

Herramientas HTML y CSS I

PEC 3

Documentación de la actividad

Alberto Carreiras González

Enero 2026

Enlace GitHub: <https://github.com/acarreiras/HerramientasHTMLyCSSI.git>

Enlace publicación PRO: <https://acarreiraspec1.netlify.app/>

ÍNDICE

Tabla y análisis del tiempo de carga.....	3
Página de inicio.....	3
Página destinos	3
Página Santiago de Compostela	4
Página Playas de la Catedral.....	4
Página Enlaces	4
Página de inicio.....	5
Página destinos	5
Página Santiago de Compostela	6
Página Playas de la Catedral.....	6
Página Enlaces	6
Página Santiago de Compostela	7
Página Playas de la Catedral.....	8
Cambios de lazy loading y carga asíncrona	8
Mejoras.....	9
Versión móvil	9
Versión escritorio.....	9
Preguntas	10
¿Qué cambios detectas en las herramientas para desarrolladores al aplicar lazy loading a las imágenes de tu web? ¿Cómo crees que afecta al rendimiento de tu página?	10
Al aplicar <i>lazy loading</i> a las imágenes se observa que las imágenes que no están visibles inicialmente no se descargan durante la carga inicial de la página. Estas solicitudes se realizan únicamente cuando el usuario hace scroll y la imagen entra en el área visible.....	10
¿Qué sucede al aplicar carga asíncrona a los scripts de tu página? ¿Qué problemas crees que podrían surgir si cargas el JavaScript de forma asíncrona? Ten en cuenta los diferentes métodos de carga asíncrona para responder esta pregunta.....	10
No hemos hecho carga asíncrona de estilos. ¿Crees que se podría hacer? ¿Qué problemas podríamos tener? Razona tu respuesta.	10

Tabla y análisis del tiempo de carga

Se han realizado un total de 7 mediciones de cada una de las páginas relevantes del sitio web. Posteriormente, se ha sacado el promedio de dichas mediciones eliminando, previamente, el valor de carga más alto y el más bajo.

Estos tiempos de carga se han obtenido utilizando el evento *finish* registrado en la pestaña *Network* de las herramientas de desarrollo de Chrome, con la caché desactivada y emulación de red 3G.

A continuación, se detallan los resultados de los diferentes sitios, así como sus mediciones parciales antes de aplicar cualquier cambio.

Página de inicio

Título de la página	Conoce Galicia
URL	https://acarreiraspec1.netlify.app/
Tiempo de carga (promedio)	26,86
Peso total	1,1 MB
Peso transferido	1,0 MB
Recursos que contiene la página	11

Número de medición	Tiempo
1	26,89s
2	26,93s
3	26,81s
4	26,90s
5	26,80s
6	26,92s
7	26,80s

Página destinos

Título de la página	Destinos Conoce Galicia
URL	https://acarreiraspec1.netlify.app/destinos
Tiempo de carga (promedio)	1,2min
Peso total	3,4 MB
Peso transferido	3,4 MB
Recursos que contiene la página	13

Número de medición	Tiempo
1	57,2s
2	1,2min
3	1,2min
4	1,3min
5	1,2min

6	1,2min
7	1,2min

Página Santiago de Compostela

Título de la página	Santiago de Compostela Conoce Galicia
URL	https://acarreiraspec1.netlify.app/santiago
Tiempo de carga (promedio)	34,62s
Peso total	5,6 MB
Peso transferido	2,6 MB
Recursos que contiene la página	34

Número de medición	Tiempo
1	35,2s
2	2,5min
3	34,74s
4	34,41s
5	34,33s
6	34,34s
7	34,46s

Página Playas de la Catedral

Título de la página	Playas de la Catedral Conoce Galicia
URL	https://acarreiraspec1.netlify.app/catedrales
Tiempo de carga (promedio)	54,78s
Peso total	6,0 MB
Peso transferido	3,0 MB
Recursos que contiene la página	33

Número de medición	Tiempo
1	59,35s
2	42,95s
3	42,85s
4	1,5min
5	42,97s
6	42,81s
7	42,98s

Página Enlaces

Título de la página	Conoce Galicia
URL	https://acarreiraspec1.netlify.app/links
Tiempo de carga (promedio)	2,1min
Peso total	6,1 MB
Peso transferido	6,1 MB

Recursos que contiene la página	19
---------------------------------	----

Número de medición	Tiempo
1	2,1min
2	2,1min
3	2,1min
4	2,1min
5	2,1min
6	2,3 min
7	2,1min

A continuación, se detallan los nuevos tiempos de carga de cada una de las páginas con los cambios de *lazy loading* y carga asíncrona aplicados:

Página de inicio

Título de la página	Conoce Galicia
URL	https://acarreiraspec1.netlify.app/
Tiempo de carga (promedio)	25,16s
Peso total	1,1 MB
Peso transferido	1,0 MB
Recursos que contiene la página	11

Número de medición	Tiempo
1	41,87s
2	24,76s
3	24,76s
4	24,75s
5	24,76s
6	26,77s
7	24,76s

Página destinos

Título de la página	Destinos Conoce Galicia
URL	https://acarreiraspec1.netlify.app/destinos
Tiempo de carga (promedio)	39,036s
Peso total	1,7 MB
Peso transferido	1,7 MB
Recursos que contiene la página	12

Número de medición	Tiempo
1	39,01s
2	39,07s
3	39,02s

4	39,09s
5	39,04s
6	39,04s
7	39,00s

Página Santiago de Compostela

Título de la página	Santiago de Compostela Conoce Galicia
URL	https://acarreiraspec1.netlify.app/santiago
Tiempo de carga (promedio)	35,494s
Peso total	5,0 MB
Peso transferido	2,0 MB
Recursos que contiene la página	34

Número de medición	Tiempo
1	35,41s
2	1,1min
3	35,54s
4	36,09s
5	35,61s
6	34,69s
7	35,54s

Página Playas de la Catedral

Título de la página	Playas de la Catedral Conoce Galicia
URL	https://acarreiraspec1.netlify.app/catedrales
Tiempo de carga (promedio)	35,85s
Peso total	5,1 MB
Peso transferido	2,1 MB
Recursos que contiene la página	33

Número de medición	Tiempo
1	35,41s
2	35,96s
3	36,04s
4	36,10s
5	35,69s
6	35,63s
7	35,93s

Página Enlaces

Título de la página	Conoce Galicia
URL	https://acarreiraspec1.netlify.app/links
Tiempo de carga (promedio)	52,62s

Peso total	2,3 MB
Peso transferido	2,3 MB
Recursos que contiene la página	11

Número de medición	Tiempo
1	52,50s
2	52,57s
3	52,65s
4	52,62s
5	52,56s
6	52,70s
7	52,71s

Las mejoras que se detectan una vez aplicado los cambios son que los tiempos se reducen algo, aunque no de manera muy significativa y que los recursos que se descargan son aquellos necesarios para el viewport. A medida que vamos realizando scroll observamos que se van descargando nuevos recursos.

Al tratarse de páginas de poco contenido, no se aprecia de manera significativa, destacando más en aquellas secciones con más recursos gráficos.

Tras las comprobaciones en Google *PageSpeed Insights* se observa que hay que realizar cambios en dos páginas, *santiago.html* y *catedrales.html*. En el resto de los ficheros no se cambia nada por lo que no modifica su medición. A continuación, se detallan los nuevos tiempos en las páginas mencionadas:

Página Santiago de Compostela

Título de la página	Santiago de Compostela Conoce Galicia
URL	https://acarreiraspec1.netlify.app/santiago
Tiempo de carga (promedio)	35,132s
Peso total	5,0 MB
Peso transferido	2,0 MB
Recursos que contiene la página	34

Número de medición	Tiempo
1	36,37s
2	35,74s
3	35,43s
4	33,80s
5	35,30s
6	35,30s
7	35,39s

Página Playas de la Catedral

Título de la página	Playas de la Catedral Conoce Galicia
URL	https://acarreiraspec1.netlify.app/catedrales
Tiempo de carga (promedio)	35,60s
Peso total	5,1 MB
Peso transferido	2,1 MB
Recursos que contiene la página	33

Número de medición	Tiempo
1	35,78s
2	35,98s
3	35,55s
4	35,43s
5	35,54s
6	35,66s
7	35,48s

Se produce una ligerísima mejora en los tiempos de carga una vez aplicado los cambios.

Cambios de lazy loading y carga asíncrona

Se aplica *lazy loading* únicamente a imágenes que no forman parte del *viewport* inicial de cada una de las páginas. En el caso de la página de inicio, se excluye la imagen principal del bloque *hero* dado que se trata de un elemento crítico. Sin embargo, se aplica en el resto de las imágenes que son todas aquellas que están incluidas en la sección de destinos destacados.

En la página *destinos.html* y *link.html*, se aplica *lazy loading* en el total de las imágenes que se usan. La razón es que ninguna de ellas forma parte del contenido visible del *viewport* inicial.

En la página de *santiago.html* y *catedrales.html* se ha aplicado *lazy loading* a todas las imágenes de contenido y al *iframe* de YouTube, ya que estos elementos no forman parte del *viewport* inicial. La imagen principal del bloque *hero* no se ha cargado de forma diferida al tratarse de un elemento crítico. En esta página se utiliza un archivo JS externo, para evitar un retraso en la carga del documento se ha aplicado el atributo *defer* al script.

Aunque el tiempo total de carga no presenta mejoras significativas en todas las páginas, la aplicación de *lazy loading* y la carga asíncrona permite optimizar la carga inicial de recursos, evitando así descargas innecesarias y priorizando el contenido visible.

Este efecto es más relevante en páginas con mayor número de recursos gráficos, donde la carga diferida contribuye a una distribución más eficiente del uso de la red.

Mejoras

Versión móvil

Página	Rendimiento	Accesibilidad	Prácticas Recomendadas	SEO
Inicio	98	100	100	100
Destinos	99	100	100	100
Santiago	57	100	77	100
Catedrales	58	100	77	100
Enlaces	86	100	100	100

Versión escritorio

Página	Rendimiento	Accesibilidad	Prácticas Recomendadas	SEO
Inicio	100	100	100	100
Destinos	100	100	100	100
Santiago	85	100	77	100
Catedrales	86	100	77	100
Enlaces	99	100	100	100

En líneas generales se observa que la aplicación web está bien desarrollada, siendo penalizada las dos páginas de contenido detallado de los destinos Santiago y Catedrales.

Según *Google PageSpeed Insights* se detecta imágenes con resoluciones excesivas. Se soluciona ajustando el atributo `srcset` eliminando las imágenes de 1200px. Esta mejora reduce el peso total transferido sin afectar a la calidad visual. Esta mejora se aplica en las páginas *santiago.html* y *catedrales.html*.

Además, se añaden los atributos `width` y `height` a las imágenes para definir su relación de aspecto y evitar cambios de diseño durante la carga (CLS). Los valores aplicados mantienen la proporción original de las imágenes y no afectan al comportamiento responsive del sitio.

El resto de los cambios que indica *Google PageSpeed Insights* responde a carácter informativo dado que la compilación del `css` ya se realiza *Netlify* y no se puede optimizar. Con estos cambios se mejora levemente el rendimiento de las páginas *santiago.html* y *catedrales.html*.

Preguntas

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

Al aplicar *lazy loading* a las imágenes se observa que las imágenes que no están visibles inicialmente no se descargan durante la carga inicial de la página. Estas solicitudes se realizan únicamente cuando el usuario hace scroll y la imagen entra en el área visible.

Con ello, se logra reducir el número de peticiones iniciales, se disminuya la cantidad de datos transferidos y por lo tanto mejore el tiempo de carga de la página.

¿Qué sucede al aplicar carga asíncrona a los scripts de tu página? ¿Qué problemas crees que podrían surgir si cargas el JavaScript de forma asíncrona? Ten en cuenta los diferentes métodos de carga asíncrona para responder esta pregunta.

Al aplicar carga asíncrona el navegador no bloquea el renderizado del HTML mientras se descarga el JS lo que mejora el tiempo de carga de la página.

Los problemas que se pueden obtener son los siguientes:

- El script puede ejecutarse tan pronto como esté descargado sin respetar el orden de carga, lo que puede provocar errores si depende de otros scripts aún no descargados.
- El script puede necesitar acceder a elementos del DOM que aún no han sido renderizados y provocar errores de ejecución.

Por ello, es sumamente importante tener en cuenta las dependencias del código JS para elegir un método de carga asíncrona adecuado.

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

Sí que se podría realizar, de todas formas, no se suele recomendar para sitios clásicos. Los estilos son recursos muy críticos en el renderizado y configurar una carga asíncrona provocaría problemas visuales, mostrándose contenido sin estilo, cambios bruscos en el diseño del sitio y una peor experiencia de usuario, especialmente en dispositivos con conexiones lentas o limitadas.