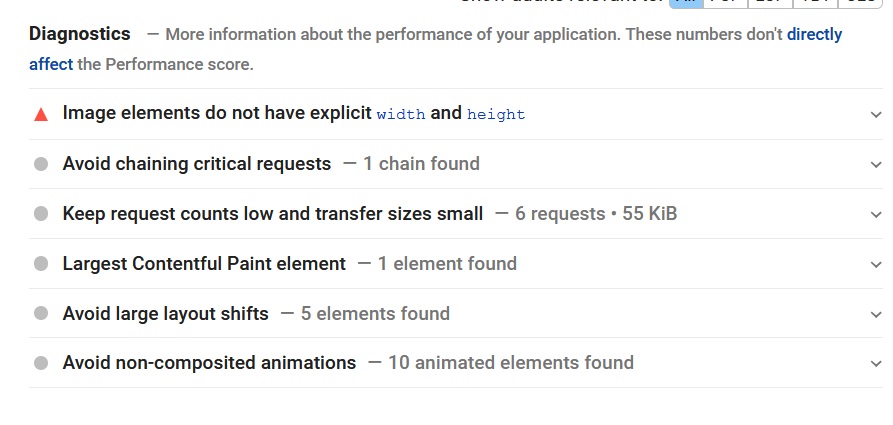
Posibles mejoras:



Versión de escritorio



Posibles mejoras



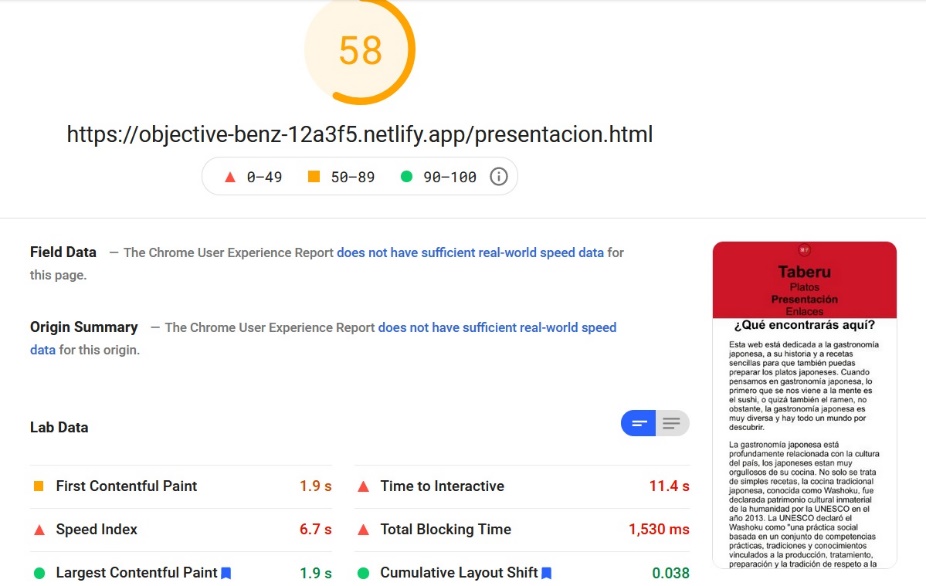
Hay dos mejoras a implementar claras. Primero es reducir el CLS, encuentra que hay 5 desplazamientos grandes. Esto en parte lo causa el script que añade la hora en Japón. Cuando la hora se añade causa que todo se desplace.

Para intentar mejorar el CLS, le doy al div que contendrá la hora japonesa una altura. El navegador descargara y cargara el CSS con la altura del div y luego cuando se añada la hora japonesa no causa un desplazamiento.

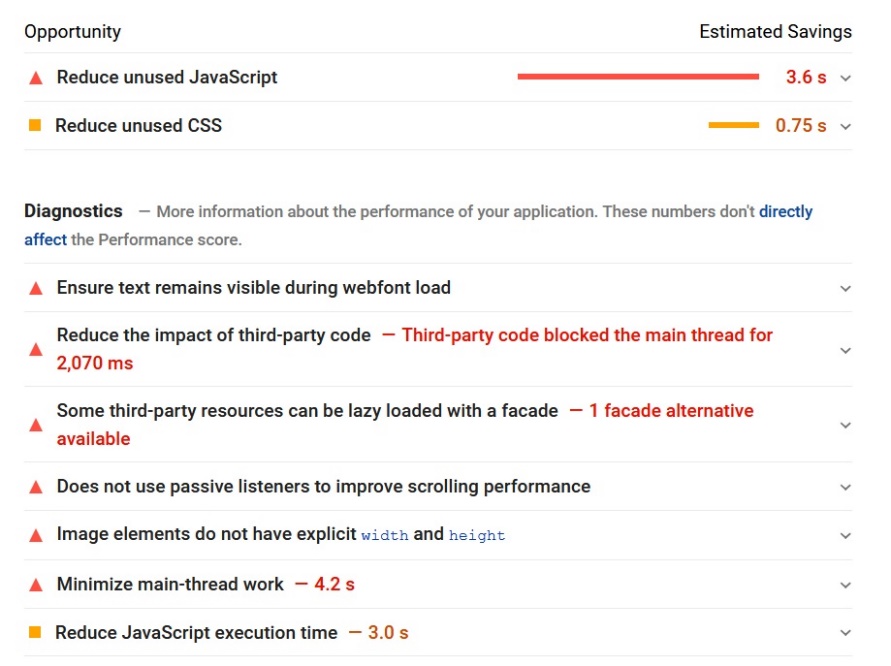
Luego también habla de que los elementos no tienen un width o height explicito. Al leer la documentación, se habla de decirle al navegador que espacio ocuparan los elementos para mejorar el rendimiento y se introduce el aspecto-ratio. Esta propiedad me parece muy útil, no solo para reducir el CLS, sino que además permite que las imágenes mantengan de forma sencilla su aspecto-ratio sin importar el tamaño de la pantalla del usuario, esto es algo muy útil porque muchas veces el “width” de una imagen va cambiando junto al tamaño de la pantalla. Yo también introduzco un aspecto-ratio para mis imágenes, aunque tiene como punto negativo que el soporte a los navegadores antiguos se reduce. Esta medida tendrá un impacto en el resto de páginas.

Presentacion.html

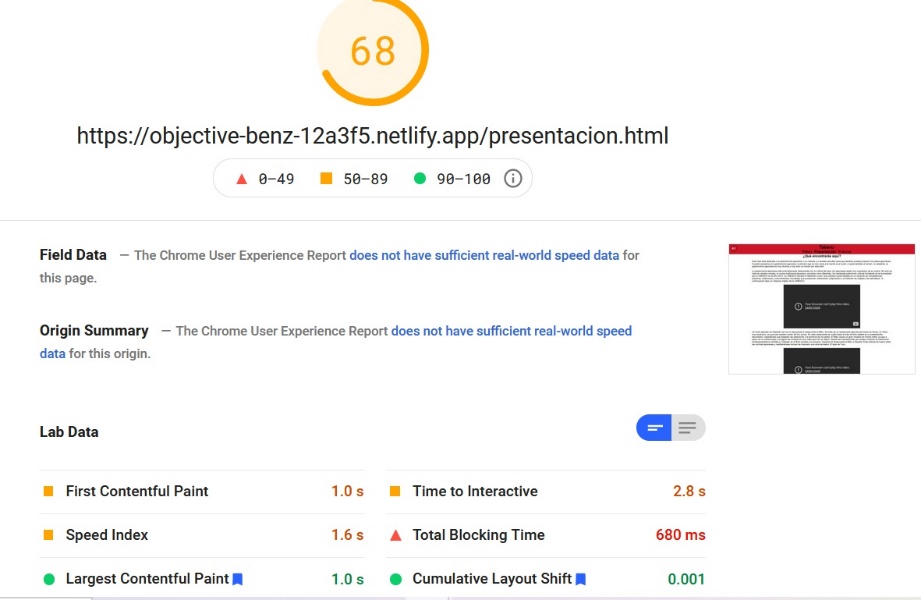
Versión móvil



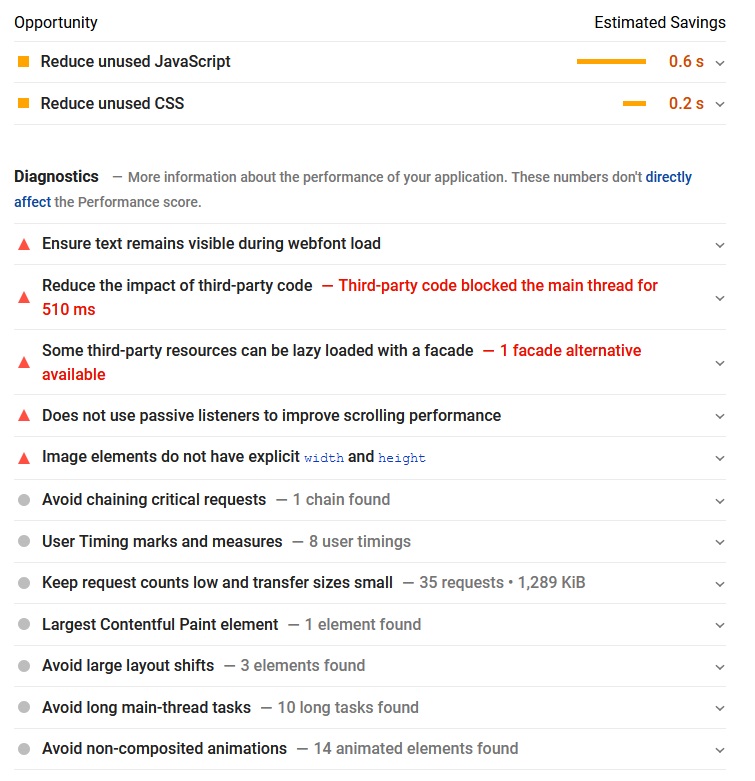
Posibles mejoras:



Versión de escritorio



Posibles mejoras a implementar:

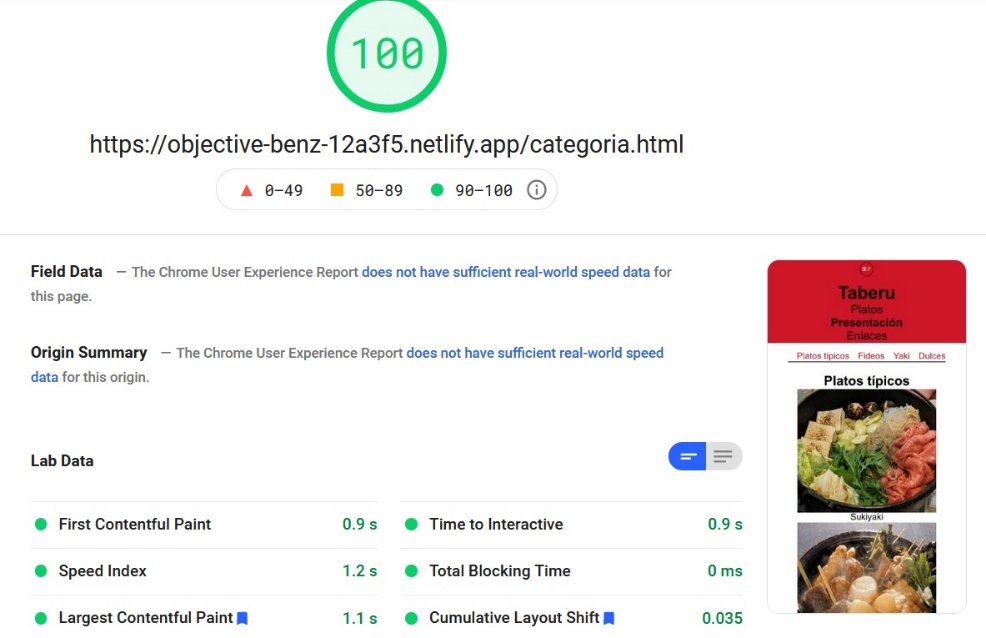


Para esta página el gran culpable de que el rendimiento sea muy bajo es el Javascript y CSS que se descarga por utilizar <iframe> para insertar dos vídeos en Youtube, para la versión móvil podría llegar a reducir hasta 6 segundos.

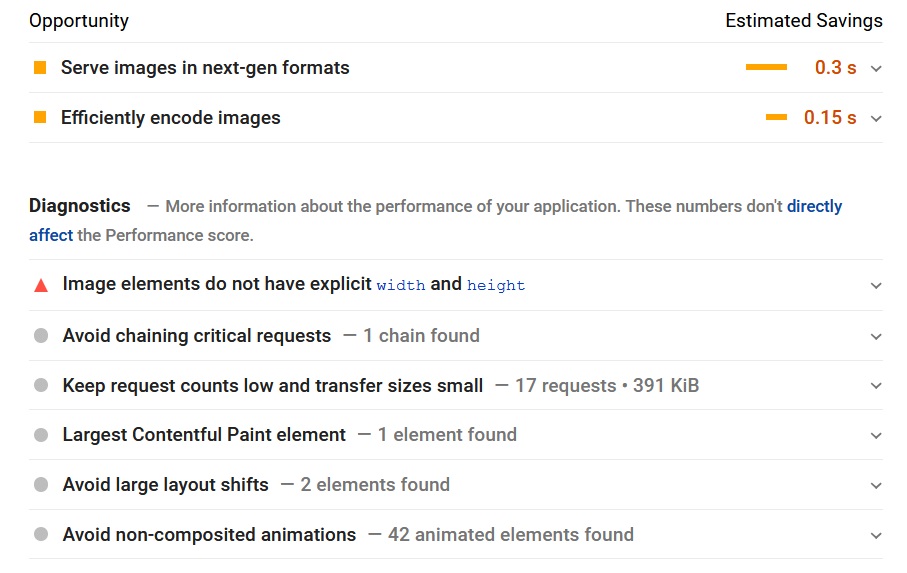
Al ser scripts externos no puedo utilizar simplemente el atributo async, así que para intentar mejorar el rendimiento añado Javascript propio, cargado asíncronamente, para que inserte los vídeos una vez cargada la página. Es decir, dejo un div en el HTML vacío y utilizo el evento onload para que cuando la página ya esté cargada, se añadan los dos vídeos y así evitar que bloqueen la página web. Este código aun puede verse comentado(hago un cambio después de la segunda iteración) en el fichero form.js.

Categoría.html

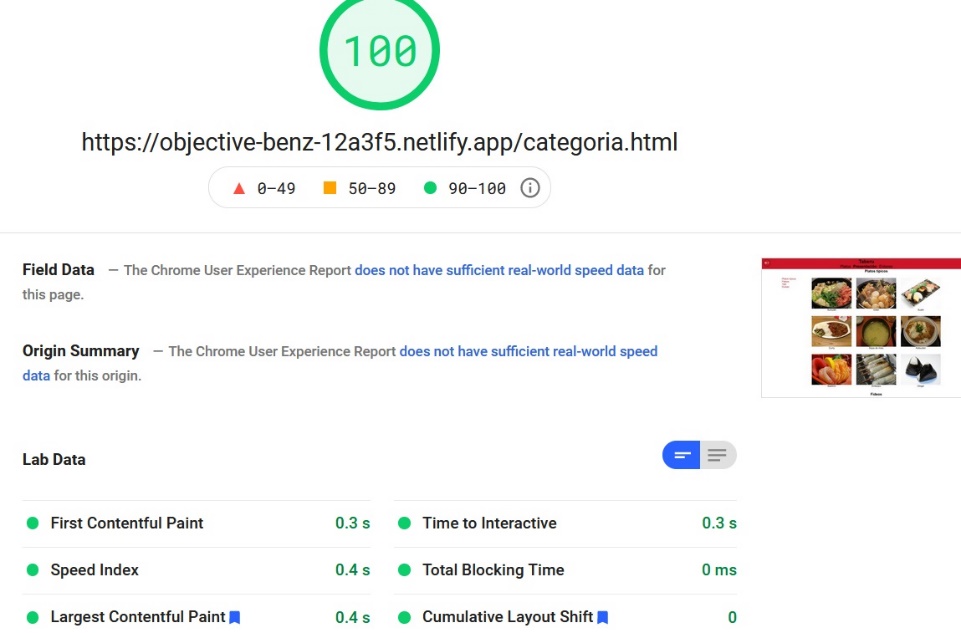
Versión móvil



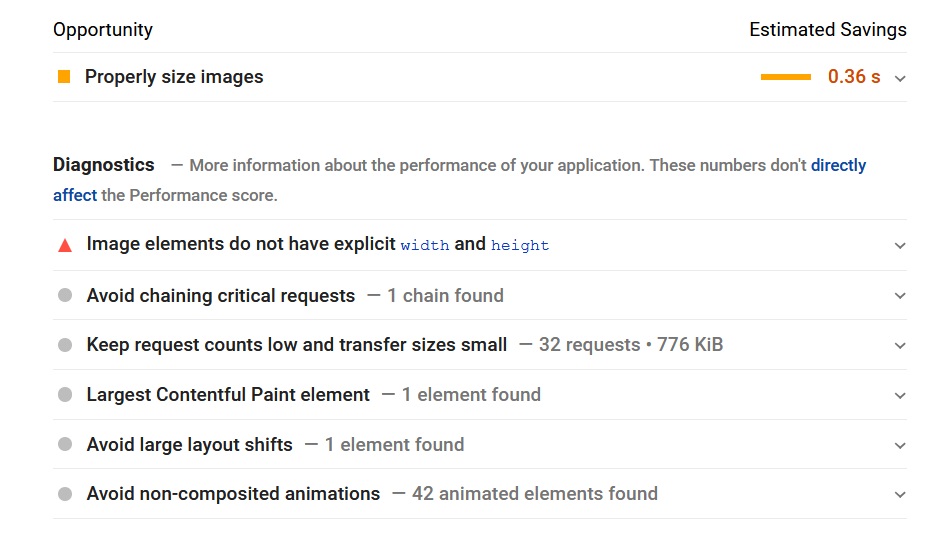
Posibles mejoras:



Versión de escritorio



Posibles mejoras



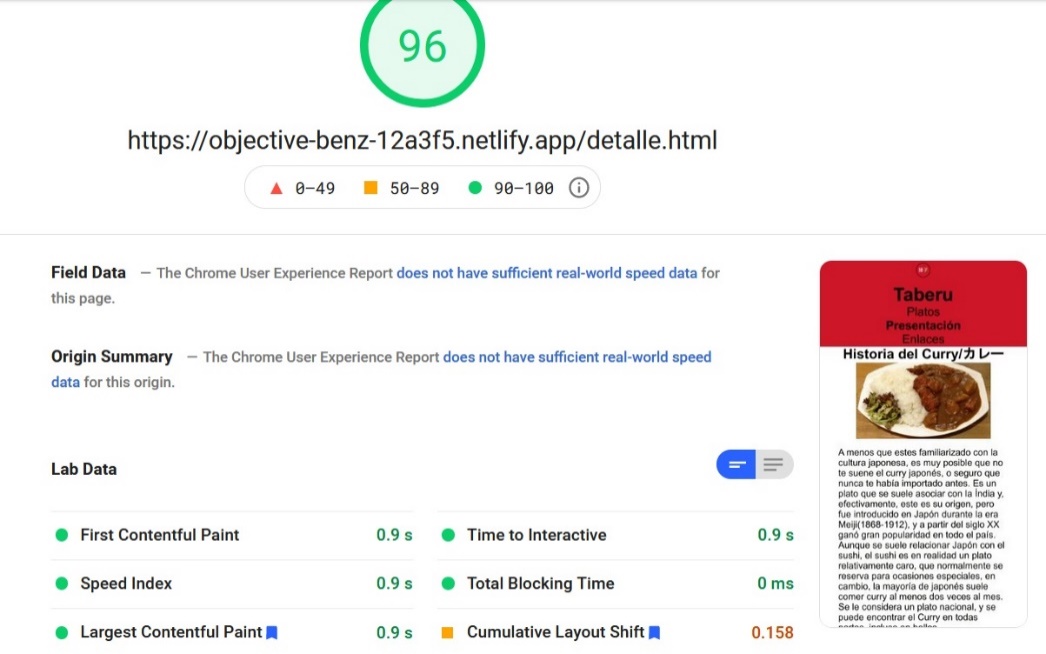
Esta página es la que utiliza más imágenes, para poder optimizarla en las oportunidades se sugiere utilizar formatos más nuevos. El problema de estos formatos es que tienen un soporte limitado, excepto WebP que lo soportan gran cantidad de navegadores. Así que para aprovechar eso convierto todas las imágenes a WebP, aunque renunció a dar soporte a Internet Explorer.

También pongo a las imágenes un width y height explícitos y el aspecto-ratio ya utilizado en index.html

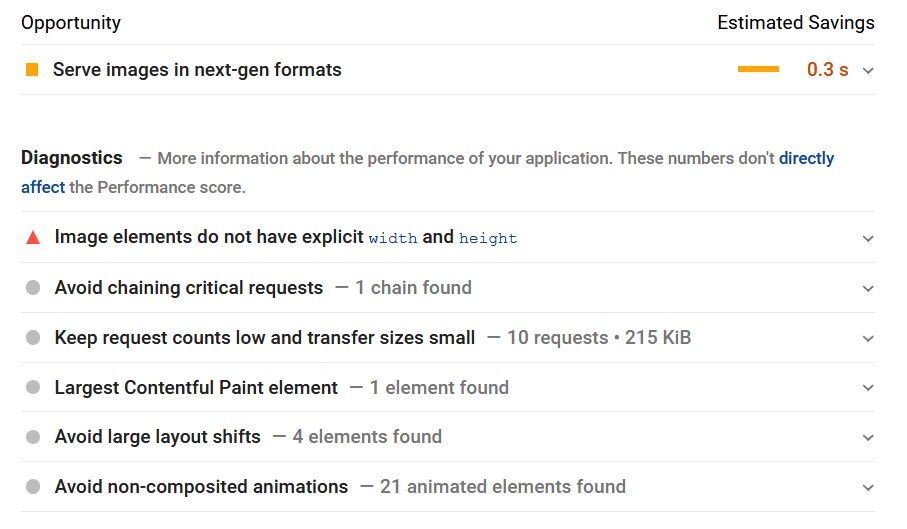
Por último, se sugiere evitar las animaciones no compuestas. Son animaciones que hacen que el navegador trabaje más porque deben volver a pintarse. Para hacer animaciones no compuestas hay que utilizar propiedades como transition y opacity. Esto no siempre es posible, pero en mis animaciones sí. Mis animaciones consisten en modificar el tamaño y la opacidad de las imágenes y enlaces, eso me permite usar transform con scale y opacity. Así que he cambiado el CSS para que las animaciones sean compuestas evitando propiedades prohibidas. Esto también tiene un impacto en el resto de páginas

Detalle.html

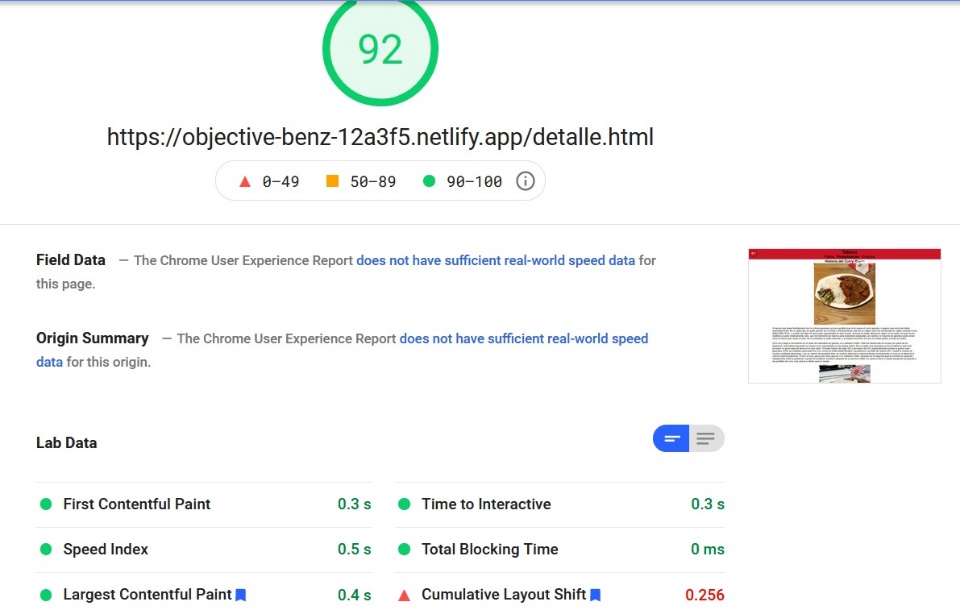
Versión móvil



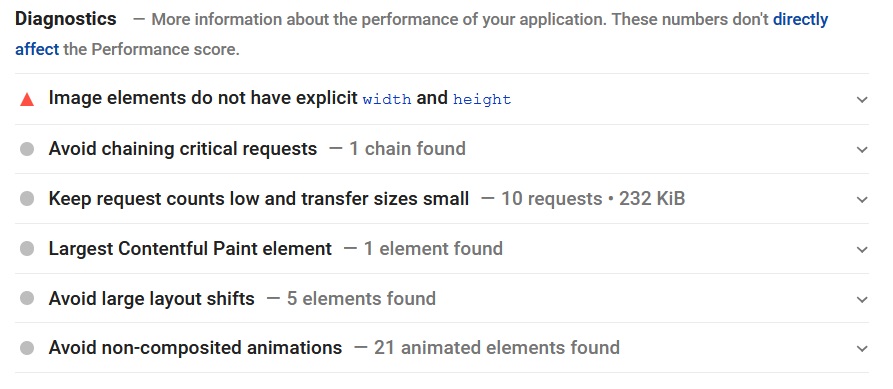
Posibles mejoras:



Versión escritorio



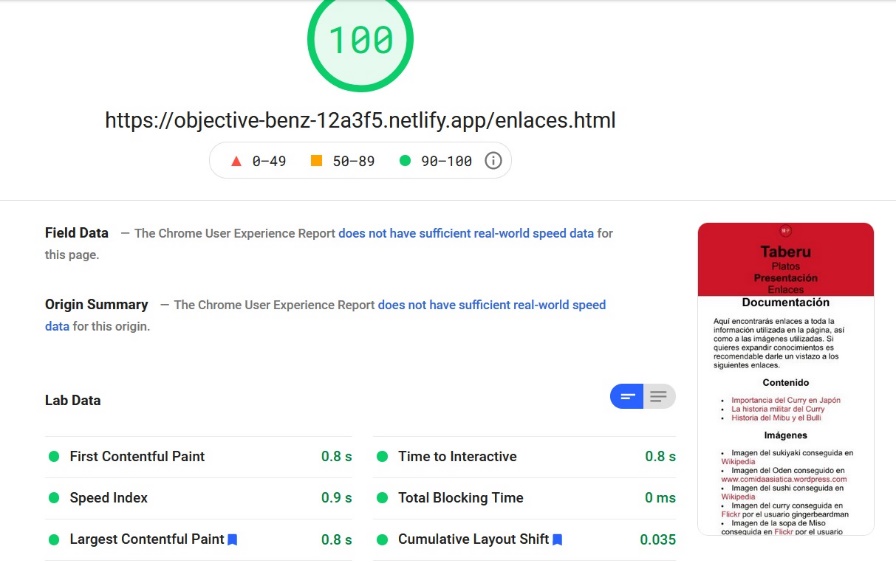
Posibles mejoras



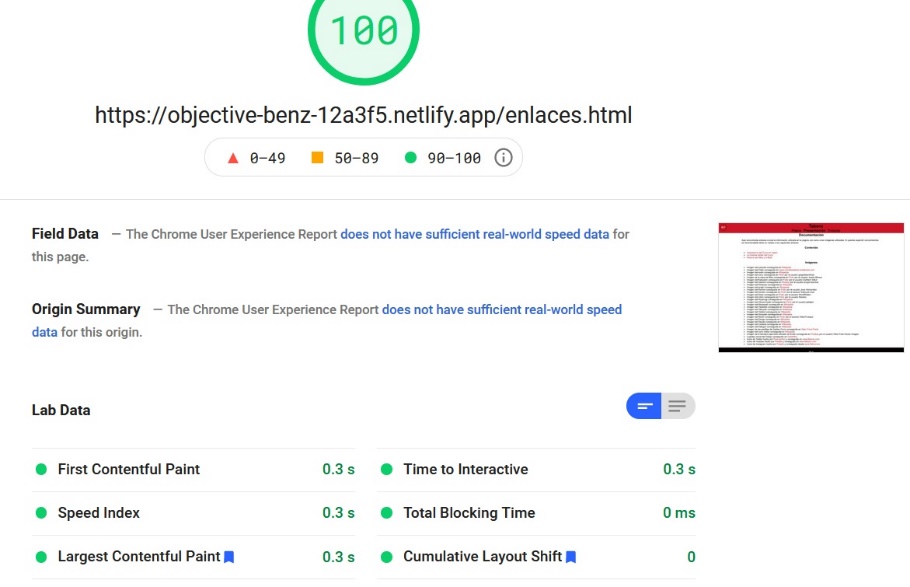
El mayor problema en esta página es el CLS, luego las mejoras que implemento son las mismas que en la página de categoría: Evitar las animaciones no compuestas, usar un aspecto-ratio, e imágenes en WebP.

enlaces.html

Versión móvil



Versión escritorio

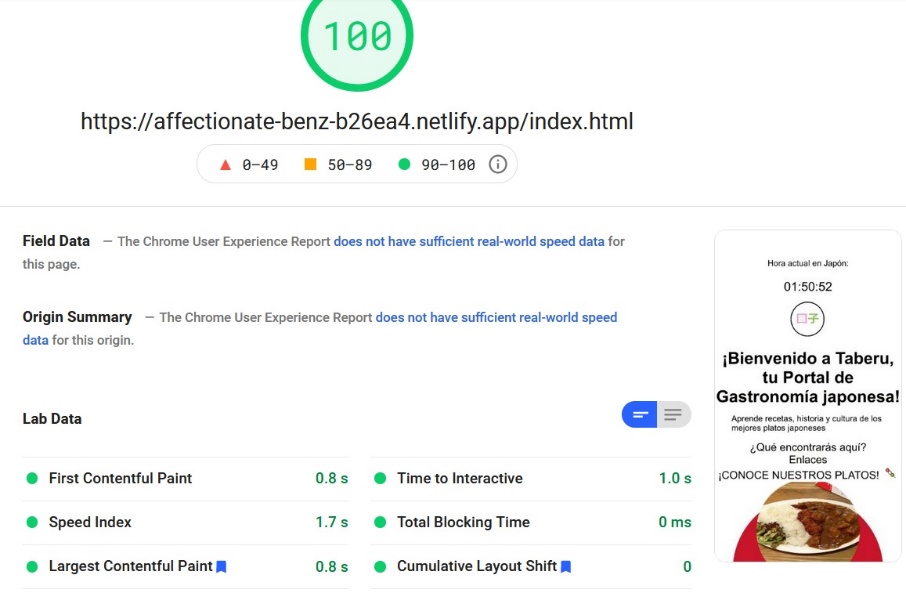


De nuevo el diagnóstico es como la página de detalle o categoría. Deben evitarse las animaciones no compuestas e indicar al navegador el tamaño que ocuparán las imágenes. Las mejoras son las mismas que en los casos anteriores.

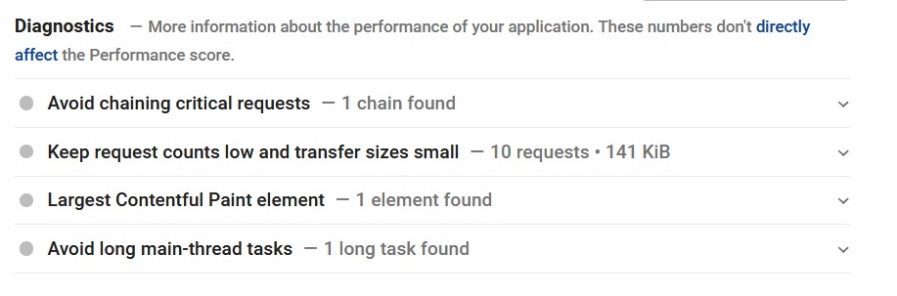
Segunda iteración

Index.html

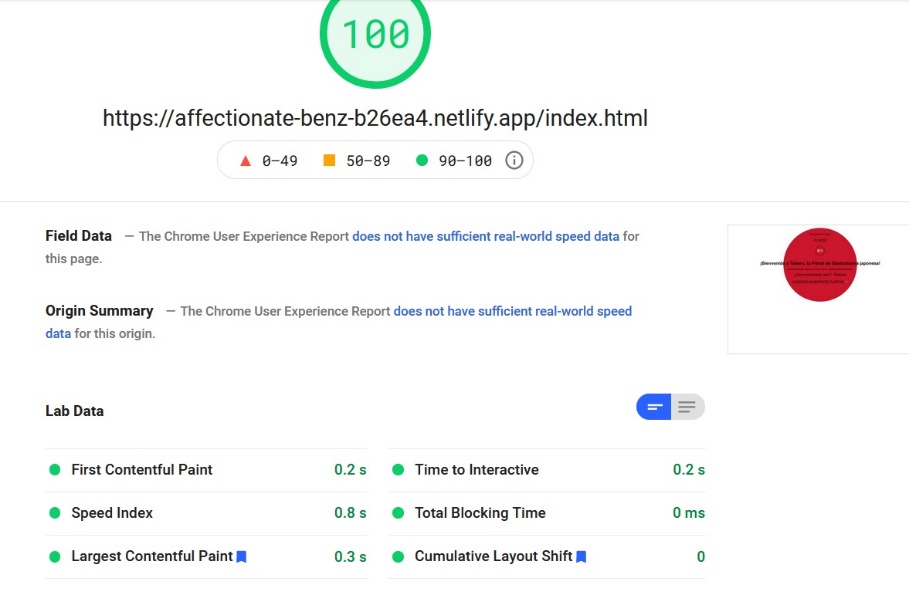
Versión móvil



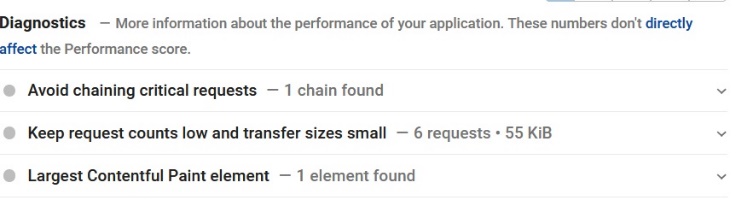
Posibles mejoras



Versión escritorio



Posibles mejoras

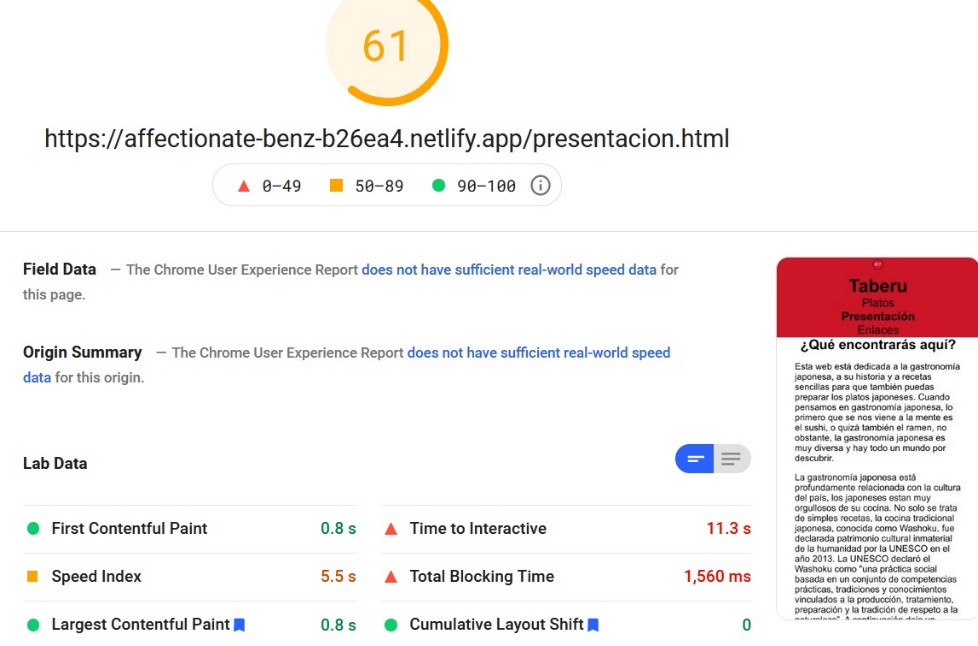


Como se puede observar, ahora sí que se ha conseguido reducir el CLS a 0 y la página obtiene buen resultado. El diagnóstico es casi el mismo en ambas versiones, dando aun ciertos consejos pero ya no considero que sea necesario aplicar más cambios.

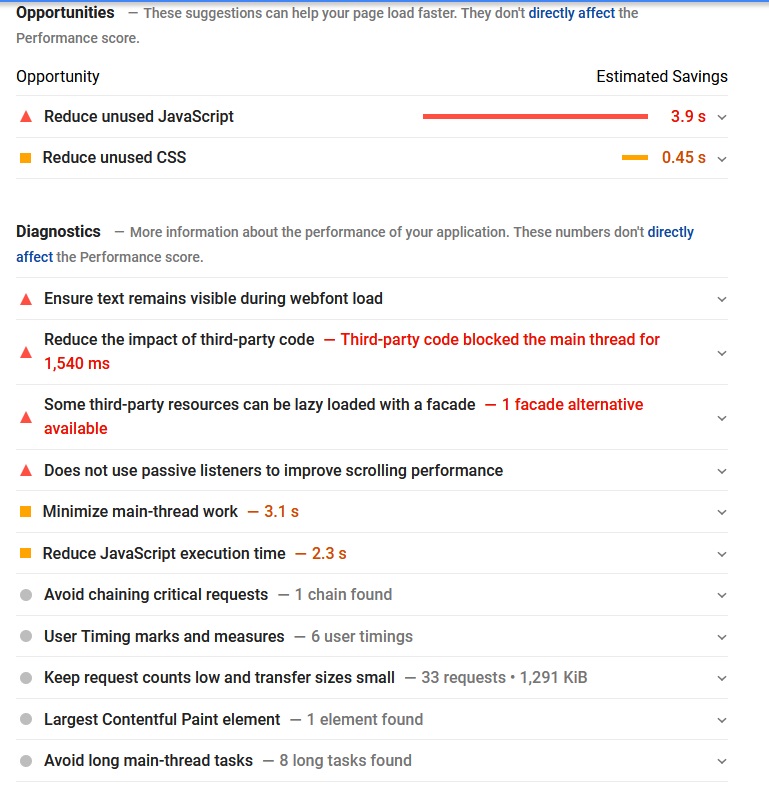
.

Presentación.html

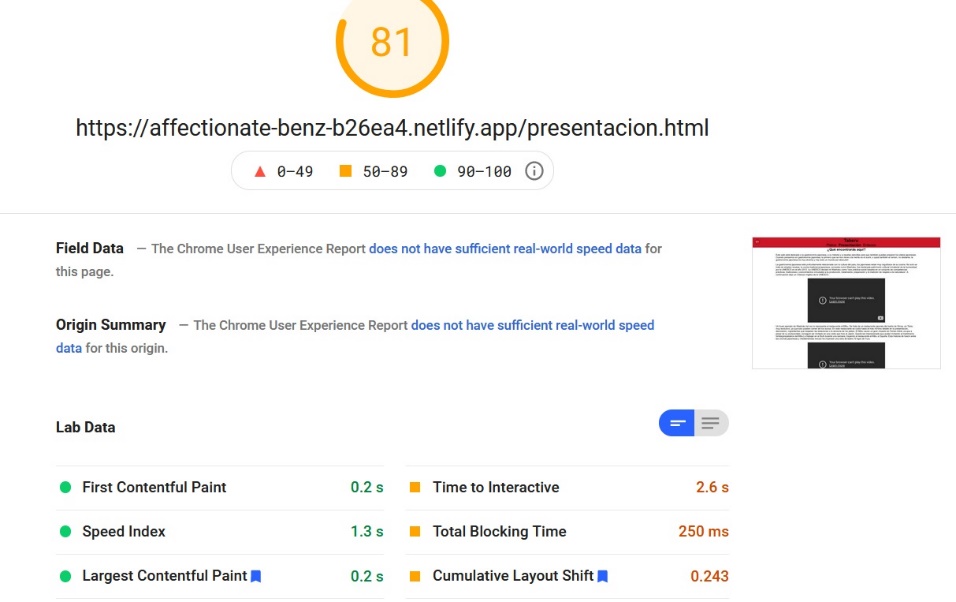
Versión móvil



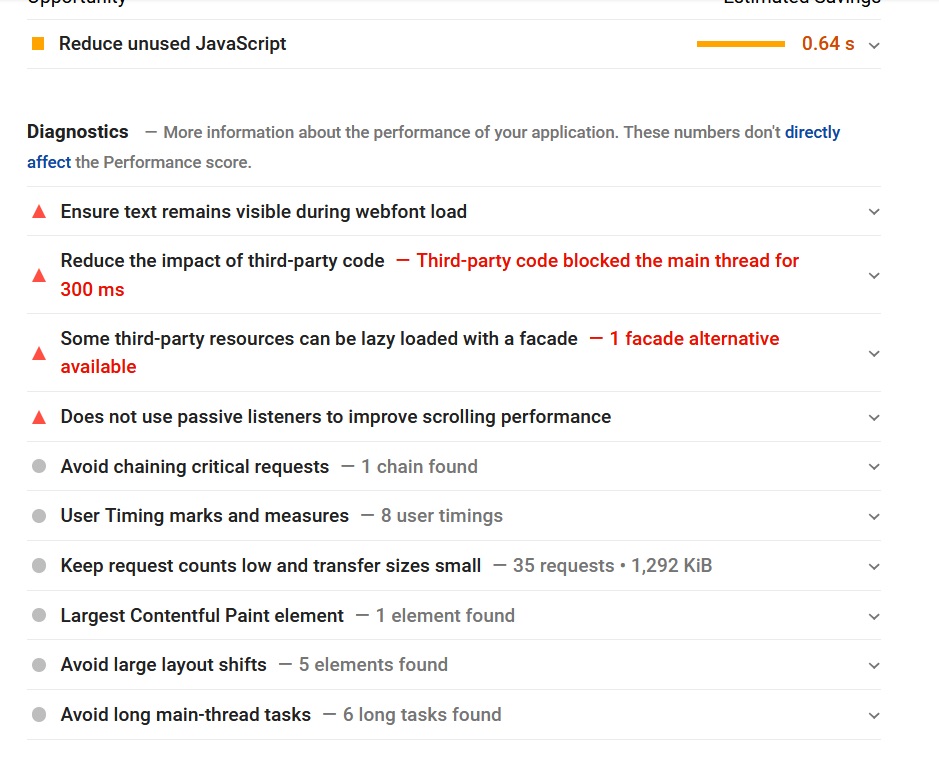
Posibles mejoras



Versión escritorio



Posibles mejoras

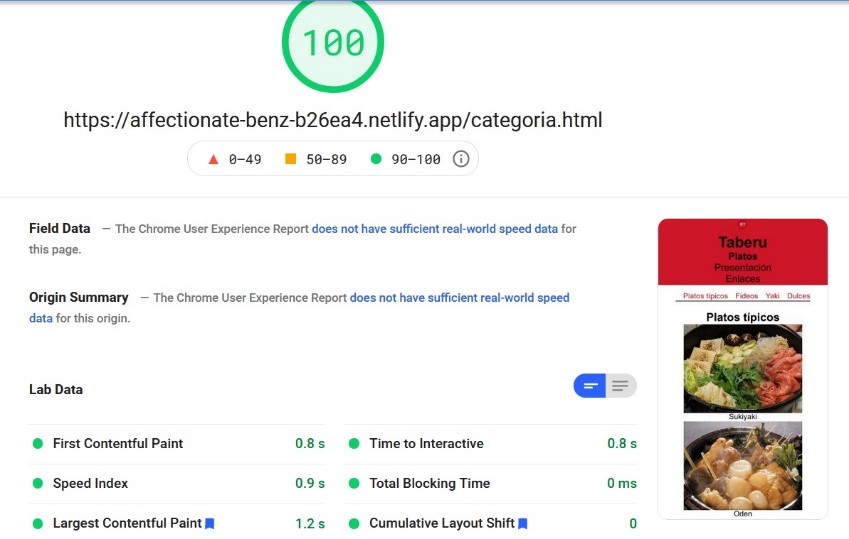


Hay mejoras evidentes en el FCP y LCP. Sobretodo mejora el rendimiento en la versión escritorio donde la puntuación ya llega a 81, pero sigue habiendo demasiado código bloqueante. Las posibles mejores siguen siendo las mismas que en la primera iteración, pero además se añade un nuevo problema, el CLS aumenta ya que ahora es el Javascript quien añade los <iframe>. Estoy dejando un espacio vacío entre los párrafos que luego se llena añadiendo el vídeo.

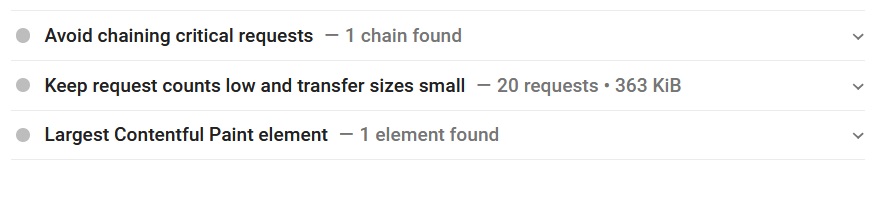
Revisando las posibles soluciones, se sugiere: Some third-party resources can be lazy loaded with a facade.

Cogiendo esa idea, edito el Javascript y cambio el HTML. En vez de dejar un espacio vacío, añado una imagen del video que se va a reproducir, que tendrá el mismo tamaño, pero en realidad es solo una imagen, el navegador no descargará todo el código Javascript asociado al <iframe>. Luego con los eventos mouseover y click(para los usuarios de móviles), se cambia la imagen por el <iframe>. De esta manera el rendimiento sí que mejora mucho, la puntuación pasa a 100 en ambos casos.

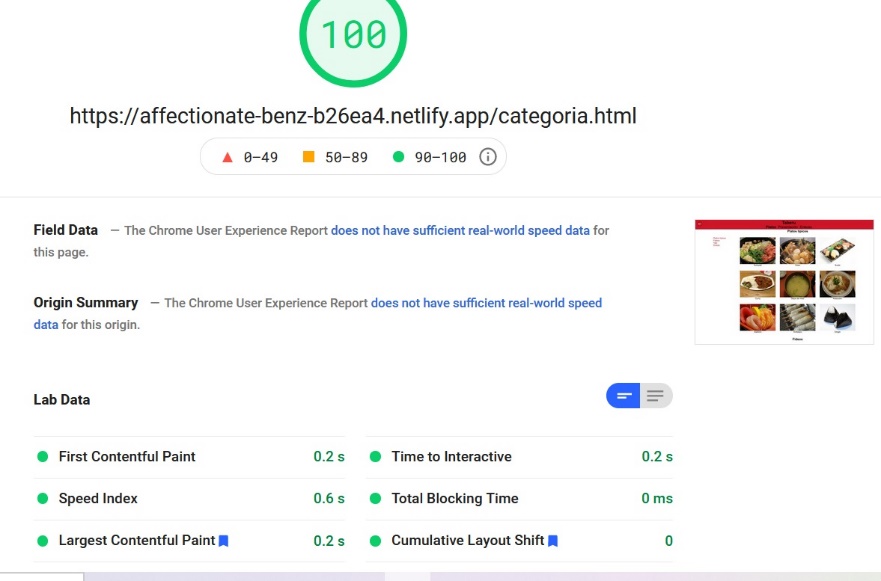
Categoria.html



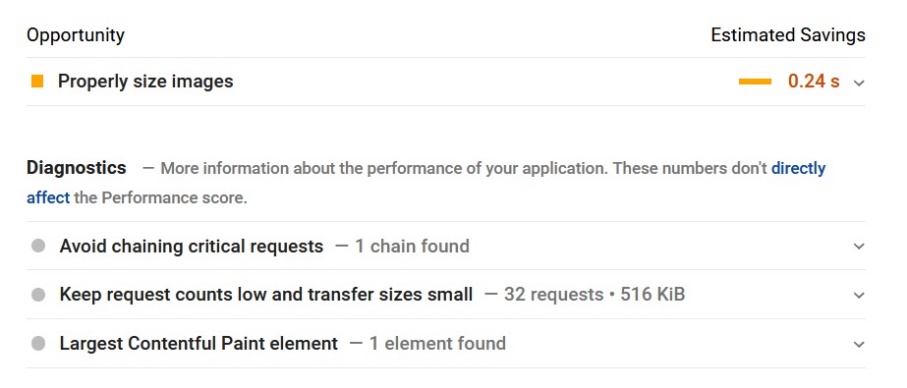
Posibles mejoras:



Versión escritorio



Posibles mejoras

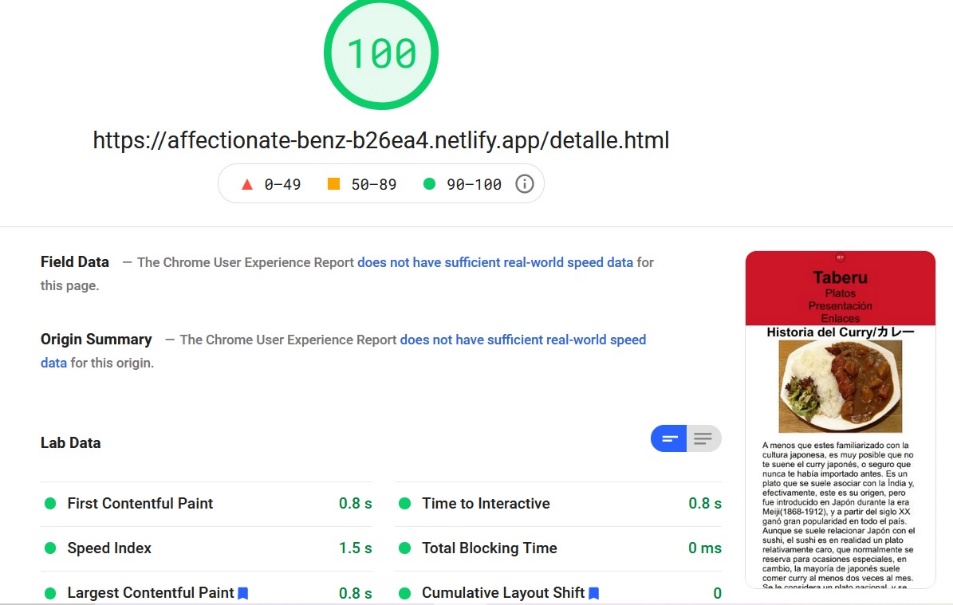


Ya no hay oportunidades para la versión móvil, y se ha conseguido reducir el CLS a 0. En general, puede apreciarse mejora en casi todos los campos de unos 0,1 segundos, excepto el LCP que en la versión móvil aumenta ligeramente.

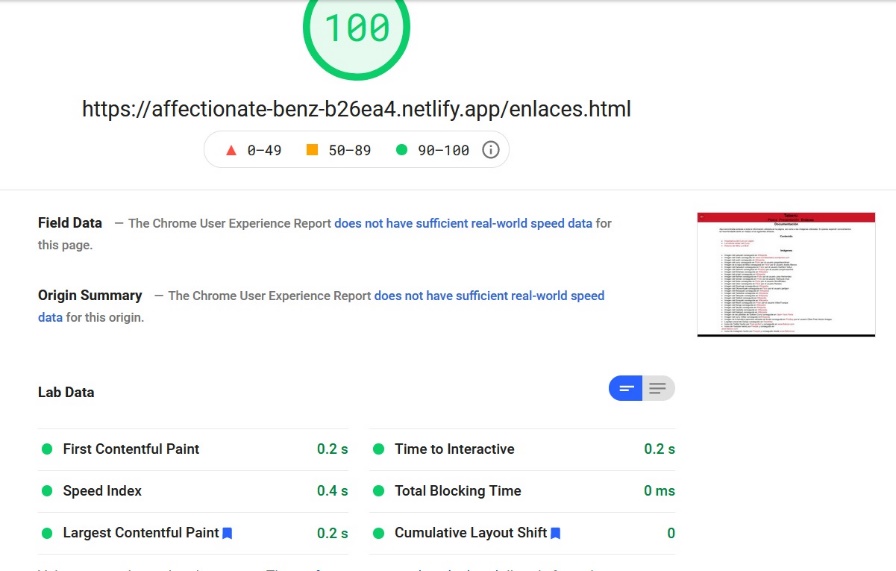
En la versión de escritorio, se está cargando una imagen demasiado grande para el tamaño que se necesita, se podría usar una más pequeña. Esto se debe a un error en el uso de imágenes responsive de la PEC2, la imagen de tamaño más grande era excesiva. El único cambio que hago es corregir esto.

Detalle.html

Versión móvil:



Versión escritorio:

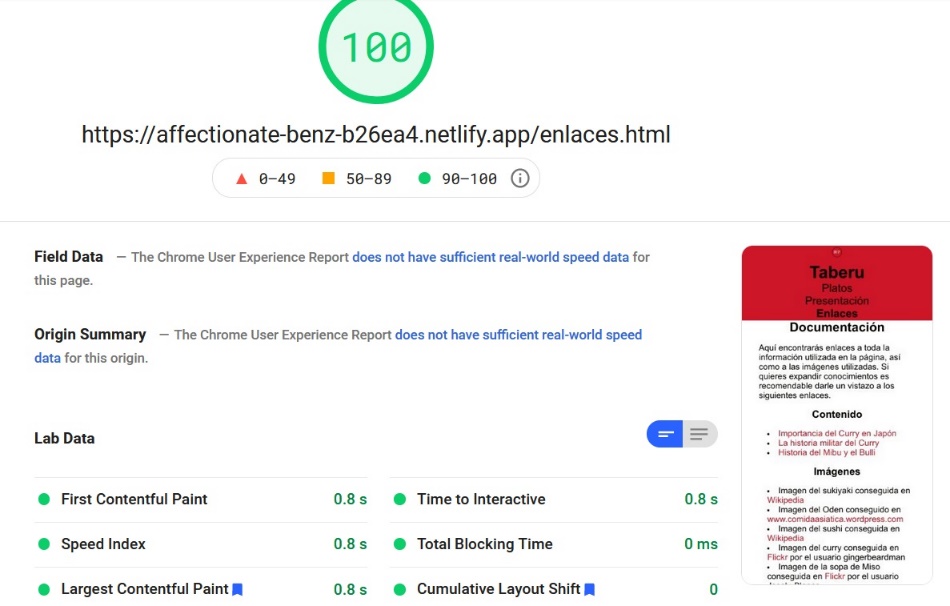


No adjunto las posibles mejores, son las mismas que en la versión móvil de la categoría. Por lo tanto, ya no aplicaré más cambios a esta página.

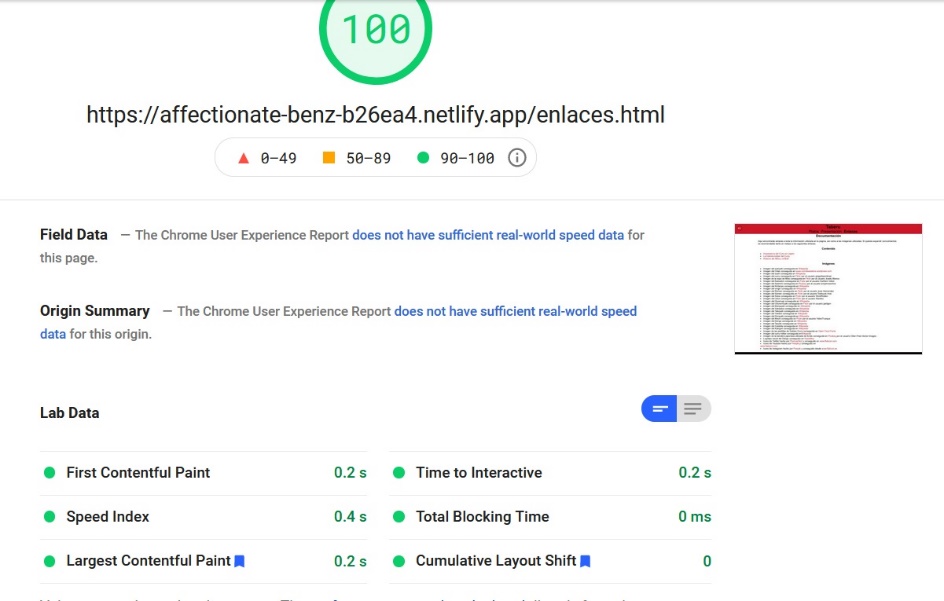
De nuevo se ve una clara mejora respecto al caso anterior, en especial en la versión escritorio. Hay pequeñas variaciones en los indicadores, por lo general se reducen(0,1 segundos), pero el cambio más evidente es en el CLS. Anteriormente la puntuación era mala, pero con los cambios se ha conseguido reducir a 0 en ambos casos.

Enlaces.html

Versión móvil



Versión de escritorio



De nuevo las posibles mejoras son idénticas al caso anterior y ya no realizo más cambios.

En esta página se ven pocos cambios. Realmente tenía poco que cambiar porque solo son enlaces. Las posibles mejoras eran principalmente por el header y footer. Con eso se ha conseguido que el CLS sea 0. Los indicadores son los mismos o hay variaciones de 0,1, principalmente en la versión escritorio donde todo baja menos el Speed Index que aumenta 0,1.