

Nasapedia

Proyecto final de Ingeniería Web

Página web: http://eim-alu-69032.lab.unavarra.es/iw_group/grupo-pangolin/final_project/src/

Álvaro Larraya
Eugen Hamuraru
Rodrigo Andrés

[Introducción](#)

[Descripción y objetivo de “Nasapedia”](#)

[Público al que va orientado](#)

[Prototipado](#)

[Prototipos iniciales](#)

[Comparación con el producto final](#)

[Justificación de las diferencias](#)

[Detalles de análisis y diseño](#)

[Modelo de BBDD](#)

[Patrones de diseño](#)

[Diagrama UML](#)

[Detalles de implementación](#)

[API](#)

[Librerías y frameworks](#)

[Detalles de despliegue](#)

[Rendimiento](#)

[Pruebas realizadas](#)

[Resultados obtenidos](#)

[Página Home:](#)

[Página APOD \(“Astronomy picture of the day”\):](#)

[Página “Solar System's planet”:](#)

[Página “Compare the Earth”:](#)

[Página “Home”:](#)

[Modificaciones propuestas](#)

[SEO](#)

[Auditorías realizadas](#)

[Resultados obtenidos](#)

[Modificaciones propuestas](#)

[Accesibilidad](#)

[Auditorías realizadas](#)

[Resultados obtenidos](#)

[Modificaciones propuestas](#)

[Usabilidad](#)

[Auditorías realizadas](#)

[Resultados obtenidos](#)

[Modificaciones propuestas](#)

[Gestión de configuración](#)

[Uso de repositorios](#)

[Integración continua](#)

[Trabajo en equipo y reparto de responsabilidades](#)

[Conclusiones](#)

[Anexos](#)

Introducción

Descripción y objetivo de “Nasapedia”

Se trata de una web con la que poder acceder a información sobre el Sistema Solar. En nuestros apartados puedes encontrar: la imagen elegida foto del día por la NASA desde que empezaron a conceder este premio hasta el día de hoy; información sobre los planetas y satélites que componen el Sistema Solar junto con fotos de los mismos para añadir apoyo visual; así como un panel interactivo con el que poder comparar las dimensiones de la Tierra con otros planetas de su mismo sistema el cual permite moverte por el plano, cambiar la perspectiva y distancia desde la que verlos.

Ya que una foto dice más que mil palabras y no hay mejor forma de aprender que interactuando con el entorno.

Público al que va orientado

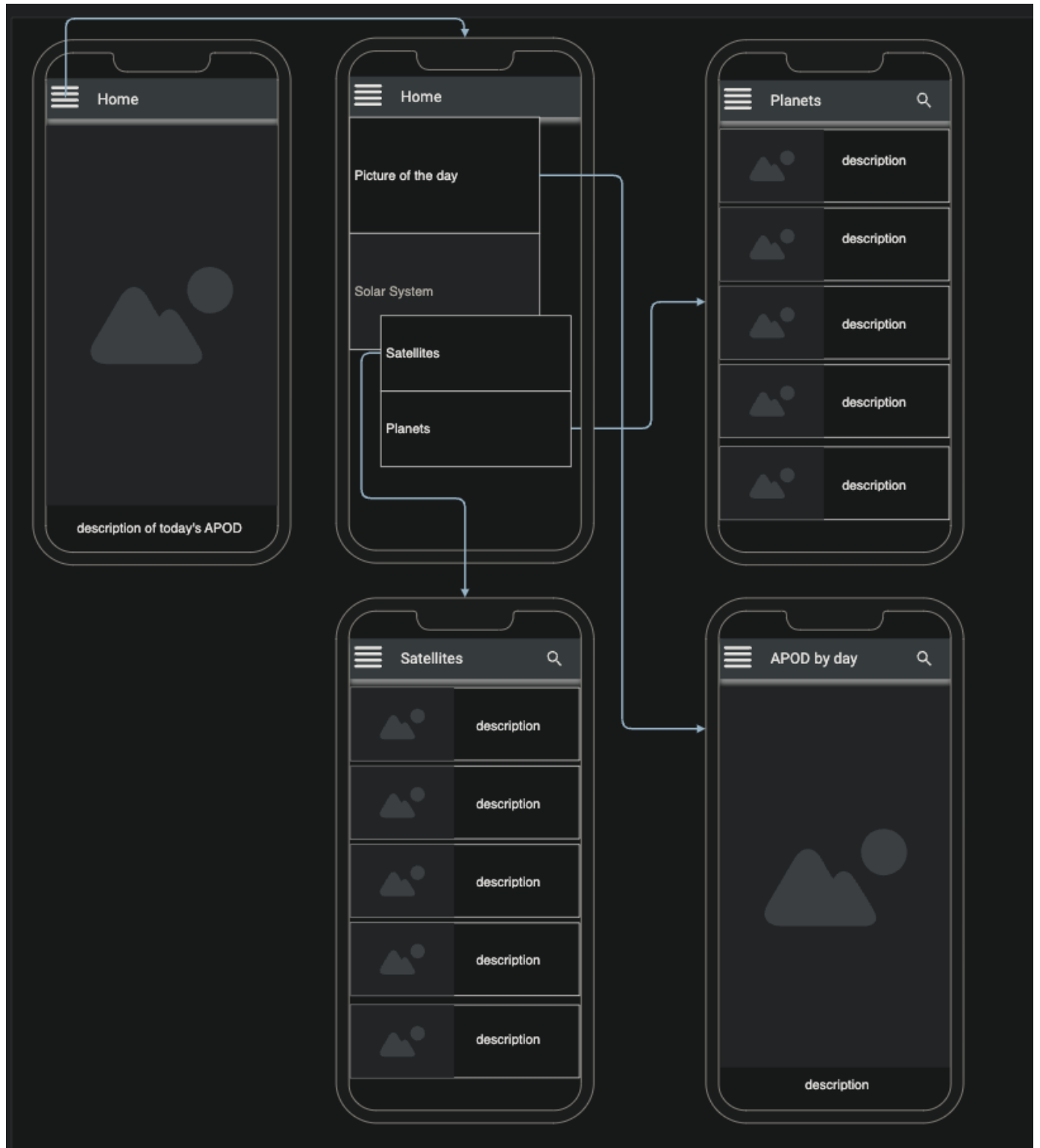
Cualquier persona que le interese el Sistema Solar, quiera nutrirse de conocimientos sobre él en un portal con información sintetizada, datos curiosos y una interfaz cuidada.

El lenguaje utilizado es el inglés por lo que la masa de gente que puede consumir la web es muy amplia. Además evitamos entrar en temas muy complejos por lo que la barrera técnica es prácticamente inexistente.

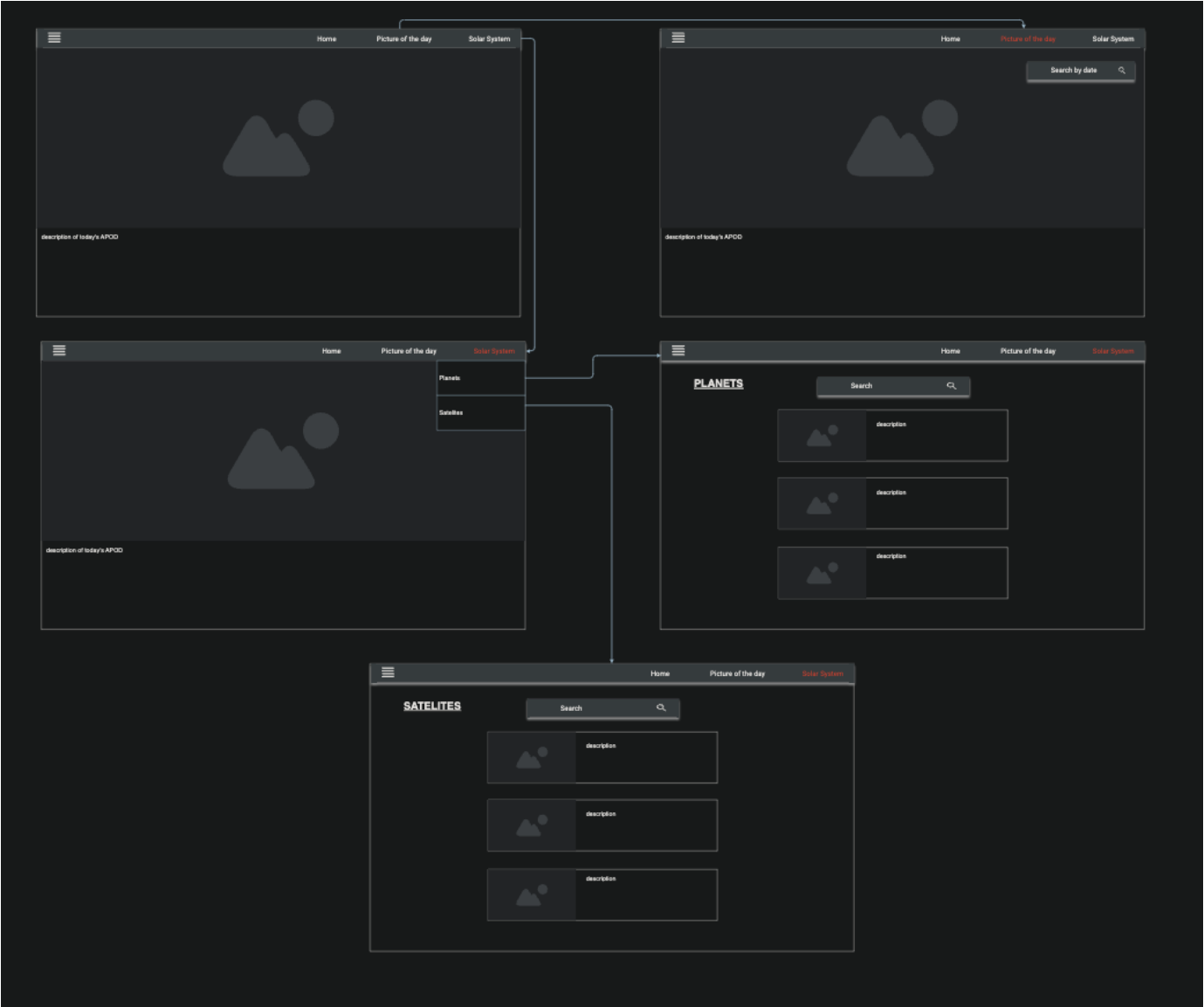
Prototipado

Prototipos iniciales

Móvil:



Escritorio:



Comparación con el producto final

En el prototipo no contemplamos la opción de hacer una página de inicio dedicada como la que tiene la web final ni una pestaña con la que poder comparar el tamaño de la Tierra.

Por otro lado durante el desarrollo hemos buscado diseños que hagan más atractivo e intuitivo el despliegue y los listados de planetas y satélites de forma que no sea simplemente una lista que vas recorriendo.

Por último la mayor diferencia que hay entre el prototipo y el producto final es la incorporación de una funcionalidad nueva en la página web que en principio no teníamos ni siquiera pensado implementar, esta es una pestaña que permite ver a escala y en tres dimensiones la comparativa de los diferentes planetas del sistema solar con la tierra.

Estos han sido en líneas generales los cambios más representativos pero no los únicos, también han habido otros cambios más sutiles fruto de haber tenido que adaptarnos a las circunstancias que se han dado durante el proyecto y que no eran posibles de prever hasta ponernos manos a la obra, algunos ejemplos de cambios menos importantes son, eliminar la barra de búsqueda del menú puesto que finalmente no le hemos visto gran utilidad o fusionar la pestaña de “picture of the day” con la otra pestaña de “picture of the day” que tiene opción de búsqueda por fecha en una sola pestaña dado que realmente no tenía mucho sentido separar dos funcionalidades tan parecidas.

Justificación de las diferencias

Al principio pensamos que no nos iba a dar tiempo a implementar tantas cosas pero cuando nos pusimos manos a la obra vimos que podíamos extender el alcance del proyecto, por eso decidimos añadir más funcionalidades que le aportan variedad y contenido a la web.

Detalles de análisis y diseño

Modelo de BBDD

Nuestra base de datos tiene un modelo muy simple con dos tablas:

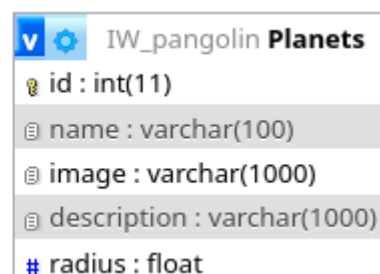
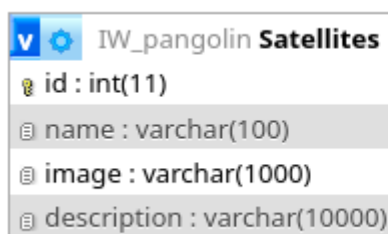
1. Planetas: con 5 columnas siendo estas nombre, id, imagen, descripción y radio. Radio no es como tal el radio en km por ejemplo sino el ratio de tamaño en comparación con la Tierra.
2. Satélites: con 4 columnas que son id, nombre, imagen y descripción.

Como Sistema Gestor de Base de Datos hemos usado MySQL

Patrones de diseño

Durante todo el trabajo hemos usado el patrón MVC, más concretamente con PageControllers. Para poder desacoplar funcionamientos, facilitarnos el mantenimiento en caso de errores y propiciar la escalabilidad. El singleton nos ha permitido solo acceder con una conexión a la BBDD y no colapsarla con peticiones innecesarias.

Diagrama UML



El diagrama UML es el que se puede apreciar en la imagen, es muy básico y ni siquiera existe algún tipo de relación entre tablas.

Detalles de implementación

API

La API que hemos usado es una oficial de la NASA que te permite buscar la foto del día por fechas, es oficial y no necesita token, además de ser totalmente gratis. El único impedimento que tiene es que solo se pueden hacer un máximo de peticiones por día, pero en caso de quedarnos cortos en un futuro podemos implementar una pestaña donde pedir un token nuevo.

Librerías y frameworks

Hemos hecho uso de Bootstrap para agilizar la programación de los css que hemos utilizado y para que sea más llamativa y atractiva visualmente la web.

También hemos aprovechado que la librería Three Js da tremendas facilidades para implementar potentes herramientas visuales controlables por el usuario y la hemos usado para hacer la pestaña de comparación de tamaños. Al ser javascript el lenguaje con el que está hecho permite no hacer cargas entre animaciones lo cual le da fluidez a la experiencia del visitante al portal.

Detalles de despliegue

Durante toda la fase de desarrollo cada uno de los miembros del equipo ha trabajado en su propio entorno de desarrollo y para poder ir probando los cambios nos hemos ayudado del servidor web Apache que ofrece XAMPP y de su base de datos MySQL.

El hecho de trabajar en entornos diferentes supuso tener que definir una serie de variables de entorno en el proyecto en un fichero “.env” de forma que cada miembro del equipo defina el valor de esas variables en su propio fichero y no haya que ir haciendo cambios en el código según el entorno en el que se ejecute el proyecto.

Algunas de estas variables de entorno han sido por ejemplo, las credenciales necesarias para realizar la conexión con la base de datos MySQL o definir las rutas absolutas en el ordenador local hasta la carpeta contenedora del proyecto.

Tener estos parámetros definidos como variables de entorno ha supuesto una gran ventaja a la hora de hacer el despliegue en el servidor centralizado ya que permite no tener que preocuparse por cambiar variables y rutas en el código de producción si no que con ajustar las variables del fichero “.env” es suficiente.

Rendimiento

Pruebas realizadas

Para llevar a cabo un estudio del rendimiento de la página nos ayudamos de las herramientas de desarrollador que ofrece el navegador y sometimos la página a diferentes condiciones para ver como respondía, para ello en primer lugar desactivamos la caché de forma que cada vez que refresquemos la página se vuelvan a hacer todas las peticiones necesarias y después fuimos limitando la conexión a internet para estudiar el rendimiento bajo esas circunstancias.

Resultados obtenidos

Página “Home”:

- Resultados Obtenidos

Inspector Consola Depurador Red Editor de estilos Rendimiento									
Filtrar las URL									
Desactivar caché Regular 4G / LTE									
Todos HTML CSS JS XHR Tipografía Imágenes Medios WS Otros									
Estado	Método	Dominio	Archivo	Iniciador	Tipo	Transferido	Tamaño	0 ms	
301	GET	localhost	pangolin	document	html	2,83 KB	2,52 ...	1896 ms	
200	GET	localhost	/pangolin/	document	html	2,52 KB	2,52 ...	1893 ms	
200	GET	localhost	index.css	stylesheet	css	498 B	498 B	40 ms	
200	GET	localhost	navbar.css	stylesheet	css	43 B	43 B	56 ms	
200	GET	cdn.jsdelivr.net	bootstrap.min.css	stylesheet	css	25,79 KB	152,1 ...	69 ms	
200	GET	code.jquery.com	jquery-3.2.1.slim.min.js	script	js	23,30 KB	67,97 ...	76 ms	
200	GET	cdn.jsdelivr.net	bootstrap.min.js	script	js	14,41 KB	47,80 ...	69 ms	
200	GET	localhost	background.avif	img		160,76 KB	160,7 ...	36 ms	
200	GET	localhost	favicon.ico	FaviconLoader...		30,17 KB	30,17 ...	22 ms	

La página principal “home” tiene un peso de 256 KB, más de la mitad del peso perteneciente a la imagen que tenemos de fondo (160 KB) , anteriormente utilizamos la

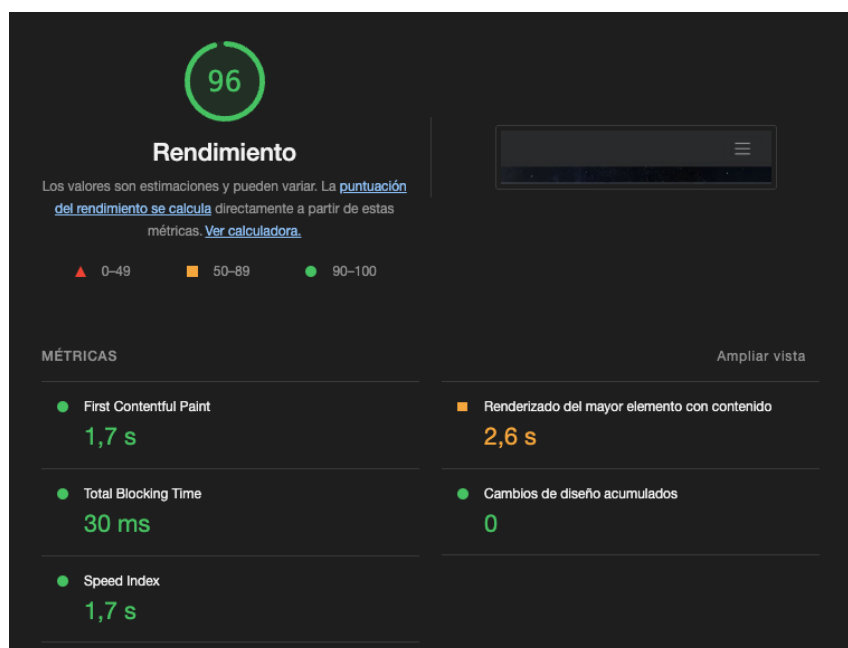
misma imagen en formato “.jpg” y esta ocupaba un total de 0.98 MB así que la mejora es significativa sobre todo de cara a usuarios con peor conexión a internet que utilicen la página web.

Los tiempos de carga de la página son los siguientes:

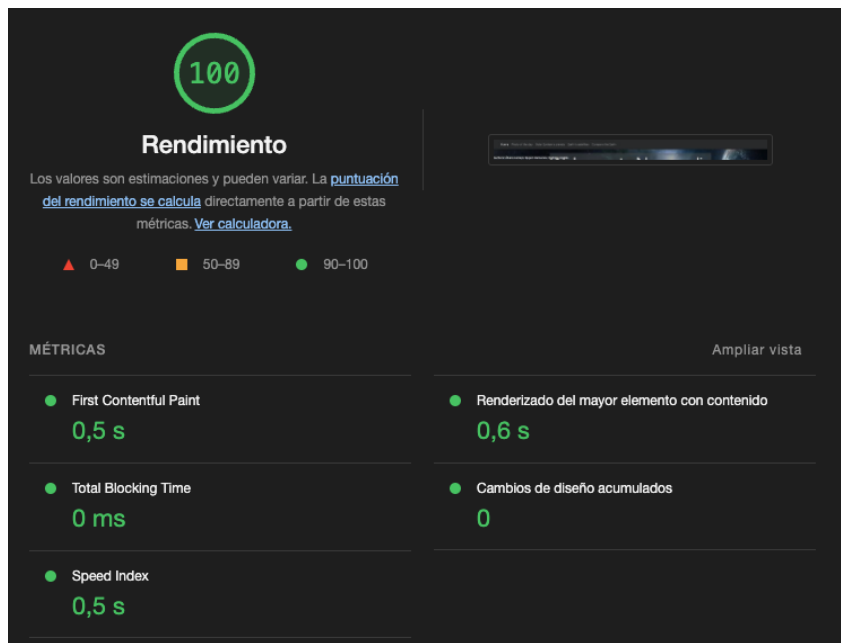
- Para 2G regular: 8.74 segundos
- Para 3G regular: 2.56 segundos
- Para 4G: 299 ms.

Son tiempos de espera para el usuario aceptables y difíciles de mejorar sin perjudicar la interfaz de usuario.

Usando Lighthouse obtenemos una puntuación en móvil:



y en escritorio:



Página APOD (“Astronomy picture of the day”):

- Resultados obtenidos:

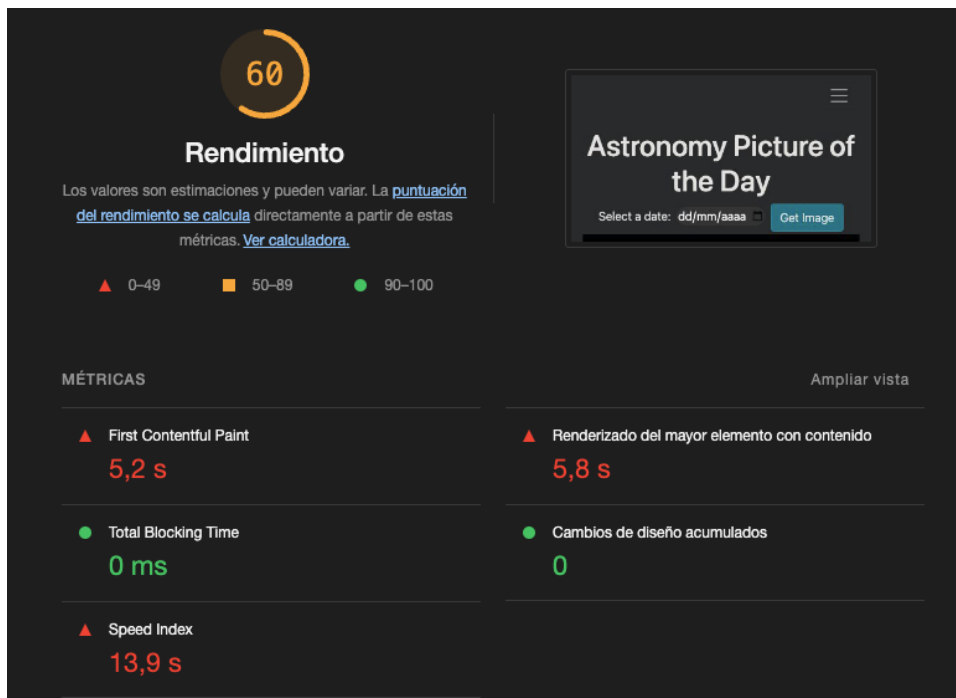
Estado	Método	Dominio	Archivo	Iniciador	Tipo	Transferido	Tamaño	Tiempo
200	GET	localhost	photoDayChooserController.php	BrowserTabC...	html	4,14 KB	4,14 ...	1207 ms
200	GET	localhost	style.css	stylesheet	css	433 B	433 B	302 ms
200	GET	localhost	navbar.css	stylesheet	css	43 B	43 B	306 ms
200	GET	cdn.jsdelivr...	bootstrap.min.css	stylesheet	css	25,05 KB	152,...	1403 ms
200	GET	code.jquery...	jquery-3.2.1.slim.min.js	script	js	23,30 KB	67,9...	411 ms
200	GET	cdn.jsdelivr...	bootstrap.min.js	script	js	13,66 KB	47,8...	1986 ms
200	GET	localhost	favicon.ico	FaviconLoad...		30,17 KB	30,1...	2278 ms
200	GET	apod.nasa....	freeflyer_nasa_960.jpg	img	jpeg	55,50 KB	55,5...	2644 ms

El peso total de la página era de 152K y ha tardado 5.8 segundos en cargarse completamente en condiciones de internet de 2G con un estado de conexión regular.

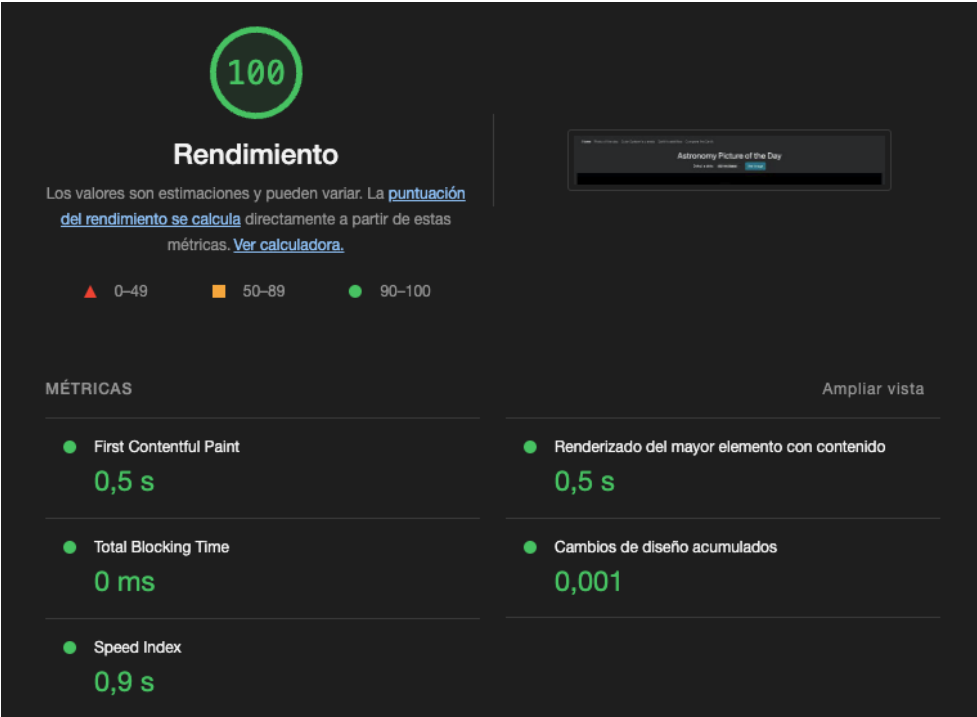
Posteriormente hicimos la prueba en las mismas condiciones pero con una conexión 3G regular y la página quedaba totalmente cargada en 2 segundos.

Realmente desde el punto de vista del usuario la página responde de forma relativamente rápida incluso con conexiones regulares de 2G o 3G, en cualquier caso el cuello de botella a la hora de cargar la página es la imagen, la cual supone una tercera parte de la carga de toda la página, esta es proporcionada directamente por la api y no podemos definir en la llamada a la api el tamaño de imagen que queremos, es por ello que estamos obligados a traernos la imagen en su tamaño original y luego a hacer un resize de la misma desde css.

Usando Lighthouse obtenemos una puntuación en móvil:



y en escritorio:



Página “Solar System's planet”:

- Resultados obtenidos:

Inspector

Consola

Depurador

Red

Editor de estilos

Rendimiento

>>

...

X

Filtrar las URL

La página de planetas del sistema solar es la página con mayor peso de toda la página web, llegando a pesar esta 3,7 MB, interesa especialmente estudiarla ya que será la que más tiempo tarda en cargar y la que más problemas de rendimiento puede generar.

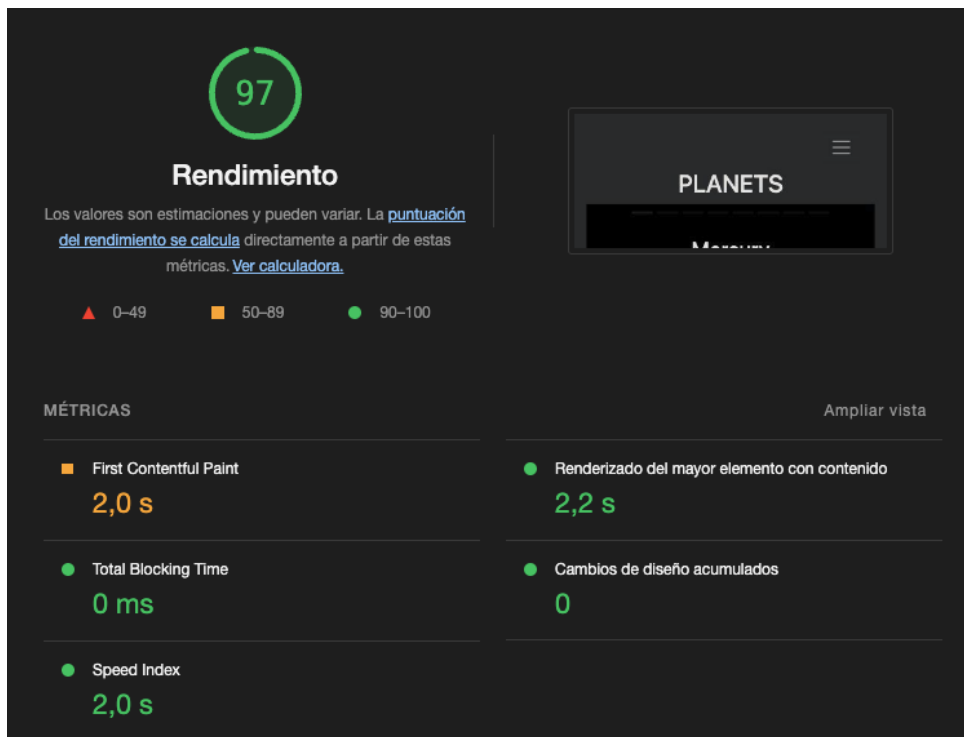
En una primera prueba con el internet limitado a 2G la página llegó a tardar 1,12 minutos en ser totalmente cargada, es un tiempo realmente alarmante y puede generar una horrible experiencia de usuario, a pesar de ello era esperable que esto ocurra ya que se trabaja con un gran número de imágenes de gran resolución que son las que suponen casi el total del peso de la página.

En una segunda prueba realizada con el internet limitado a 3G la página tardó 19 segundos en cargar totalmente, un tiempo que sigue siendo muy elevado teniendo en cuenta que un usuario promedio no está dispuesto a esperar ese tiempo para ver un contenido en una página web.

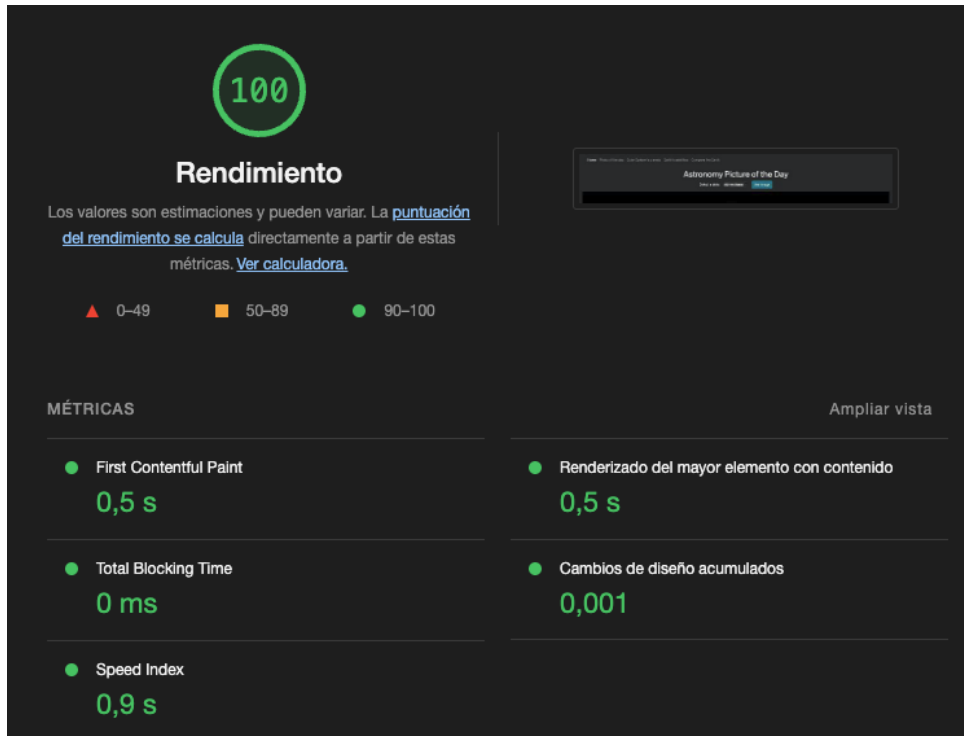
Por último se realizó una prueba con 4G y la página quedaba totalmente cargada en 7 segundos y desde el punto de vista del usuario no se percibe ninguna espera que pueda resultar molesta o tediosa.

Estos resultados hacen pensar que es una página realmente pesada y exige unas condiciones de internet favorables para poder disfrutar de forma fluida del contenido que se ofrece, es una realidad que a día de hoy 4G ocupa más del 60% de la población y no va a tener problemas para ver el contenido que se ofrece, a pesar de ello hay que intentar maximizar el número de personas a las que puede llegar este servicio.

Usando Lighthouse obtenemos una puntuación en móvil:



y en escritorio:



Página “Compare the Earth”:

- Resultados obtenidos:

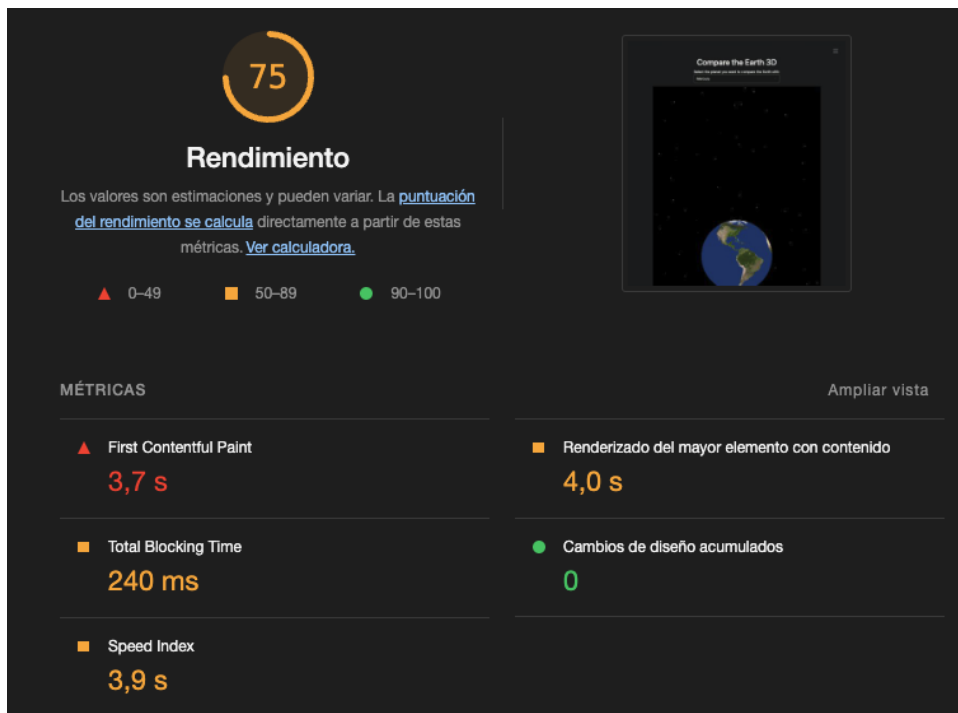
Estado	Método	Dominio	Archivo	Iniciador	Tipo	Transferido	Tamaño	0 ms
200	GET	localhost	compareEarthDimensions.php	BrowserTabCh...	html	3,70 KB	3,70 ...	341 ms
200	GET	localhost	navbar.css	stylesheet	css	43 B	43 B	328 ms
200	GET	cdn.jsdelivrn...	bootstrap.min.css	stylesheet	css	25,05 KB	152,1 ...	7336 ms
200	GET	code.jquery...	jquery-3.2.1.slim.min.js	script	js	23,30 KB	67,97 ...	988 ms
200	GET	cdn.jsdelivrn...	bootstrap.min.js	script	js	13,66 KB	47,80 ...	6973 ms
200	GET	localhost	compareEarthDimensions.css	stylesheet	css	419 B	419 B	324 ms
200	GET	cdn.jsdelivrn...	bootstrap.min.css	stylesheet	css	29,63 KB	190,3 ...	8325 ms
200	GET	cdnjs.cloudflare...	three.js	script	js	177,08 KB	1,14 ...	6341 ms
200	GET	cdnjs.cloudflare...	d3.min.js	script	js	74,47 KB	269,9 ...	11336 ms
200	GET	cdn.jsdelivrn...	OrbitControls.js	script	js	5,45 KB	24,22 ...	11320 ms
200	GET	localhost	compareEarthDimensions.js	script	js	4,07 KB	4,07 ...	322 ms
200	GET	localhost	favicon.ico	FaviconLoader...		30,17 KB	30,17 ...	4376 ms
200	GET	localhost	texture3.jpg	three.js:36135 ...	jpeg	452,23 KB	452,2 ...	15007 ms
200	GET	localhost	texture1.jpg	three.js:36135 ...	jpeg	852,10 KB	852,1 ...	43022 ms

De nuevo repetimos el mismo procedimiento que en el resto de páginas, el peso total de la página es de 1.65 MB, menos de la mitad que “solar systems planets”, los tiempos de carga que hemos registrado han sido:

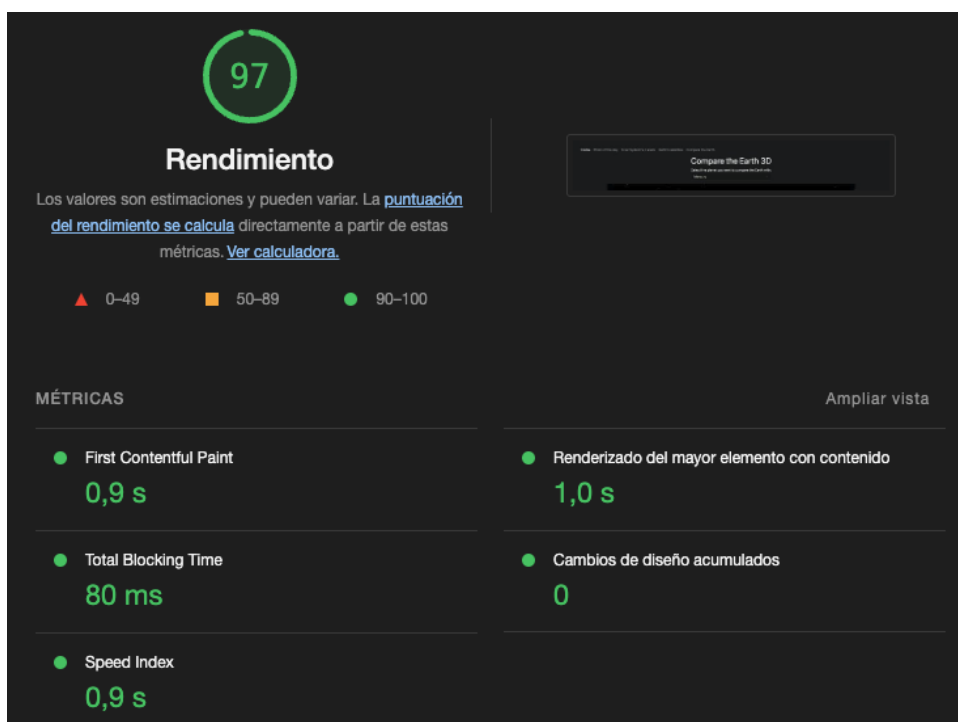
- Para 2G regular: 54,73 segundos
- Para 3G regular: 18.55 segundos
- Para 4G: 3.37 segundos

Una característica de esta página es que lo que más tarde en cargar son las texturas de los dos planetas a comprar y hasta que estos no carguen completamente el usuario no puede interactuar con la página y eso produce un mayor impacto en la percepción del usuario acerca del tiempo de espera.

Usando Lighthouse obtenemos una puntuación en móvil:



y en escritorio:



Modificaciones propuestas

- Cambiamos la imagen de fondo en la pestaña Home de formato .jpg a .avif haciendo la página “home” hasta 3 veces más rápida de lo que era antes.
- Descargamos las fotos de planetas y satélites en vez de acceder a ellas desde internet con el enlace, en primer lugar ajustamos la resolución de las imágenes a las que necesitábamos para no almacenar imágenes tan pesadas que luego redimensionábamos mediante css, de esta forma conseguimos unos resultados realmente impresionantes en las páginas de Satélites y planetas.

Pagina “Solar System’s planets” antes:

Inspector

Console

Depurador

Red

Editor de estilos

Rendimiento

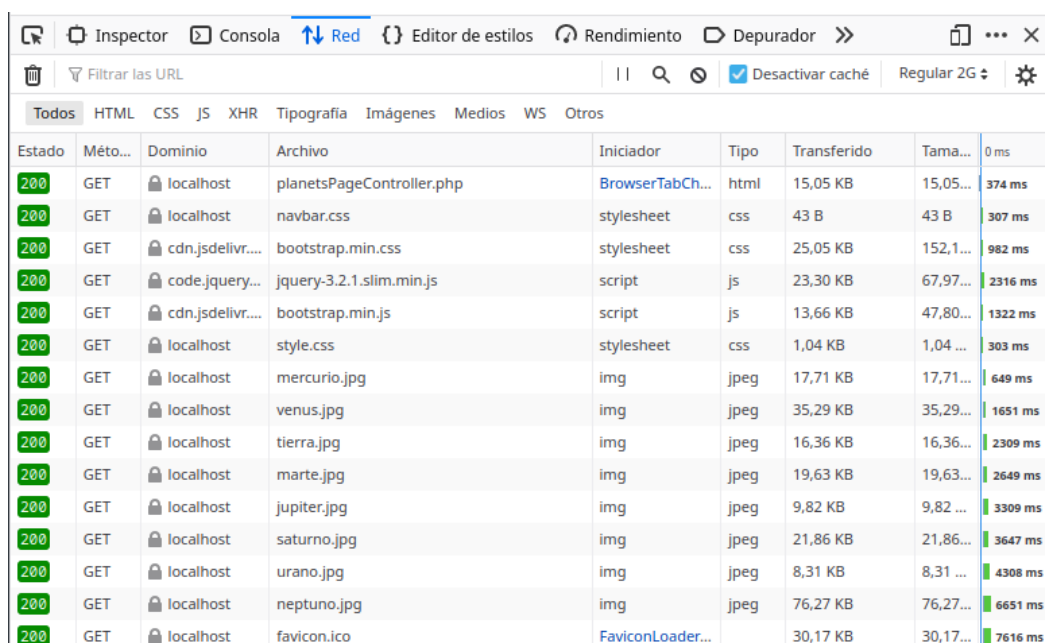
>>

...

X

Filtrar las URL

Página “Solar System’s planets” después:



Estado	Método...	Dominio	Archivo	Iniciador	Tipo	Transferido	Tamaño...	0 ms
200	GET	localhost	planetsPageController.php	BrowserTabCh...	html	15,05 KB	15,05...	374 ms
200	GET	localhost	navbar.css	stylesheet	css	43 B	43 B	307 ms
200	GET	cdn.jsdelivr.net	bootstrap.min.css	stylesheet	css	25,05 KB	152,1...	982 ms
200	GET	code.jquery.com	jquery-3.2.1.slim.min.js	script	js	23,30 KB	67,97...	2316 ms
200	GET	cdn.jsdelivr.net	bootstrap.min.js	script	js	13,66 KB	47,80...	1322 ms
200	GET	localhost	style.css	stylesheet	css	1,04 KB	1,04 ...	303 ms
200	GET	localhost	mercurio.jpg	img	jpeg	17,71 KB	17,71...	649 ms
200	GET	localhost	venus.jpg	img	jpeg	35,29 KB	35,29...	1651 ms
200	GET	localhost	tierra.jpg	img	jpeg	16,36 KB	16,36...	2309 ms
200	GET	localhost	marte.jpg	img	jpeg	19,63 KB	19,63...	2649 ms
200	GET	localhost	jupiter.jpg	img	jpeg	9,82 KB	9,82 ...	3309 ms
200	GET	localhost	saturno.jpg	img	jpeg	21,86 KB	21,86...	3647 ms
200	GET	localhost	urano.jpg	img	jpeg	8,31 KB	8,31 ...	4308 ms
200	GET	localhost	neptuno.jpg	img	jpeg	76,27 KB	76,27...	6651 ms
200	GET	localhost	favicon.ico	FaviconLoader...		30,17 KB	30,17...	7616 ms

La página pasó de pesar 3.7 MB a pesar 313 KB, lo cual es realmente asombroso y estos cambios fueron realmente significativos en términos de rendimiento ya que los tiempos de carga pasaron a ser:

Tiempos de carga	2G	3G	4G
Antes	1.12 min	19 s	7 s
Después	10.33 s	3.35 s	409 ms

Realmente supuso una gran mejora tanto en rendimiento como en experiencia de usuario, y la mejor parte es que no se sacrificaba en absoluto la calidad de las imágenes ya que en lugar de coger la imagen completa y aplicarle un resize ahora directamente guardamos la imagen del tamaño que nos interesaba sin ocupar espacio innecesario.

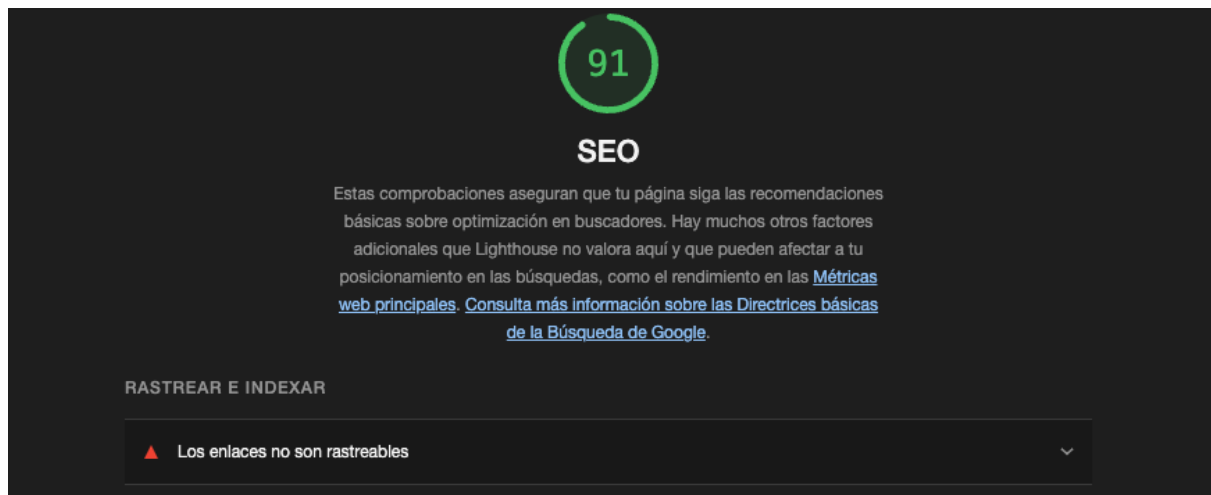
SEO

Auditorías realizadas

Hemos sometido a todas las páginas a Lighthouse

Resultados obtenidos

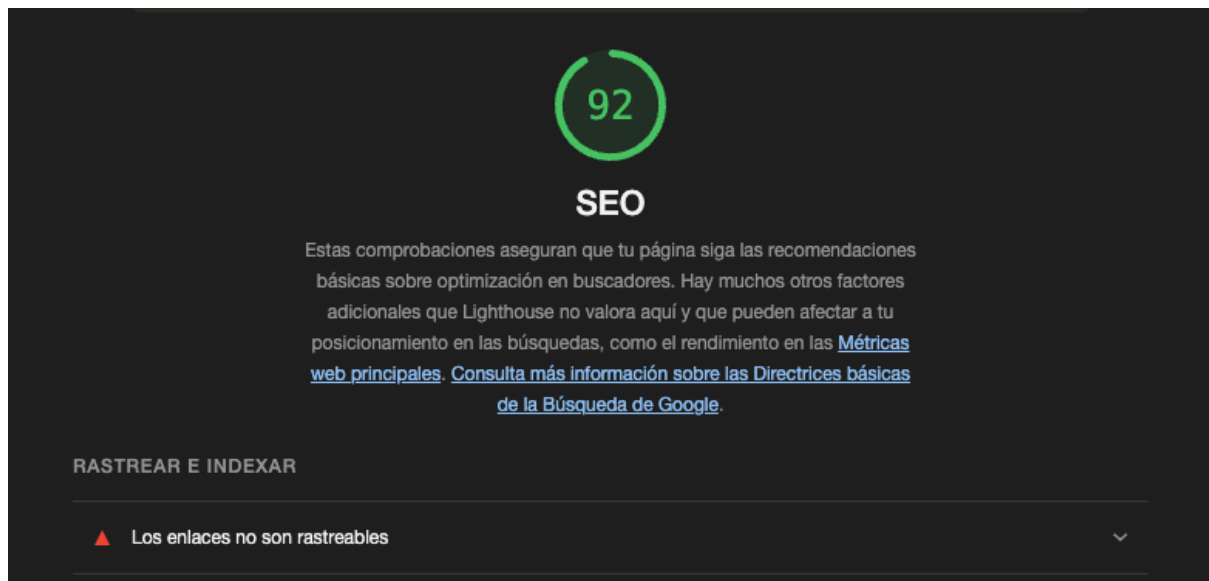
Página “Home”:



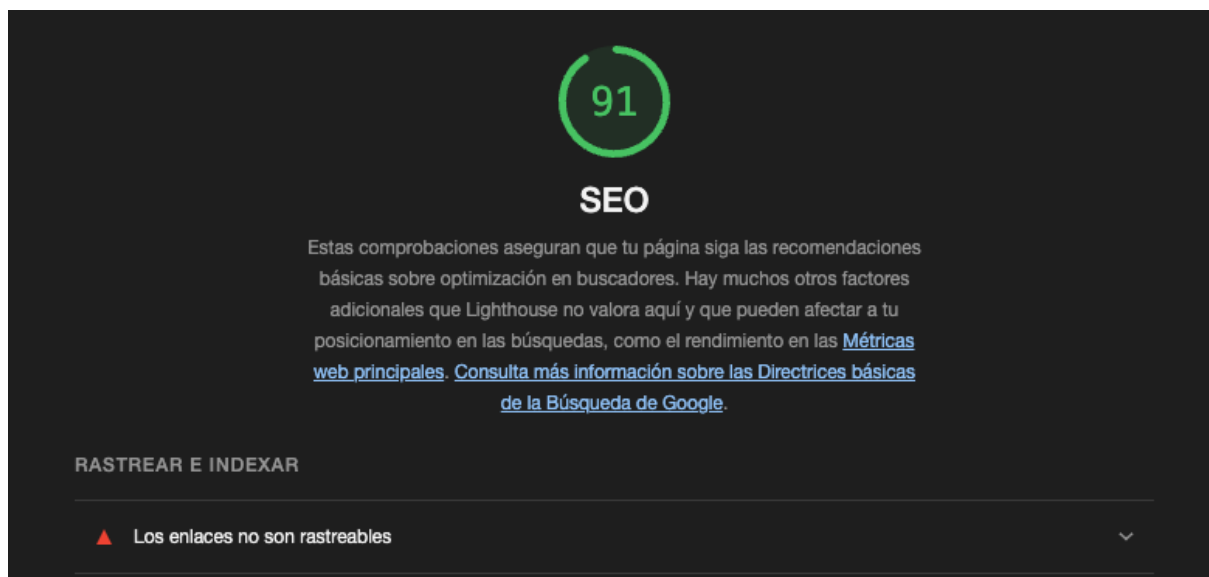
Página “APOD”:



Página “Solar System's planets”:



Página “Compare the Earth”:



Modificaciones propuestas

Hemos añadido metaetiquetas como la descripción o el viewport a todas las páginas para mejorar el SEO.

El único problema grave que tenemos en el momento es que las páginas no son rastreables ya que usamos plantillas de enlaces que rellenamos en el controlador php. Para solucionar esto podríamos crear un sitemap en xml por ejemplo que ayudase con la tarea de crawling.

Accesibilidad

La accesibilidad era un aspecto que tuvimos en cuenta desde un principio y ya de partida nos preocupamos por hacer un correcto uso de las herramientas con las que trabajamos para que la página web sea accesible en la mayor medida posible.

Por un lado, aprovechamos las etiquetas de html dotando a las diferentes pestañas de la página web de una estructura por la que es fácil desplazarse con el tabulador.

Por otro lado estaba la utilización de bootstrap, que es una librería que ofrece muchísimas herramientas y componentes para facilitar el diseño, su uso implica que muchos de los componentes html y sus atributos no están creados por nosotros si no que vienen predefinidos por bootstrap.

Esto implica que en algunas situaciones tenemos que delegar sobre bootstrap la gestión de la accesibilidad de sus componentes, pero por suerte en la mayoría de componentes que utilizamos, bootstrap se ha preocupado por que sean accesibles haciendo uso de atributos aria.

```
<a class="navbar-brand" href=""> </a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarCollapse"
  aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">
  <span class="navbar-toggler-icon"></span>
</button>
```

Este fragmento de código de nuestra página web es un pequeño ejemplo de cómo un componente de bootstrap hace uso de las atributos aria, de esta forma si una persona con discapacidad hace uso de algún tipo de aplicación como JAWS o NVDA entre otros, podrá obtener información de estos atributos.

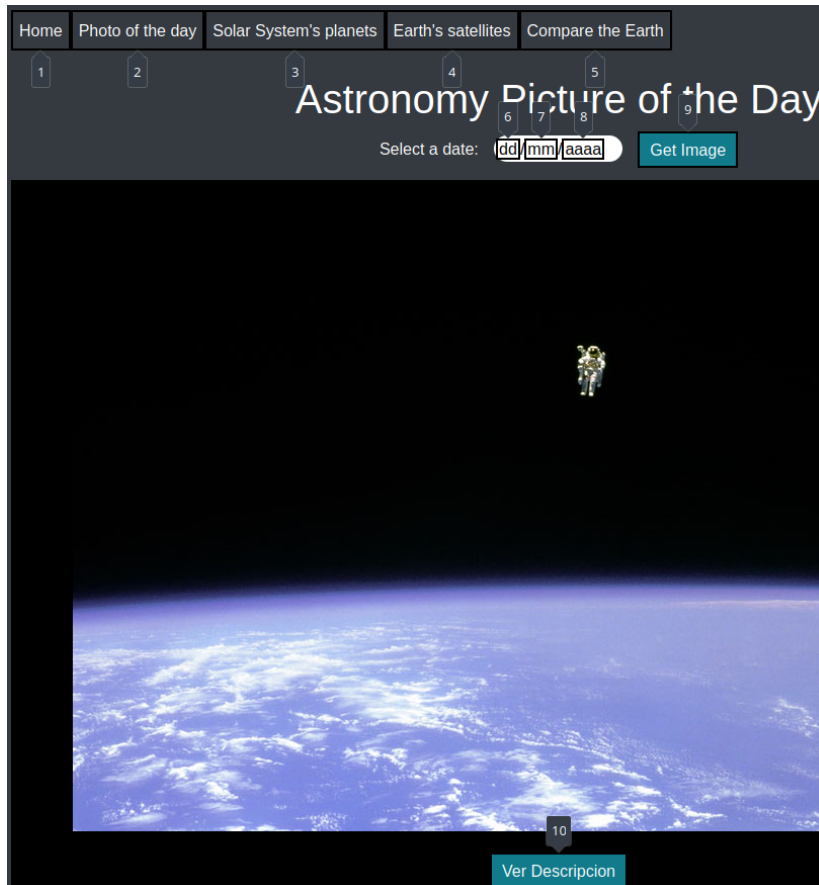
También hemos tenido en cuenta de cara a la accesibilidad el uso de contrastes adecuados para facilitar la lectura de textos y el uso de fuentes de fácil legibilidad y tamaño adecuado.

Parte de nuestra página web se caracteriza por la utilización de imágenes acompañadas de una descripción de la misma, desde el punto de vista de la accesibilidad a personas que no puedan ver las imágenes, estas imágenes pasan a ser meramente decorativas ya que no tendría sentido utilizar el atributo alt para describir algo que ya está descrito en la propia página web, es por ello que todas aquellas imágenes que ya cuenten con descripción tendrán el atributo alt="" indicando que son meramente estéticas.

Auditorías realizadas

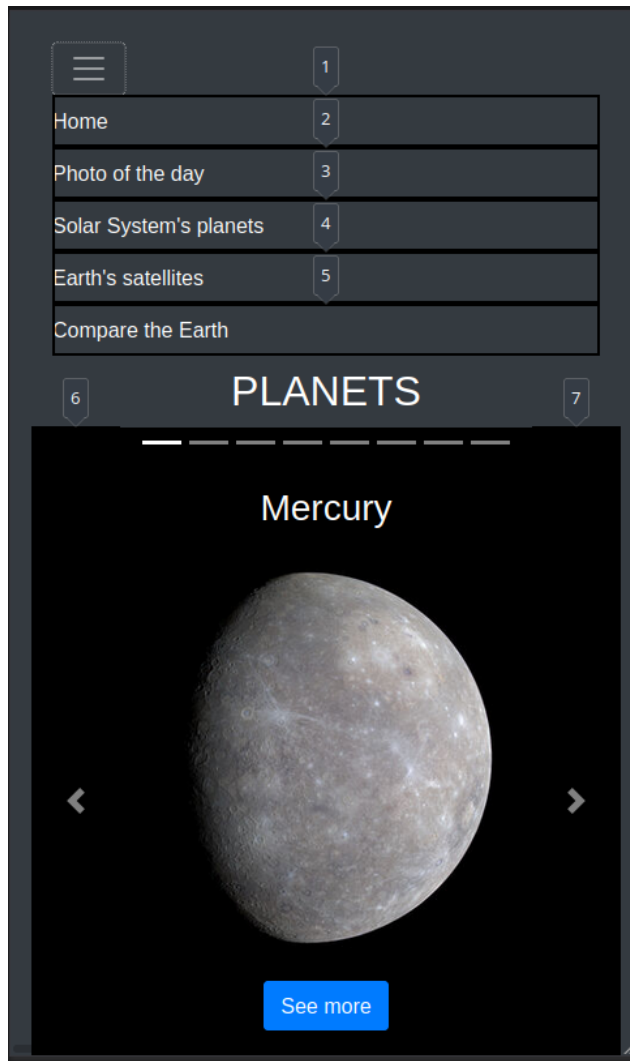
Pruebas similares a las otras categorías con Lighthouse.

También hemos utilizado las herramientas de desarrollador del navegador que ofrecen asesoría de accesibilidad y gracias a ello hemos podido estudiar el orden en el que el tabulador recorre los elementos de la página, la interacción con los inputs, recomendaciones del etiquetado y de la semántica ...



En esta imagen podemos apreciar el orden en el que el tabulador irá recorriendo los diferentes componentes de la página.

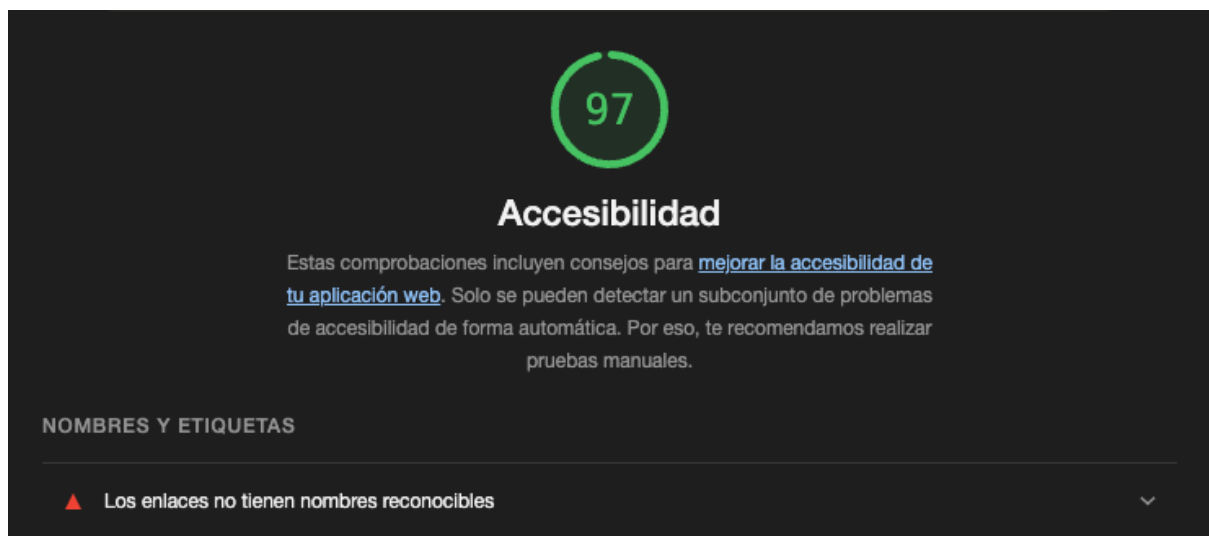
Gracias a ellos nos hemos asegurado de que no haya conflictos de bucles de los que no se puede salir con el tabulador ni ventanas emergentes que no se puedan cerrar o dificulten la navegación.



Las mismas pruebas las hemos realizado en la versión de móvil para asegurarnos que también es totalmente accesible y navegable por el tabulador.

Resultados obtenidos

Página “Home”:



Página “APOD”:

94

Accesibilidad

Estas comprobaciones incluyen consejos para [mejorar la accesibilidad de tu aplicación web](#). Solo se pueden detectar un subconjunto de problemas de accesibilidad de forma automática. Por eso, te recomendamos realizar pruebas manuales.

CONTRASTE

▲

 Los colores de fondo y de primer plano no tienen una relación de contraste adecuada.

▼

Página “Solar System's planets”:

98

Accesibilidad

Estas comprobaciones incluyen consejos para [mejorar la accesibilidad de tu aplicación web](#). Solo se pueden detectar un subconjunto de problemas de accesibilidad de forma automática. Por eso, te recomendamos realizar pruebas manuales.

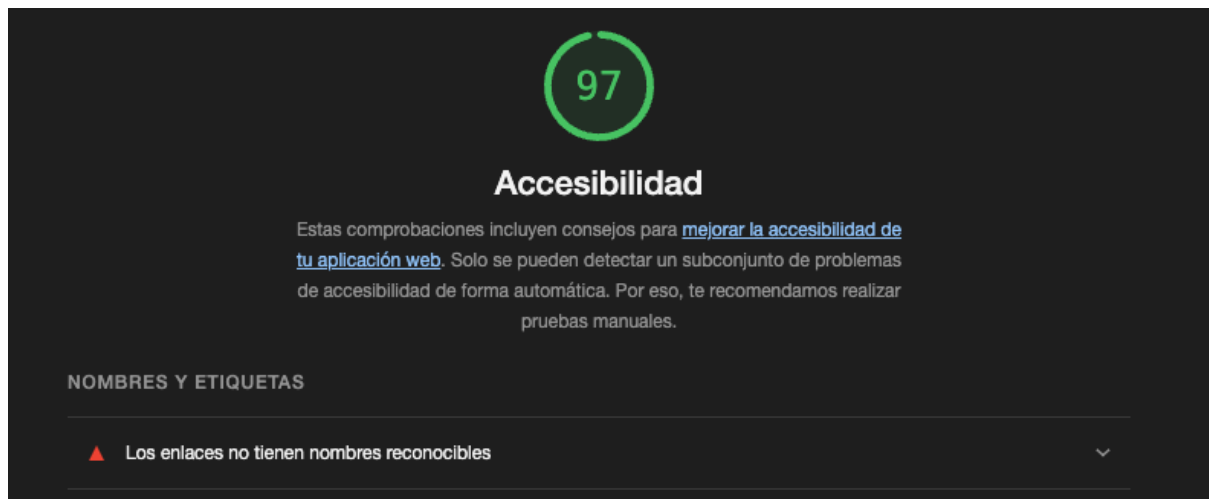
NOMBRES Y ETIQUETAS

▲

 Los enlaces no tienen nombres reconocibles

▼

Página “Compare the Earth”:



Modificaciones propuestas

Para mejorar podríamos cambiar los nombres de los enlaces para que sean más expresivos y aumentar el contraste de color entre fondo y letras en la pestaña de APOD.

Usabilidad

La página está hecha sabiendo que las personas normalmente centran su atención en el centro de la página primero y luego van escaneando el resto, es por eso que todo lo principal está en el centro y con un tamaño grande.

La interfaz intenta ser lo más intuitiva posible incluyendo poca información para no saturar al leyente y aportando mucho recurso visual que transmite más rápida información.

La barra de navegación se mantiene constante durante toda la experiencia de exploración por lo que refuerza el sentimiento de familiaridad en todo el viaje. Lo mismo pasa con el fondo y las letras que intentan formar una sensación de uniformidad.

También es notable que los colores utilizados son en su mayoría oscuros, así que es una interfaz respetuosa con la vista del usuario.

Todas las pestañas son accesibles desde todas las demás lo que permite fluir rápidamente por todo el entorno.

Gestión de configuración

Uso de repositorios

Como repositorio hemos usado el Gitlab que brinda los servidores de la UPNA, a ese repositorio tenemos acceso los tres integrantes del grupo y trabajamos directamente sobre la rama main.

Integración continua

Para nuestra estrategia de integración continua, hemos implementado un pipeline con GitLab, utilizando el archivo `.gitlab-ci.yml`. Este pipeline es esencialmente un script que GitLab ejecuta cada vez que se realiza un push al repositorio. Esto nos ayuda a automatizar varias tareas y asegurar la calidad del código.

Nuestro pipeline se compone de tres etapas: `beautify`, `minify` y `deploy`.

- En la primera etapa, `beautify`, se hace un embellecimiento del código para asegurar que se mantenga un estándar coherente y una fácil legibilidad.
- En la segunda etapa, `minify`, minimizamos nuestros archivos HTML, JS y CSS para optimizar la carga de la página web. Esto es especialmente importante para mantener un buen rendimiento de la página web.
- En la última etapa, `deploy`, copiamos todos nuestros archivos al servidor de producción. Esto facilita un despliegue rápido y sencillo de nuestras últimas actualizaciones y cambios en el código.

Nuestra pipeline se ejecuta en un runner en el escritorio virtual de la asignatura. Esto asegura que cada vez que hacemos un cambio en el código y lo pusheamos al repositorio, la pipeline se ejecuta automáticamente en este entorno controlado.

Trabajo en equipo y reparto de responsabilidades

Para organizarnos dentro del equipo hemos usado de las pizarras con formato kanban que ofrece el propio Gitlab con cuatro apartados:

- Open: donde ponemos las historias de usuario con formato 'Como...quiero...para...'
- TODO: aquí dejamos las tareas ligadas a su debida HU que están por hacer
- Doing: las tareas que se están haciendo actualmente
- Closed: las HU y tareas que ya se han terminado

Además cada vez que creamos una tarea nos asignamos uno de los integrantes y estimamos el tiempo que íbamos a tardar en completar dicha tarea, una vez terminada documentamos el tiempo real invertido para comparar y ver que tal estimamos, además de servirnos para saber si vamos bien de tiempo y llegamos a los plazos establecidos.

Hemos intentado paralelizar el trabajo todo lo posible para no estar los tres trabajando en lo mismo, ser lo más eficientes y productivos posible.

Conclusiones

Estamos muy contentos con el resultado final, creemos que hemos sabido plasmar muy bien los conocimientos adquiridos durante esta asignatura y nos ha quedado una pagina web accesible, usable, con buen SEO, que aporta información interesante y visualmente aparente.

Anexos

API de la NASA: <https://api.nasa.gov/>