Xavier Ledesma Pons

Enlace al repositorio: https://github.com/xalepo4/HtmlAndCssl

Enlace web pública: https://youthful-darwin-c744cc.netlify.app

Tabla de carga antes de aplicar ningún cambio

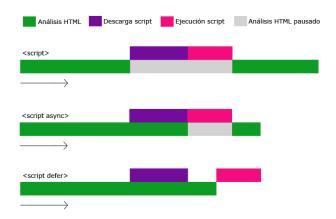
Antes de empezar, se han medido los parámetros solicitados para esta PEC y se ha realizado la siguiente tabla:

Título	URL (https://agitated-poincare-	Tiempo de carga	Peso total	Peso transferido	Cantidad de
	1e3e8f.netlify.app)	(s)	(KB)	(KB)	recursos
Home	/index.html	6,37	627,66	577,71	9
Categorías	/categories.html	6,4	702,73	653,45	9
Escalivada	/detail-escalivada.html	7,43	745,60	693,84	9
Esqueixada	/detail-esqueixada.html	7,85	744,84	693,53	9
Trinxat	/detail-trinxat.html	11,79	745,21	693,03	9
Fricandó	/detail-fricando.html	12,19	745,56	693,42	9
Presentación	/presentation.html	3,46	372,85	319,89	7
Enlaces	/links.html	5,12	450,75	402,35	6

Preguntas

- Al aplicar lazy loading a las imágenes de mi web, observo que cuando se hace la petición no se descargan todos los recursos, sino que solo se descargan aquellos que son visibles. Conforme hago scroll, veo que los recursos que pasan a estar visibles se descargan. Esto afecta al performance de la web, haciendo que en primera instancia esta cargue mas rápido, ya que se va descargando el contenido conforme va siendo necesario.
- 2) Al aplicar carga asíncrona sobre los scripts de la pagina, forzamos a que si tenemos el apartado scripts en el <head>, no se quede bloqueada la carga hasta que se descargue el script sino que la descarga se realiza en segundo plano y cuando esta finaliza, si usamos la opción async, se ejecuta el código bloqueando la ejecución de carga del HTML. Si usamos la opción de carga defer, se realiza la descarga en segundo plano y posteriormente, cuando se acaba de cargar el HTML se ejecuta el script.

Se adjunta imagen explicativa de este proceso:

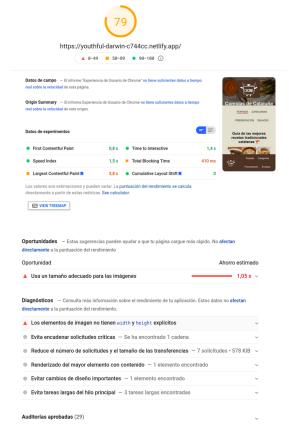


Uno de los problemas principales de realizar una descarga de **Javascript** asíncrona reside en casos como por ejemplo en mi web, que se anima un elemento que hasta que no esta analizado el **HTML**, no existe. Si el código **Javascript** se descarga y justamente se ejecuta antes de que el elemento **HTML** sea visible, el script fallará.

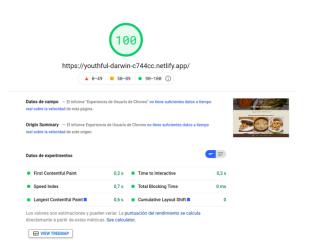
3) Se podría hacer carga asíncrona de estilos siempre y cuando no fueran estilos críticos que afectaran a la porción de pagina que se visualiza inicialmente. Uno de los problemas que podrían suceder es que, al aplicar una clase en un elemento, no se hubieran todavía cargado las hojas de estilos y, por tanto, durante una porción de tiempo, no se visualizara correctamente el contenido. Según he podido leer en diversas documentaciones, existe la posibilidad de definir los estilos críticos en el propio HTML usando el atributo style y haciendo que los estilos no críticos de carguen asíncronamente, reduciendo así el tiempo de carga del sitio web. Por otro lado, se podría cargar una hoja de estilos con los estilos mas críticos de manera tradicional y otra con los elementos menos críticos de manera asíncrona.

Informe primera iteración

Home dispositivo móvil

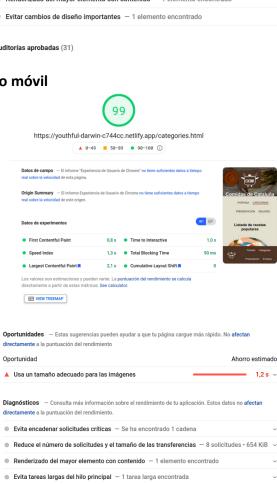


Home escritorio



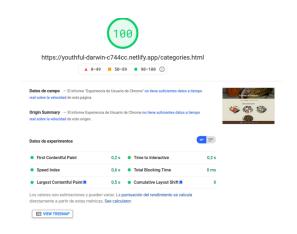


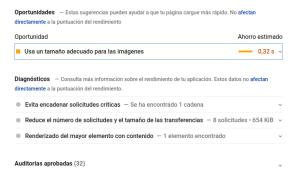
Categorías dispositivo móvil



Categorías escritorio

Auditorías aprobadas (31)

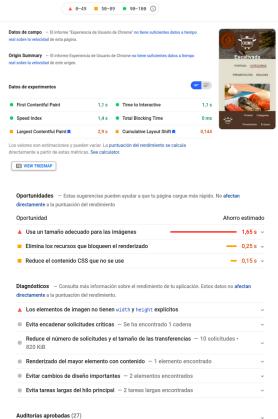




Escalivada dispositivo movil



https://youthful-darwin-c744cc.netlify.app/detail-escalivada.html



Escalivada escritorio



 $https://youthful\hbox{-}darwin\hbox{-}c744cc.net lify.app/detail\hbox{-}escalivada.html$

▲ 0-49 ■ 50-89 ● 90-100 ①

real sobre la velocidad de esta página.

Origin Summary — El informe Experiencia de Usuario de Chrome no tiene suficientes datos a tiempo real sobre la velocidad de este origen.

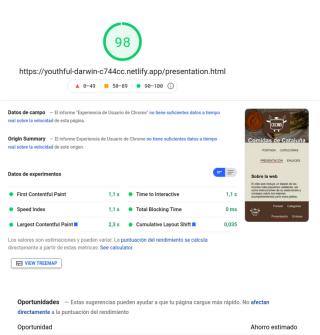


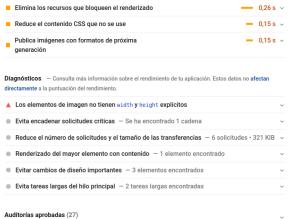
Datos de experimentos			==
First Contentful Paint	0,3 s	Time to Interactive	0,3 s
Speed Index	0,6 s	Total Blocking Time	0 ms
■ Largest Contentful Paint ■	0,4 s	■ Cumulative Layout Shift	0,182

Los valores son estimaciones y pueden variar. La puntuación del rendimier directamente a partir de estas métricas. See calculator.

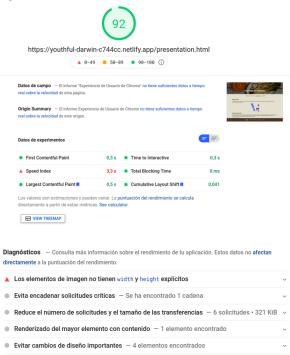
■ VIEW TREEMAP

Presentación dispositivo móvil





Presentación escritorio



Enlaces dispositivo móvil

Auditorías aprobadas (31)



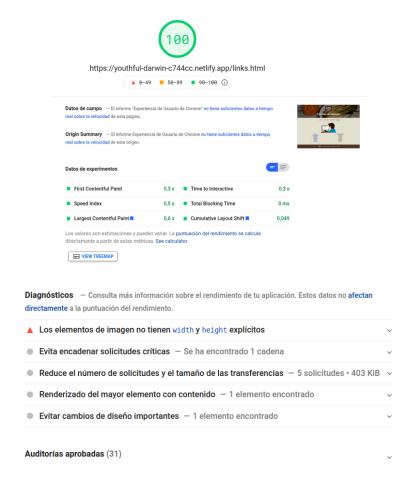
94

Oportunidades — Estas sugerencias pueden ayudar a que tu página cargue más rápido. No afectar directamente a la puntuación del rendimiento

Ahorro estimado



Enlaces escritorio



Análisis de las mejoras aplicadas sobre la web

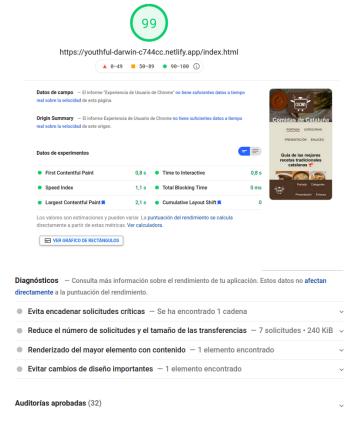
Una vez ejecutado el análisis, he aplicado las siguientes mejoras:

- Aplicar el atributo width y height a todas las imágenes tal y como se indica en el apartado de oportunidades. Con ello, se pretende reservar el espacio necesario para poner las imágenes cuando se lee inicialmente el código HTML.
- Para la página de categorías, se usaban las mismas imágenes que en el apartado de detalle, pero en una versión bastante más pequeña. Según el reporte, estas imágenes deberían ser mas pequeñas para reducir el tiempo de carga. Para ello, he creado una versión especial con el tamaño especifico que van a tener las imágenes en el apartado de categoría y en el apartado de otras recetas de las vistas de detalle. De esta manera, se descargan imágenes de menor tamaño y se usa menos ancho de banda y menor tiempo.
- En al apartado de enlaces, se usaba un GIF de un tamaño bastante grande que luego se escalaba por CSS. Para ahorrar tamaño de descarga, he reducido la imagen a un tamaño óptimo de mucho menor peso y mayor velocidad de descarga.

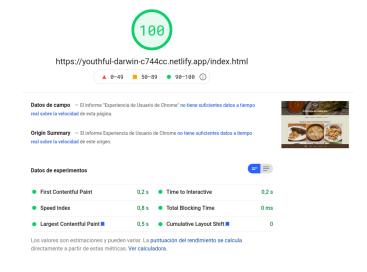
- Las imágenes de tamaño pequeño que se usaban para dispositivos móviles en el apartado de detalles eran muy grandes y realmente en dispositivos móviles nunca se mostraban con su tamaño real. Para ello, he aplicado una reducción del 50%, haciendo que el tamaño se reduzca y la descarga sea más rápida.
- La imagen que se mostraba en la portada, no tenia una versión responsive para dispositivos móviles. Para ello, he modificado el fichero index.html añadiendo la característica de cargar la imagen que toque en función de la resolución. También, he añadido la imagen con un formato reducido.
- La imagen que se mostraba en la portada en versión de escritorio era muy grande y se ha reducido un 25% para ahorrar tamaño y tiempo de descarga.
- En la versión evaluada, todos los ficheros HTML incluían una hoja de estilos llamada main.scss. Esta hoja, importaba todos los partials de los layouts, haciendo que todas las páginas descargaran siempre todo el CSS del sitio web. Como se usa una librería CSS llamada fontawesome, el fichero CSS que se descargaba cada vez era bastante pesado y afectaba al tiempo de carga. Para solucionarlo, he eliminado el fichero main.scss y he hecho que cada página solo incluya su hoja de estilos con los estilos que va a usar. De esta forma, solo la página principal (que es la que usa fontawesome) descarga sus estilos. Como se usa scss, el proceso ha sido bastante sencillo, ya que solo he tenido que convertir los partials en hojas de estilo e importar los partials de shared que se usan en cada página.
- La imagen del header que se carga en cada página era demasiado grande y he aplicado una reducción de su tamaño para hacer que el tiempo de carga sea menor en cada página y mejorar la puntuación.
- En uno de los apartados de los reportes, se hace recomienda usar formatos de nueva generación como JPG o JPEG y substituir los PNG. Para solucionarlo, el mapa del apartado presentación lo he convertido de formato.

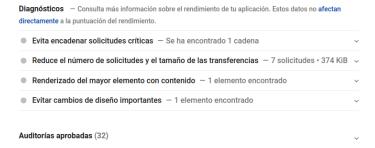
Informe segunda iteración

Home dispositivo móvil

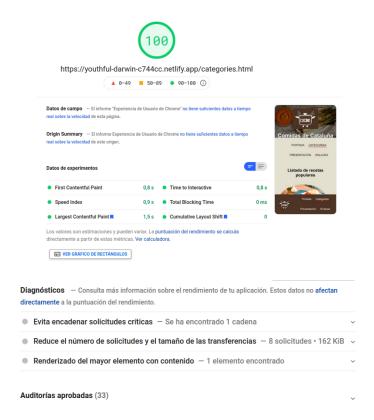


Home escritorio

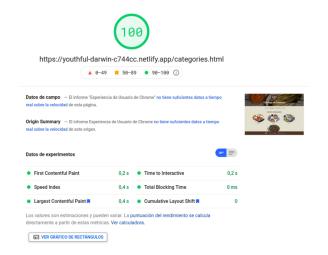


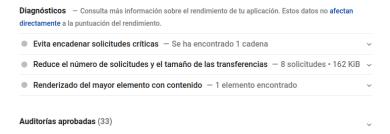


Categorías dispositivo móvil

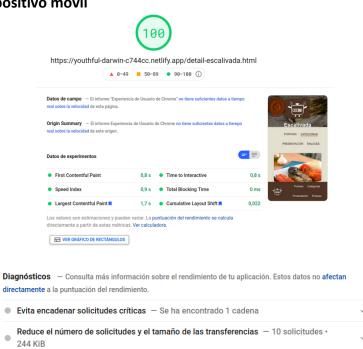


Categorías escritorio



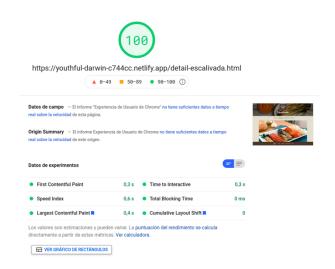


Escalivada dispositivo móvil



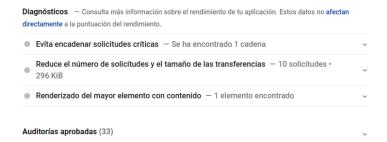
Escalivada escritorio

Auditorías aprobadas (32)

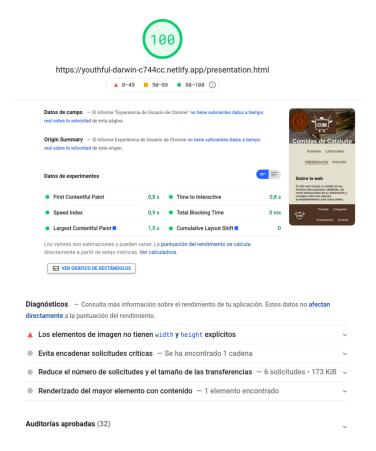


■ Renderizado del mayor elemento con contenido − 1 elemento encontrado

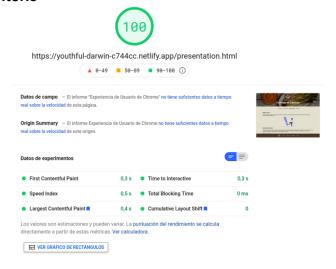
■ Evitar cambios de diseño importantes — 1 elemento encontrado



Presentación versión móvil



Presentación escritorio



 Diagnósticos
 — Consulta más información sobre el rendimiento de tu aplicación. Estos datos no afectan directamente a la puntuación del rendimiento.

 ▲ Los elementos de imagen no tienen width y height explícitos
 ✓

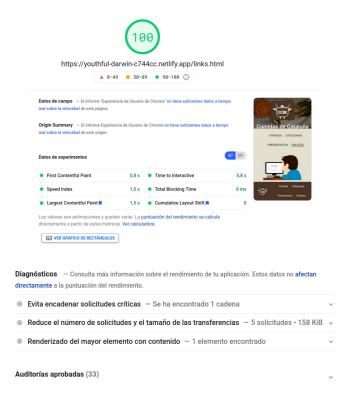
 ● Evita encadenar solicitudes críticas
 — Se ha encontrado 1 cadena
 ✓

 ● Reduce el número de solicitudes y el tamaño de las transferencias
 — 6 solicitudes * 173 KIB
 ✓

 ● Renderizado del mayor elemento con contenido
 — 1 elemento encontrado
 ✓

 Auditorías aprobadas (32)
 ✓

Enlaces versión móvil



Enlaces escritorio



https://youthful-darwin-c744cc.netlify.app/links.html



Diagnósticos — Consulta más info directamente a la puntuación del rendi	rmación sobre el rendimiento de tu aplicación. Estos datos no afectan imiento.	
 Evita encadenar solicitudes c 	ríticas — Se ha encontrado 1 cadena	~
Reduce el número de solicitudo	des y el tamaño de las transferencias -5 solicitudes \cdot 158 KiB	~
 Renderizado del mayor elemento 	ento con contenido — 1 elemento encontrado	~
Auditorías aprobadas (33)		~

Análisis segunda iteración

Tal y como se puede observar, se han resuelto muchos de los puntos que se sugerían en el reporte de la primera iteración. En todos los apartados se ha obtenido una puntuación de 100 menos en la página principal que en versión móvil se ha obtenido 99.

En esta segunda iteración, observando los resultados obtenidos no se ha realizado ninguna modificación, ya que se han implementado de golpe todas las mejoras que se han considerado oportunas.

Una vez realizadas las mejoras, se realiza una segunda tabla para analizar el cambio en los datos.

Título	URL (https://youthful-darwin-	Tiempo de carga	Peso total	Peso transferido	Cantidad de
	c744cc.netlify.app)	(s)	(KB)	(KB)	recursos
Home	/index.html	3,55	423,29	372,80	8
Categorías	/categories.html	1,68	165,00	160,46	9
Escalivada	/detail-escalivada.html	2,52	220,89	214,51	6
Esqueixada	/detail-esqueixada.html	2,85	252,78	250,45	6
Trinxat	/detail-trinxat.html	2,88	234,88	231,78	6
Fricandó	/detail-fricando.html	2,84	196,13	193,42	6
Presentación	/presentation.html	1,92	127,20	122,53	6
Enlaces	/links.html	2,12	159,31	156,43	6

Tal y como podemos observar, al aplicar mejoras como no descargar CSS innecesario y reducir el tamaño de las imágenes, los pesos totales y transferidos se han visto bastante reducidos. Los tiempos de carga, al aplicar **lazy loading**, son mucho mejores ya que solo descargamos el contenido que se visualiza en el **viewport** y conforme hacemos scroll, se va incrementando porque se descargan nuevos contenidos.

Anexo

En la primera PEC, me equivoque y use **UOC** boilerplate en vez de uno creado de cero tal y como se pedía. En la entrega de la **PEC2**, ya corregí esto y por tanto la modificación se ha visto propagada también a esta **PEC**. También corregí el error de accesibilidad que cometí en la **PEC1**. En la documentación de la **PEC2** incluí un apartado donde realicé el análisis de las transformaciones que faltaban en la **PEC1**.