

**Universidad
Rey Juan Carlos**

Escuela Técnica Superior
de Ingeniería Informática

Grado en ingeniería informática

Curso 2022-2023

Trabajo Fin de Grado

**APLICACIÓN DE RASTREO SATELITAL
MEDIANTE USO DE DATOS DE POSICIONAMIENTO
Y TIEMPO EXTRAÍDOS DE SISTEMAS GNSS**

**Autor: Esther Teso Cuesta
Tutor: Jesús Sánchez-Oro Calvo**

Agradecimientos

Primero de todo, quería expresar mi agradecimiento por la oportunidad de haber podido desarrollar este proyecto, a mi tutor Jesús por haber ejercido como mi guía durante todo esta experiencia de aprendizaje y mejora, a Glenn y a Adrián por ofrecerme la idea y orientarme a través de este primer contacto con la planificación GNSS, y a mis compañeros Alejandro e Inés por haberme brindado su ayuda, conocimientos y por supuesto su grata compañía. No podía haber caído en mejores manos.

Quiero agradecer a mi familia: mi madre, mi padre, mi hermano Soro y mi gata, por todo el apoyo que me han brindado afrontando este proyecto pero también a lo largo de todo el grado: por avalar mis capacidades y alentarme en mis afanes. Gracias también a todas mis amistades, en especial Ángela, Carlos y Mario por estar siempre a mi lado, por ofrecerme una mano asistente y un hombro reconfortante siempre que lo he necesitado. Y especial mención a Lucas por haberme aconsejado con tanta paciencia.

Y también quiero expresar mi agradecimiento hacia aquellos integrantes del profesorado y del alumnado que me han acompañado a lo largo de mi paso por la universidad, gracias por la inspiración, por la camaradería y por esta maravillosa vivencia en la que hemos disfrutado, trabajado y experimentado juntos para seguir avanzando.

Este proyecto supone el final de un ciclo pero sólo para abrir paso al inicio de otro nuevo, y no estaría completo sin mencionar que esto no sería lo que es hoy si no fuera por vuestra aportación a mi entropía, así que ¡Gracias a todos!

Resumen

En este Trabajo de Fin de Grado se expone el proceso seguido para el desarrollo de una aplicación de rastreo satelital que emplee datos de posicionamiento y tiempo centrándose en ofrecer resultados relativos a la visibilidad de elementos involucrados de sistemas GNSS.

El proyecto se ha ideado como una aplicación web programada en Python empleando la herramienta “Flask” para su soporte y las librerías “D3.js” y “Leaflet.js” para la gestión del apartado interactivo y el despliegue de un mapa para la visualización de los datos y resultados, respectivamente; adicionalmente también la aplicación está preparada para ser desplegada directamente en una máquina o en un contenedor “Docker”.

Se tienen como objetivos principales la elaboración de una aplicación web de rastreo satelital que permita la visualización de una red de estaciones terrestres y de las posiciones de las órbitas de los satélites a lo largo del avance del tiempo como su proyección en un mapa de la superficie terrestre, soportando la distinción de estas según la constelación GNSS a la que pertenecen y ofreciendo cálculos de visibilidad entre objetos posicionados en el cielo y en la superficie terrestre empleados para valorar el número de estaciones que un satélite tiene o tendría ubicado en una coordenada concreta.

Se obtuvo como producto una aplicación que cubre todos los objetivos principales y secundarios planteados permitiendo su utilización para una serie de casos de uso involucrados en la planificación GNSS. Para evaluar este producto resultante, se ha calificado el desempeño como página web bajo una serie de métricas que se han involucrado en el desarrollo de sus facciones finales.

Palabras clave:

- GNSS
- Rastreo satelital
- Coordenadas y geometría
- Red de estaciones
- Órbitas de satélites

- Python, D3.js y Leaflet
- HTML, CSS y Javascript

Índice de contenidos

Índice de tablas	XI
Índice de figuras	XIII
Índice de códigos	XV
1. Introducción	1
1.1. Contexto	1
1.2. Motivación	4
1.3. Estado del arte	6
1.3.1. Mercado actual	6
1.3.2. Software de rastreo satelital	6
1.4. Metodología empleada	8
2. Objetivos	9
2.1. Descripción de la propuesta	9
3. Descripción informática	11
3.1. Diseño	11
3.1.1. Librerías	12
3.1.2. Back-End	16
3.1.3. Front-End	23
3.2. Funcionalidad	31
3.2.1. Casos de uso	34
3.2.2. Despliegue	35
4. Validación	37
4.1. Evaluación del alcance	37
4.2. Métricas	38
4.2.1. Página que utiliza la plantilla HTML del Mapa	39
4.2.2. Páginas que utilizan la plantilla del Menú principal	43
5. Conclusiones y trabajos futuros	48
5.1. Cumplimiento de los objetivos	48

5.2. Campos de desarrollo	49
Bibliografía	51
Apéndices	54
A. Análisis de métricas	56
A.1. Análisis base	56
A.1.1. Página de visualización del mapa	56
A.1.2. Página del menú principal	84
A.1.3. Página de gestión de la red de estaciones	111
A.1.4. Página de gestión de los datos orbitales	137
A.2. Análisis posterior	163
A.2.1. Página de visualización del mapa	163
A.2.2. Página del menú principal	190
A.2.3. Página de gestión de la red de estaciones	215
A.2.4. Página de gestión de los datos orbitales	239

Índice de tablas

3.1. Ejemplo del formato de la red de estaciones	16
3.2. Ejemplo de formato de la cabecera de SP3	16
3.3. Ejemplo del formato de cada órbita en el cuerpo de SP3 (encabe- zada por su época)	17
3.4. Ejemplo del formato de la cabecera de nuestro fichero geojson . . .	17
3.5. Ejemplo del formato de la parametrización de una sección de la cuadrícula de nuestro mapa de calor	18
3.6. Ejemplo de las equivalencias entre identificador, constelación y color	28

Índice de figuras

1.1. Principal objetivo del proyecto	4
1.2. Segundo objetivo del proyecto	4
1.3. Tercer objetivo del proyecto	5
1.4. Cuarto objetivo del proyecto	5
3.1. Diagrama del diseño	11
3.2. Coordenadas	18
3.3. Coordenadas horizontales	19
3.4. Esquema de la interacción entre el diccionario de capas y las capas del mapa	25
3.5. Esquema de la utilización del botón de filtrado por visibilidad desde una posición y las diferente situaciones que se atraviesan	29
3.6. Menú principal	31
3.7. Estaciones	31
3.8. Órbitas	32
3.9. Mapa interactivo	32
3.10. Información de una estación	33
3.11. Filtrar órbitas visibles	33
3.12. Mapa de calor	34
3.13. Primer caso de uso	34
3.14. Segundo caso de uso	35

Índice de códigos

1

Introducción

1.1. Contexto

El sector espacial ha experimentado un desarrollo y cambio intensos en los últimos años de la mano del devenir de un mundo cada vez más interconectado que requiere de sus prestaciones en numerosos ámbitos mucho más allá del militar, y nuevas líneas de negocio relacionadas se han desarrollado en torno a este, entre las que se encuentra la que aquí se trata en profundidad, la navegación por satélite.

Es también por estas razones un sector de carácter variable que se ve profundamente afectado por sucesos políticos y económicos, condiciones meteorológicas, etc que se mantiene en crecimiento.

Para entender el sector de trabajo de esta aplicación es necesario exponer previamente algunos conceptos.

Un **sistema global de navegación por satélite** (Global Navigation Satellite System, GNSS) es un sistema de posicionamiento global y determinación de tiempo que implica una o más constelaciones artificiales de satélites, receptores de aeronaves y monitorización de la integridad del sistema; aumentando de ser necesario para dar soporte el rendimiento requerido por la operación prevista.

Este desempeño suyo que se podría resumir en la determinación de la posición y hora (cuatro dimensiones) con exactitud de un usuario que se le proporciona, es aplicado para todo tipo de fines como navegación, transporte, comunicación, vigilancia, geodesia, meteorología, etc.

La carrera de la navegación por satélite da sus primeros pasos a lo largo de

la década del 60 con el lanzamiento del primer satélite artificial, Sputnik I, y el establecimiento de la primera constelación de satélites artificiales, TRANSIT, constituida sólo por 6 satélites describiendo órbitas polares bajas. Los avances en materia de relojes atómicos más precisos para los satélites permitió finalmente por parte de la Armada y la Fuerza Área de los Estados Unidos el desarrollo del sistema NAVSTAR GPS, un sistema GNSS conformado inicialmente por 11 satélites y que continuó en crecimiento con los años. La antigua Unión Soviética también desarrolló un sistema similar con el nombre de “GLONASS”, ahora gestionado por la Federación Rusa, y en las últimas décadas la Unión Europea desarrolló su propio sistema llamado “Galileo” al igual que la República Popular China construyó el suyo, conocido como “BeiDou”.

Este uso militar de los GNSS que fue su origen, es en la actualidad una de sus principales aplicaciones, sin embargo también y cada vez más, por parte de los sistemas de posicionamiento por satélite del mundo se están ofreciendo servicios de uso civil, utilizados por muchas empresas para la generación de productos GNSS.

La estructura de un sistema espacial, y por ende de un GNSS, consta de los siguientes componentes operacionales:

- **Segmento espacial:** Comprende el satélite o constelaciones de satélites y sus enlaces ascendentes y descendentes con el resto del sistema.
- **Segmento terrestre:** Engloba todos sus elementos posicionados en tierra: estaciones terrestres, centros de control, redes terrestres, etc.
- **Segmento de usuario:** Se refiere al equipo receptor de la señal del servicio y usa la información transmitida para realizar cálculos de posición y tiempo.

Las principales sistemas GNSS que existen hoy en día son los siguientes:

- **GPS** (Global Positioning System) administrado por la Fuerza Espacial de los Estados Unidos
- **GLONASS** (traducido, Global Navigation Sputnik System) administrado por la Federación de Rusia.
- **Galileo** administrado por la Agencia de la Unión Europea para el Programa Espacial (EUSPA) y de creación, gestión y uso civil.
- **BeiDou** administrado por la República Popular China.
- **QZSS** (Quasi-Zenith Satellite System) administrado por Japón.
- **IRNSS** (Indian Regional Navigation Satellite System) administrado por la India.

Los principales tipos de datos que se manejan mayoritariamente sobre los sistemas GNSS y en la generación de productos GNSS a partir de la información de los servicios que ofrecen, son los de posicionamiento de los satélites a lo largo de las épocas del día, sus relojes utilizados y su emisión de señales y las frecuencias implicadas.

Otro factor importante en relación a esto es la gran cantidad de datos involucrada en el desempeño de estos sistemas, que debe ser seleccionada, procesada y corregida, generalmente bajo diferentes condiciones en función de distintos clientes, en procesamiento en tiempo real o por lotes; y que supone una dificultad añadida en el ejercicio de la generación de productos GNSS.

Una de las actividades más importantes tanto en la valoración de las prestaciones dependientes de estos sistemas y la calidad de los productos GNSS derivados, es la visibilidad que tiene el usuario de satélites en órbita del sistema y la visibilidad de estos satélites por una serie de estaciones de referencia, lo cual implica de información proporcionada por el segmento terrestre y por el segmento espacial.

Para esto es necesaria una selección de una red de confianza de estaciones GNSS de referencia a lo largo de todo el territorio implicado la cual debe suponer una fuente consistente, completa, fiable y asequible de información. Uno de los objetivos más directos que debe cumplir esta red es el de proporcionar buena visión de todo el cielo terrestre con el fin de evitar que no haya datos de referencia insuficientes para satélites en alguna época.

Debido a los múltiples factores involucrados en la selección: recursos para el procesado, el volumen, calidad, disponibilidad y confidencialidad de los datos, etc; y a la variabilidad del estado de las estaciones, esta tarea se ve comprometida a una revisión manual y periódica de la red de estaciones de referencia.

Un **Sistema de Información Geográfica** (Geographical Information System, GIS) es un conjunto de herramientas diseñadas para distintas actividades de tratado de información geográficamente referenciada.

Para que una aplicación sea considerada en esta categoría debe pretender cubrir las siguientes cuestiones: Localización de elementos, filtrado según condición, reconocimiento de tendencias, cálculo de rutas óptimas, identificación de pautas entre elementos distintos y generación de modelos.

En su sentido más amplio, esta definición puede estar relacionada con multitud de sectores de aplicaciones; para nuestro proyecto son de especial interés aquellas desarrolladas para el rastreo satelital. Las cuales estrictamente no pertenecen al campo de las GIS pero conforman un modelo de aplicación con bastantes similitudes. En la sección de estado del arte se ofrecen más detalles sobre las principales características de estas.

1.2. Motivación

En este entorno se expone la siguiente motivación de desarrollo del proyecto:

OBJ-01	Desarrollo de una aplicación web de rastreo satelital
	<hr/> El objetivo principal de este proyecto es el desarrollo de una herramienta para rastreo satelital enfocada en especial en facilitar el tratado con información de planificación GNSS.

Figura 1.1: Principal objetivo del proyecto

Con este planteamiento de proyecto, nos hemos interesado por las posibles ventajas de la utilización de elementos gráficos y visualización de datos. Se pretende que la aportación de un soporte visual pueda agilizar la revisión manual de grandes cantidades de información orbital, compatible con su integración en un sistema mayor de archivado y siguiendo los estándares de formato y referencias del sector GNSS para así evitar movimientos y transformaciones que impliquen dichos grandes datos.

OBJ-02	Visualización de las orbitas de los satélites
	<hr/> La información de posicionamiento de los satélites proporcionada por el usuario será mostrada a lo largo del tiempo a través de su proyección sobre la superficie terrestre, catalogada por su sistema GNSS en caso de pertenecer a alguno de los cuatros principales: Galileo, GPS, Glonass y Beidou.

Figura 1.2: Segundo objetivo del proyecto

Este planteamiento incluye la utilización de una red de estaciones propia seleccionada por el usuario, que junto a gráficos relativos a la visibilidad que esta red ofrece, sean integrados en el entorno visual para, junto a la visualización de los datos de posicionamiento, permitir juzgar su desempeño de una forma más directa.

OBJ-03	Señalización de la red de estaciones terrestres <hr/> <p>Se mostrará sobre la superficie terrestre un marcador en la posición de cada estación terrestre proporcionada por el usuario.</p>
--------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figura 1.3: Tercer objetivo del proyecto

Otras de las facilidades que ofrece es el estudio intuitivo de eventos de interés en relación a las órbitas: gaps o ausencia de datos en algún periodo de tiempo, anomalías en la órbita descrita por un satélite, carencia de visibilidad del posicionamiento de un satélite. Sucesos que si pasan inadvertidos en el abundante y fluido procesado de datos pueden hacer que se vean comprometidos el estado del sistema o a la calidad final del producto generado.

OBJ-04	Cálculo de visibilidad entre objetos posicionados en el cielo y en la superficie terrestres <hr/> <p>Se evaluará la visibilidad de la que dispone un punto en la superficie terrestre de un punto en el cielo terrestre mediante cálculos de coordenadas.</p>
--------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Figura 1.4: Cuarto objetivo del proyecto

Estos mecanismos, a parte de estar pensados para complementar la tarea de inspección de los datos, también ofrecen la posibilidad de ser dedicados a aumentar la transparencia de estas partes del proceso con el cliente, permitiendo una visión general de las distintas informaciones obtenidas sin necesidad de involucrarse en el manejo del sistema.

Con el propósito de satisfacer las necesidades antes descritas cumpliendo con estos cuatro objetivos identificados, se plantea el desarrollo de una aplicación web que permita el rastreo satelital a través de tres funcionalidades clave:

- Tratado de datos con el posicionamiento de un conjunto de satélites, distinguidos según el sistema GNSS al que pertenezcan; y de la distribución de una red de estaciones de referencia.
- Suministro de información derivada de cálculos de visibilidad de posiciones en el cielo terrestre desde posiciones en la superficie terrestre.
- Visualización interactiva de los datos de satélites y estaciones aportados junto a los resultados de los cálculos de visibilidad asociados.

Se idea además proporcionar a la herramienta de flexibilidad, licencia libre y facilidad de ampliación mediante el empleo de estándares de formato y coordenadas de posicionamiento, y el uso de material de licencia libre.

1.3. Estado del arte

Es por el carácter variado y mutable del entorno de los datos y la multitud de formatos utilizados, que se ha estimado fundamental realizar un estudio previo acerca de las herramientas análogas actualmente empleadas en el ámbito de GNSS con la intención de obtener referencias y localizar carencias para complementar el planteamiento del propio proyecto.

1.3.1. Mercado actual

En la actualidad, el mercado de las aplicaciones geomáticas esta dominado por un modelo diseñado para datos de la superficie terrestre expresados en latitud, longitud y altura, y no a posiciones espaciales que además suelen ser expresados en otros sistemas de referencia. Y aunque sí que existen herramientas que permiten sacar partido a datos “terrestres” y “celestes” por igual, en algunos casos haciendo uso de lo conocido como “dos dimensiones y media” o tres dimensiones, como es el caso de «Google Earth Enterprise»; no es frecuente que estas soporten posicionamientos en otro tipo de coordenadas que no sean geodésicas, como cartesianas o tropocéntricas.

Por este motivo también, pese a que se utilizan diversos formatos de ficheros estándar en el ámbito para describir información de posiciones y gráficos, no suelen encontrarse entre los soportados los empleados para información de satélites y estaciones sobre la integridad de los sistemas. Por otro lado, el enfoque de estas aplicaciones suele ser la manipulación y elaboración de nuevos datos geográficos a partir de fuentes previas, y no tanto la evaluación y contraste de estos. También existen aplicaciones del ámbito distribuidas bajo licencias de software libres o uso gratuito, como es el caso de «Marble», pero buscar que satisfagan condiciones más concretas como las expuestas anteriormente llevan a servicios más especializados pero bajo licencia de pago, como ocurre con «WinTrack Pro» para Windows o «Orbitrack» para Android.

1.3.2. Software de rastreo satelital

En lo relativo particularmente al software de rastreo satelital (satellite tracking software), este es un campo de mercado en expansión desde la última década y en aumento en los últimos años de la mano de los avances contemporáneos en

el sector GNSS. Son entonces de especial interés para este proyecto, aplicaciones dentro de este ámbito de trabajo enfocadas al trato con información orbital, donde como ejemplos podemos destacar «Gpredict», «PreviSat 5.0» u «Orbitrón».

Como puntos fuertes de esta esfera de programas se pueden nombrar:

- Su frecuente compatibilidad con formatos específicos del ámbito espacial. En muchos casos sólo suelen permitir la lectura de archivos de tipo “TLE” acompañado de predicción orbital para los posicionamientos a través del tiempo, pero existen alternativas que leen e interpretan toda clase de formatos para dicha acción.
- Aunque su objetivo principal es la visualización de órbitas, siendo conocedores de las demandas de las actividades del mercado destacan por permitir además la señalización de la posición de estaciones terrestre en la superficie del planeta.

Por otro lado, entre las más corrientes carencias de estas aplicaciones se puede mencionar

- Su principal planteamiento en la visualización de información, ya que no suele ser significativa la manipulación de datos orbitales, pero no siempre ofrecen funcionalidades de visibilidad que sí resultan más útiles; de las cuales la más común es la ilustración del horizonte de visibilidad sobre el mapa y no suelen ofrecer cálculos similares sobre la interacción entre los datos de órbitas y estaciones.
- Varias de estas aplicaciones pueden no ser del todo accesibles, al no ser soportadas por sistemas operativos más modernos o contar con interfaces desactualizadas (como ocurre con «Orbitrón») o al requerir de unos amplios conocimientos técnicos de la materia para comenzar a utilizarlas.
- Adicionalmente es bastante frecuente que estos sistemas no estén implementados para que el usuario pueda proporcionar la información de las estaciones terrestres, es la propia aplicación la que mantiene dicha red actualizada y a cambio permite al usuario seleccionar de esta sólo las estaciones de su interés. Es un planteamiento que supone la ventaja de poder ofrecer cálculos ya previamente realizados sobre las estaciones y por ende más rápido, como podría ser el caso del horizonte de visibilidad; pero también trae consigo la desventaja de no permitir su uso con estaciones que no se hayan añadido porque su posicionamiento no sea público, como puede suceder debido a confidencialidad.

1.4. Metodología empleada

Para la planificación y el trabajo en el proyecto se ha decidido emplear una metodología de desarrollo de software ágil, es decir un conjunto de métodos de ingeniería del software enfocados a desempeñar un desarrollo de software iterativo e incremental.

En este caso se planteó una dinámica inspirada en el marco de trabajo “Scrum” utilizado también en el entorno de trabajo, y esta fue ideada sobre tres pilares fundamentales:

- Ritmo de trabajo iterativo, dividido en fases definidas por los objetivos generales a cumplir en ellas.
- Desarrollo incremental con convivencia de las distintas implementaciones en la versión ejecutable de cada fase.
- Realización de una evaluación del avance del proceso, los objetivos cumplidos y pendientes, a lo largo y al final de la fase.

Estas fases a sus vez se dividieron en ciclos asociados a una cierta duración estimada y definidos por tareas concretas relativas a los objetivos de la fase a la que pertenecen; para permitir reaccionar con flexibilidad a imprevistos o cambios en las decisiones mediante la evaluación y planificación de estos ciclos sin alterar drásticamente el planteamiento de las fases.

De esta manera, la vida del proyecto se dividió en una serie de fases enfocadas principalmente al desarrollo e integración de las siguientes funcionalidades:

- Definición y planificación del proyecto.
- Aplicación web funcional.
- Lectura de datos.
- Visualización de los datos.
- Cálculos con los datos.
- Comprobación de la integración, mejora de la eficiencia y ampliación.

Cada una de estas fases se extendió durante varios ciclos según necesidad, contó con una versión “ejecutable” relativa a las implementaciones realizadas hasta el momento y previniendo retrocesos, y como parte de las evaluaciones en las fases sobre el avance del desarrollo adicionalmente se realizaron reuniones con el tutor de la universidad y del puesto de trabajo con cierta regularidad.

2

Objetivos

2.1. Descripción de la propuesta

Se propone como proyecto con la intención de lidiar con la motivación descrita anteriormente, el desarrollo de una **aplicación web** para **rastreo satelital** enfocada especialmente en facilitar el tratado con información de **posicionamiento** y **visibilidad** de los **satélites** de los sistemas GNSS y una **red de estaciones** terrestre, aportando agilidad a la herramienta mediante la incorporación de un tablero interactivo.

Esta propuesta se define con los siguientes objetivos principales y secundarios:

1. Utilización de estándares y formatos GNSS:

- Utilización de sistemas de referencia geodésicos globales que representan la Tierra como un elipsoide y de uso extendido en el ámbito:
 - **Geodesic Reference System 1980 (GRS80).**
 - **World Geodetic System 1984 (WGS 84) .**
- Formato de fichero SP3 para la expresión de los datos de posicionamiento orbital de los satélites.
- Soporte para satélites de constelaciones de los cuatro principales sistemas GNSS: Galileo, GPS, Glonass y Beidou.

2. Configuración y utilización de datos relevantes proporcionados por el usuario:

- Red de estaciones personalizada.
- Suministro de archivos SP3 con la información de las órbitas.
- Robustez frente a formatos incorrectos o datos corruptos.

3. Mapa geográfico de la superficie terrestre:

- Mostrado del posicionamiento de satélites según su órbita mediante su proyección sobre el plano terrestre.
- Señalización de las estaciones en el plano terrestre.
- Configuración de la visualización de las capas que conforman el mapa.
- Visibilidad terrestre del satélite, línea límite sobre la Tierra donde la elevación del satélite es inferior a un umbral configurable
- Mapa de calor de la cantidad de estaciones en vista que tendría un satélite (de la constelación Galileo) en dicha posición del cielo

4. Utilización de recursos de licencia libre:

- Librerías de Python y Javascript.
- Ilustración del plano terrestre.
- Hojas de estilo CSS.

5. Despliegue en contenedor de Linux

3

Descripción informática

3.1. Diseño

Se ha estructurado el diseño del proyecto en 2 módulos principales: “Back-end” y “Front-end”.

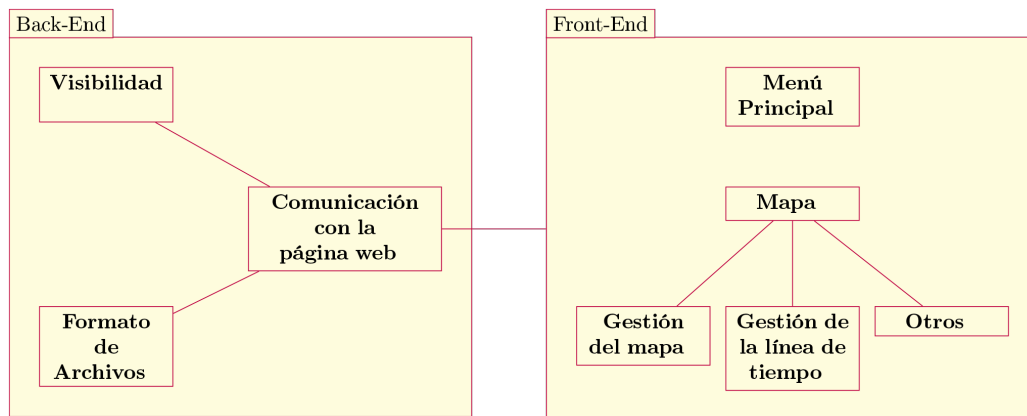


Figura 3.1: Diagrama del diseño

El módulo de “Back-end” está desarrollado en Python (versión 3.8) y engloba la API Rest, la sección dedicada a expresión de coordenadas y cálculos de visibilidad, y las clases encargadas de la lectura del posicionamiento de órbitas y estaciones.

Se he decidido utilizar Python ya que este lenguaje de programación cuenta con los siguientes puntos fuertes

- Se trata de un lenguaje de alto nivel y cuenta con una complejidad menor para su uso, al igual que buena legibilidad.
- Uso altamente extendido y gran cantidad de librerías para diferentes aplicaciones.
- Cuenta con gran versatilidad al ser considerado un lenguaje de paradigmas múltiples: programación estructurada, funcional y orientada a objetos.
- Es de código abierto bajo licencia Open Source.

Y la versión «3.8-slim-buster», porque es una de las más modernas, estable y cuenta con una imagen para su uso en el despliegue de la aplicación web empleando Docker.

El módulo descrito se comunica a través de la aplicación web con el otro módulo “Front-end”, este engloba las dos páginas web empleadas: el menú principal y el mapa terrestre; desarrolladas según el patrón MVC (“Modelo”, “Vista” y “Controlador”) en HTML, CSS y Javascript, y que la API Rest completa con la información adicional requerida ante una petición.

Y se han empleado estos lenguajes de programación HTML, CSS y Javascript porque se tratan del estándar para el desarrollo de páginas web.

3.1.1. Librerías

Se han utilizado librerías externas para la implementación tanto del front-end en lenguaje Javascript, como del back-end en lenguaje Python:

Python

Para la realización de los cálculos matemáticos involucrados en el funcionamiento de la aplicación web, se ha empleado la librería «**Numpy**» (versión 1.22.1).

Sus principales puntos fuertes son:

- Es una librería de código abierto bajo la licencia BDS.
- Cuenta con un rendimiento excepcional gracias a su núcleo de código en C optimizado.
- Ofrece una gran cantidad de funciones matemáticas, como integrales, transformadas de Fourier, etc.
- Se trata de una de las librerías más utilizadas para el tratamiento y procesamiento de datos en numerosos ámbitos.

Ha sido escogida por tratarse de una de las librerías de cálculos en Python más utilizada y por ser empleada, y por tanto compatible, por otra de las librerías seleccionadas para el proyecto, «Pandas», introducida a continuación. Esto la convertía una gran candidata para la realización de las operaciones para la transformación entre coordenadas y proyecciones.

Para el procesado de los datos de estaciones y órbitas se ha utilizado la librería «**Pandas**» (versión 1.4.0), enfocada a la manipulación de conjuntos de datos con indexación integrada en formato de tabla.

Sus principales puntos fuertes son:

- Es una librería de código abierto bajo la licencia BDS.
- Permite lectura y escritura de datos en diferentes formatos como CSV, Microsoft Excel o bases de datos SQL.
- Altamente optimizada para el rendimiento: filtrado, fusión y unión de datos.
- Preparada para el manejo inteligente de datos ausentes.
- Permite una intuitiva indexación de varios niveles.
- Utilizada en numerosos ámbitos académicos y comerciales.

Ha sido escogida para su uso en el proyecto debido a su amplia utilización en el ámbito y su compatibilidad con otras librerías, el manejo optimizado que ofrece de la multindexación de las órbitas por época y satélite para agilizar el acceso a los datos, pero también por la optimización en el filtrado de datos por criterios de visibilidad; factores clave en las principales funcionalidades del sistema.

Para el desarrollo de la aplicación web se ha utilizado el micro WSGI Framework «**Flask**» (versión 2.0.2).

Sus principales puntos fuertes son:

- Es una librería de código abierto bajo la licencia BDS.
- Compatible con Python3.
- Permite un buen manejo de las rutas de la aplicación por parte del controlador.
- Enfocado a un desarrollo sencillo que mantenga compatibilidad con la extensión de funcionalidades.
- Se trata de un micro Framework sin ORMs, por lo que permite el enfoque buscado en nuestro caso de integración en un sistema de archivado previo.

Fue decidida su utilización debido a la flexibilidad que ofrece al proyecto para ser integrado en el sistema en uso de la empresa, basado en una estructura de archivado y una red de funcionalidades implementada en Python. Facilitando además esta integración y posible extensión en el nuevo sistema gracias a su carácter de compatibilidad.

Extendiendo la funcionalidad de la anterior librería y para permitir la realización de la compresión de texto en el intercambio de información de la aplicación

web desarrollada, se ha utilizado la librería «**Flask-Compress**» (version 1.13).

Cuenta con un funcionamiento sencillo: antes una petición a la aplicación implementada con Flask, comprueba si cliente y servidor utilizan algún algoritmo de compresión común y en caso de hacerlo y de que los datos sean comprimibles, los comprime y agrega los encabezados pertinentes a las respuestas.

En colaboración con las anteriores librerías mencionadas y para la utilización de un servidor WSGI para soportar la aplicación web desarrollada, se ha utilizado la librería «**Waitress**» (versión 2.1.2).

Ha sido utilizada para el proyecto en virtud de que la propia librería de Flask cuenta con funcionalidades para ejecutar la aplicación con un servidor de desarrollo, sin embargo este no está pensado para el despliegue, por lo que es necesario hacer uso de otro servidor eficiente, estable y seguro, y es para esta tarea para la cual se ha seleccionado la librería Waitress.

JavaScript

Para el desarrollo del apartado gráfico dinámico de la aplicación web se ha utilizado la librería «**D3.js**» (Data-Driven Documents) (versión 7.3.0), basada en la manipulación eficiente de documentos vinculando datos a un DOM (Document Object Model).

Sus principales puntos fuertes son:

- Es una librería de código abierto bajo la licencia BDS.
- Basado en 4 sencillas dinámicas: selecciones, transiciones, asociaciones de datos y gestiones de nodos basadas en datos.
- El uso de un DOM permite un manejo total y complejo del resultado visual final, similar a ofrecido por lenguajes orientados a objetos.
- Enfocado a la rapidez de reacción con grandes conjuntos de datos.
- Vocabulario heredado de los estándares web HTML, SVG y CSS.
- Extendido uso en el desarrollo de mapas.

Se decidió su utilización debido a la flexibilidad y control que aporta al proyecto sobre los elementos del archivo HTML. Para poder cumplir con los objetivos del proyecto, sería necesario poder manipular ágilmente desde el script de Javascript los elementos involucrados en la visualización de las estaciones y del posicionamiento de los satélites agrupados por sus constelaciones, de manera que pudiesen ser mostrados conformen avanzaban las épocas y a la par que se mostraban otros datos sobre visibilidad. Es decir, el comportamiento de los elementos tenía que ser dinámico y para lograrlo se empleó esta librería.

También se investigaron otras herramientas similares como «Node.js» para este objetivo, pero finalmente se decidió hacer uso de D3.js debido a que esta

librería está particularmente enfocada a la visualización dinámica e interactiva de datos, y esta tarea tenía una fuerte componente visual.

Para la generación y manejo del mapa gráfico se ha utilizado la librería «**Leaflet**» (version 1.7.1), enfocada en el despliegue de mapas web en mosaico alojados en servidores públicos con superposiciones personalizables.

Sus principales puntos fuertes son:

- Es una librería de código abierto bajo la licencia BDS.
- Soportada por la mayoría de plataformas de móvil y ordenador.
- Basado un sistema de superposición de capas como agrupaciones de mosaicos, sencillas y eficientes de manejar.
- Sin uso de plugins soporta capas de GeoJSON, WMS (Web Map Service) y capas de Vector.
- No utiliza dependencias externas.

En un inicio, para la composición y manejo del mapa se valoraron otras opciones como la elaboración de todos los elementos gráficos directamente a partir de datos en formato «Geojson» o «Topojson» mediante otras herramientas para su interpretación como «Noje.js». Finalmente la librería “Leaflet” fue escogida para este propósito debido a

- La facilidad de generación de un mapa de proyecciones geográficas que ofrece: en contraste con la complejidad de generar una geométrica de cero con «Geojson» o «Topojson».
- Todo el abanico de funcionalidades y gráficos de posicionamiento sobre el plano terrestre ya implementados: Como los punteros, las etiquetas asociadas a punteros y la manipulación de capas de datos para visibilizarlas u ocultarlas con un panel de control.
- La optimización previa con la que ya cuenta para manejar grandes cantidades de este tipo de datos: Favoreciendo la necesaria manipulación continuada de los elementos mostrados en el mapa que por ejemplo se requiere para poder mostrar la variación de los datos orbitales a lo largo del día.
- Compatibilidad con el formato «GeoJson» para adicionalmente hacer uso de gráficos más complejos o personalizados sin limitaciones.

3.1.2. Back-End

Módulos de formato de archivos / Parsers

Tres tipos de formatos son utilizados por la aplicación y esta sección se encarga de su parseo y escritura dependiendo del tipo de archivo.

- Red de estaciones en formato CSV:** Una clase realiza la lectura de un fichero .csv con obligatoriamente la siguiente información en sus columnas: “#StationName”, “Latitude” (en grados), “Longitude” (en grados), “ReceiverName” y “ReceiverSatelliteSystem”, donde “#StationName” debe ser el identificador único de la estación.

Una muestra de este formato sería

Tabla 3.1: Ejemplo del formato de la red de estaciones

#StationName	Latitude	Longitude	ReceiverName	ReceiverSatelliteSystem
GIEN	46.631	8.507	SEPT POLARX5	GAL+GPS+GLO+BDS
GAMG	35.591	127.925	TRIMBLE NETR9	GAL+GPS+GLO+BDS+QZSS

Se interpreta como una estructura “dataframe” de la librería “Pandas” similar al formato de una tabla y después de comprobar que contiene las columnas obligatorias, eliminar las filas con valores ausentes y comprobar que no hay coordenadas geodésicas incorrectas, se almacena la información en la clase.

- Órbitas de los satélites en formato SP3:** Una clase procesa un fichero en formato .sp3.

Primero realizando la lectura de la cabecera del archivo, cuya información más importante será la fecha de las épocas que contiene, el número de épocas y la lista de satélites para los que contiene información el archivo.

Y después realizando la lectura del cuerpo del archivo, que contiene por cada época una cabecera siguiendo este estándar

Tabla 3.2: Ejemplo de formato de la cabecera de SP3

Symbols	Year Start	Month Start	Day of Month	Hour Start	Minute Start	Second Start
*	2001	8	8	0	0	0.0000000

y las sucesivas órbitas de los satélites en esa época, siguiendo este estándar

Tabla 3.3: Ejemplo del formato de cada órbita en el cuerpo de SP3 (encabezada por su época)

Symbol	Vehicle Id	x-coordinate (km)	y-coordinate (km)	z-coordinate (km)
P	G01	-11044.805800	-10475.672350	21929.418200
189.163300				
X-sdev (b**n mm)	Y-sdev (b**n mm)	Z-sdev (b**n mm)	C-sdev (b**n mm)	
18	18	18	2019	
Clock Event Flag	Clock Pred Flag	Maneuver Flag	Orbit Pred. Flag	
E	P	M	P	

Cabecera y cuerpo se interpretan como una estructura “dataframe” de la librería “Pandas” similar al formato de una tabla, que se almacena en la clase esta información se presenta doblemente indexada primero por el año, mes y día de todas las épocas y luego por las horas, minutos y segundos de cada época.

Adicionalmente, la clase que parsea estos archivos permite introducir para parsear un strings o clase Path (de Pathlib) con la dirección del archivo, o una lista de ellos de manera que se devuelve una tabla con todas las cabeceras y otra tabla con todas las órbitas.

- Gráfico heatmap del número de estaciones visibles en formato Geojson:** Se realiza la escritura de los cálculos de visibilidad de estaciones de la cuadrícula del mapa antes mencionada como un mapa de calor en un fichero siguiendo el formato geojson, un estándar definido para codificar variedad de estructuras de datos geográficos en formato json y que contiene la siguiente información: “type”, “name”, “features” y “properties”, donde adicionalmente para graficar adecuadamente nuestro heatmap se almacena el valor “opt_value” para representar el óptimo grado de visibilidad considerado.

Una muestra de cómo se utiliza en el caso de nuestro mapa de calor sería

Tabla 3.4: Ejemplo del formato de la cabecera de nuestro fichero geojson

type	name	properties	features
FeatureCollection	Visibility of stations heatmap	opt_value: 0.8	...

La sección “features” se trata de las parametrizaciones de todos los polígonos que modelan la figura además de ciertas propiedades adicionales, en el caso de los polígonos que conforman nuestro mapa de calor se aporta además su densidad de visibilidad del 0 al 1.

Un ejemplo del formato de estas parametrización sería

Tabla 3.5: Ejemplo del formato de la parametrización de una sección de la cuadrícula de nuestro mapa de calor

type	properties	geometry				
		type	coordinates			
			vértice 1	vértice 2	vértice 3	vértice 4
Feature	density: 0.65	Polygon	[-75.0, -67.5]	[-70.0, -67.5]	[-70.0, -65.0]	[-75.0, -65.0]

Módulo de visibilidad

Es el encargado de todos los cálculos relativos a sistemas de referencia espaciales (SPR), transformación entre coordenadas y comprobación de la condición de visibilidad desde un punto a otro en función de dichas coordenadas.

Las posiciones de las órbitas de los satélites a lo largo de las épocas del día vienen expresadas en coordenadas ECEF (Earth-centered, Earth-fixed), pero las posiciones para su visualización en el plano terrestre deben ser proyecciones geográficas a coordenadas geodésicas, en este proyecto interpretadas según el sistema de referencia GRS80 (Geodetic Reference System 1980); por lo que dos funciones que realiza este módulo son la transformación de la información de un modelo a otro y viceversa.

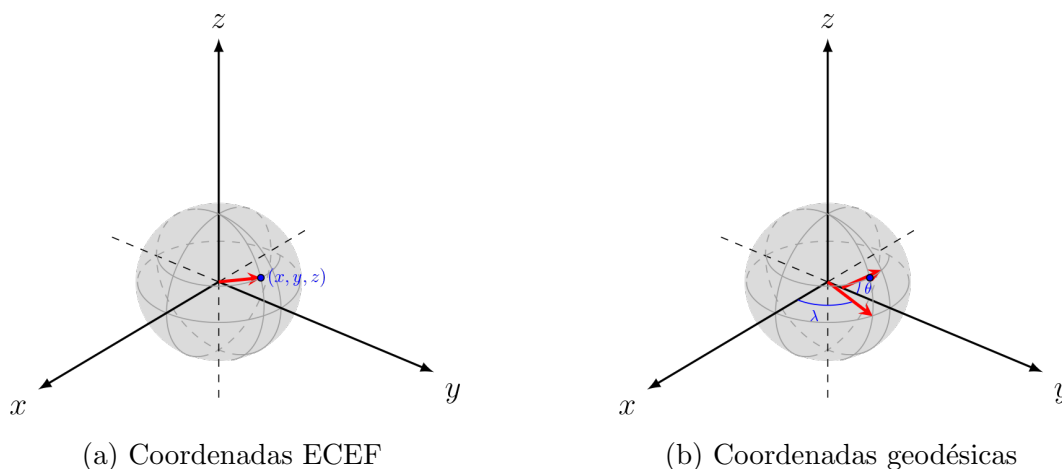


Figura 3.2: Coordenadas

La comprobación de la visibilidad de un punto del cielo desde un punto del plano terrestre se realiza con ambos puntos expresados en coordenadas ecef, mediante el cálculo de las coordenadas horizontales del punto estelar relativas al observador, en este caso el punto terrestre. Sobre estas coordenadas se define como condición de visibilidad que el valor de elevación sea superior a un umbral, por defecto 5° (estipulado como constante).

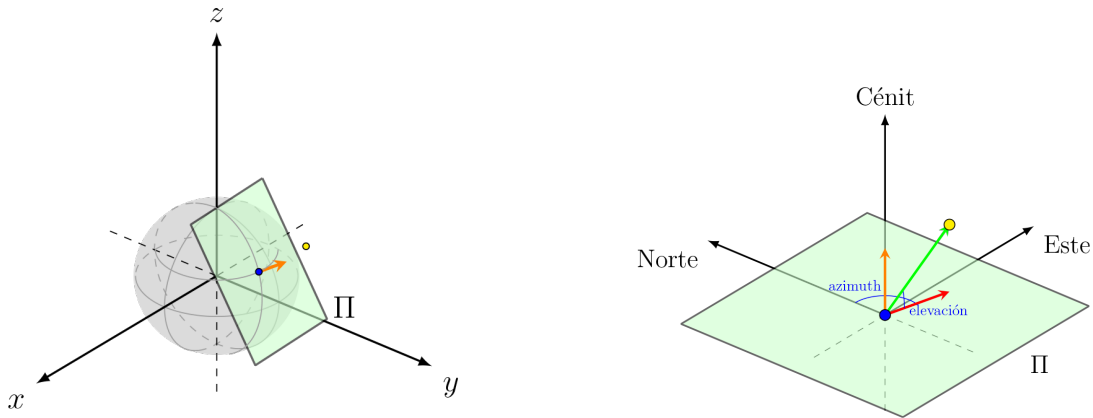


Figura 3.3: Coordenadas horizontales

Para la transformación entre coordenadas se han implementado los siguientes algoritmos:

Sean, según el modelo GRS80,

- a el eje semimayor del elipsoide terrestre.
- b el eje semimenor del elipsoide terrestre.
- $e = 1 - \frac{b}{a}$ la primera excentricidad del elipsoide terrestre.
- $f = \frac{a-b}{a}$ el achatamiento del elipsoide terrestre.

- **Coordenadas geodésicas a ECEF:** Sean (ϕ, λ) latitud y longitud respectivamente y h altura

$$X = (N + h) \cdot \cos(\phi) \cdot \cos(\lambda) \quad (3.1)$$

$$Y = (N + h) \cdot \cos(\phi) \cdot \sen(\lambda) \quad (3.2)$$

$$Z = \left(\frac{b^2}{a^2}N + h\right) \cdot \sen(\phi) = ((1 - e^2) \cdot N + h) \cdot \sen(\phi) \quad (3.3)$$

$$\text{con } N = \frac{a}{\sqrt{1 - e^2 \cdot \sen^2(\phi)}}.$$

- **Coordenadas ECEF a geodésicas:** Sean

$$\lambda = \arctan\left(\frac{Y}{X}\right) \quad (3.4)$$

Se definen

$$p = \sqrt{X^2 + Y^2} \quad (3.5)$$

$$h = \frac{p}{\cos(\phi) - N} \quad (3.6)$$

$$\text{con } N = \frac{a}{\sqrt{1-e^2 \cdot \text{sen}^2(\phi)}}$$

y el valor inicial

$$\phi = \arctan\left(\frac{Z}{p \cdot \left(1 - \frac{e^2 \cdot N}{N+h}\right)}\right) \quad (3.7)$$

- **Coordenadas horizontales a partir de dos coordenadas ECEF:** Sean p_1 y p_2 las posiciones del observador y del observado respectivamente.

- Cálculo del vector normal a la superficie terrestre en la posición del observador:

Sea $\frac{x^2}{b^2} + \frac{y^2}{b^2} + \frac{z^2}{a^2} - 1 = 0$ la ecuación implícita de la parametrización del elipsoide terrestre, utilizando el gradiente \implies

$$\vec{n} = \nabla f(x, y, z) = \left(2\frac{x}{b^2}, 2\frac{y}{b^2}, 2\frac{z}{a^2}\right) = \frac{2}{b^2} \left(x, y, \frac{z \cdot b^2}{a^2}\right) \quad (3.8)$$

por lo que, $\overrightarrow{\text{normal}} = \frac{\vec{n}}{|\vec{n}|} = (n_x, n_y, n_z)$ es lo mismo que normalizar el siguiente vector

$$\vec{m} = \left(x, y, z \frac{b^2}{a^2}\right) = (x, y, z \cdot (1-f)^2) \quad (3.9)$$

$$\text{con } (1-f)^2 = \left(1 - \frac{a-b}{a}\right)^2 = \left(\frac{b^2}{a^2}\right)$$

- Cálculo de las **coordenadas del plano tangente local** de la posición del observador, concretamente su variante **ENU**:

Según la matriz de transformación, estos son los vectores unitarios

$$\overrightarrow{\text{normal}} = (\cos(\lambda) \cdot \cos(\phi), \text{sen}(\lambda) \cdot \cos(\phi), \text{sen}(\phi)) \quad (3.10)$$

$$\overrightarrow{\text{sur norte}} = (-\cos(\lambda) \cdot \text{sen}(\phi), -\text{sen}(\lambda) \cdot \text{sen}(\phi), \cos(\phi)) \quad (3.11)$$

$$\overrightarrow{\text{oeste este}} = (-\text{sen}(\lambda), \cos(\lambda), 0) = (-y, x, 0) \quad (3.12)$$

con (ϕ, λ) latitud y longitud en coordenadas geodésicas.

Puesto que el vector normal unitario ya ha sido calculado, para ahorrar la transformación intermedia a coordenadas geodésicas se expresan vectores paralelos a sur norte y oeste este en función de las coordena-

das de $\overrightarrow{\text{normal}}$.

$$\begin{aligned} \overrightarrow{\text{sn}} &= \cos(\phi) \cdot \overrightarrow{\text{sur norte}} = \\ \cos(\phi) \cdot (-\cos(\lambda) \cdot \text{sen}(\phi), -\text{sen}(\lambda) \cdot \text{sen}(\phi), \cos(\phi)) & \quad (3.13) \\ &= (-n_x \cdot n_z, -n_y \cdot n_z, n_x^2 + n_y^2) \end{aligned}$$

$$\overrightarrow{\text{we}} = \cos(\phi) \cdot \overrightarrow{\text{este oeste}} = (-n_y, n_x, 0) \quad (3.14)$$

- Cálculo de las coordenadas horizontales:

Sea $\hat{p} = p_2 - p_1$ el vector desde el observador hasta el observado y $\overrightarrow{p_1 p_2} = \frac{\hat{p}}{|\hat{p}|}$ su normalización.

Aplicando la equivalencia entre coordenadas tropocéntricas y horizontales

$$\text{elevación} = \arcsen(\overrightarrow{p_1 p_2} \cdot \overrightarrow{n}) \quad (3.15)$$

$$\text{azimuth} = \arctg\left(\frac{\overrightarrow{\text{we}} \cdot \overrightarrow{p_1 p_2}}{\overrightarrow{\text{sn}} \cdot \overrightarrow{p_1 p_2}}\right) \quad (3.16)$$

Comunicación con la página web

Se trata de la sección del Back-End más en contacto con el Front-End y se encarga de la recepción y envío de información junto al correcto procesamiento de esta.

Por un lado se encuentran las funciones implementadas para la navegación por la aplicación, cuyo funcionamiento se basa en la utilización de una funcionalidad de Flask para completar con información adicional una plantilla HTML asociada a una hoja de estilo CSS y en el caso de la página del Mapa, asociada también a un script Javascript, y así proporcionar el acceso a la solicitada página web.

La idea con estas plantillas es generar desde el Front-End ciertas partes del HTML como variables de tipo string, para que después su contenido forme parte de la página web. Para esto, al final de la función asociada a la correspondiente URL, se utiliza la siguiente función

```
return render_template('Plantilla.html', nombre_de_la_variable=variable)
```

que devuelve la plantilla “Plantilla.html” que se encuentra en la carpeta “templates” con el contenido de “variable” insertado allí donde se encuentre escrito en la plantilla el siguiente llamamiento `{{url_for(nombre_de_la_variable)}}`.

Por defecto, esta función evita la inserción de texto HTML que pueda manipular la estructura de la plantilla; por este motivo, para aquellos textos con código

HTML que se ha necesitado introducir de forma literal, se ha comprobado en su proceso de generación que no supongan una vulnerabilidad y se han insertado literalmente de la siguiente manera `{{nombre_de_la_variable | safe}}`.

Y por el otro lado están las funciones encargadas de la gestión de peticiones get y post para el intercambio de información.

Esta sección está basada en dos páginas web construidas según el patrón MVC con HTML, CSS y Javascript. En total se tratan de dos plantillas HTML cuyo interior es completado por la API Rest implementada con “Flask” con la información relativa a cada sección de la aplicación; cuatro módulos Javascript encargados de gestionar la funcionalidad: uno principal que orquesta toda la funcionalidad de la página del mapa, uno relativo a acciones sobre el mapa, uno relativo a la línea de tiempo y sus interacciones y otro relativo a mensajes de error; y dos hojas de estilo CSS: una para el apartado gráfico general y otra para ajustes relativos sólo a la página del mapa.

Cumple con las siguientes funciones:

- **Comunicación con el front-end:** Conformada por
 - Las páginas principales de la aplicación: Gestión de la red de estaciones a utilizar, gestión de las órbitas de varios días y la visualización de dichos datos en el mapa.
 - Los métodos get y post que permiten respectivamente la solicitud y la subida de los datos de la red de estaciones, los ficheros de órbitas y las órbitas de los satélites en una fecha concreta.
- **Cálculos de visibilidad:**
 - Órbitas visibles desde un punto terrestre: Adicionalmente, en la petición de datos orbitales se permite proporcionar unas coordenadas geodésicas para solicitar sólo las órbitas del día visibles desde esa posición.
Para esta funcionalidad primero se realiza la lectura de los datos orbitales de la fecha, los cuales se reciben en formato “dataframe” y estos se filtran bajo la condición de que sean visibles desde la posición terrestre proporcionada en coordenadas geodésicas y transformada a coordenadas ECEF.
 - Escala de visibilidad del mapa de calor: Una función se encarga de dividir el mapa terrestre en una cuadrícula con secciones con unas dimensiones por defecto de 2.5^o de latitud por 5^o de longitud (estipuladas como constantes) y realizar los cálculos de visibilidad pertinentes para, en función de un número óptimo de estaciones visibles, calcular

el grado de visibilidad que le corresponde a cada sección.

Este grado de visibilidad se calcula tomando las coordenadas geodésicas del punto medio de esa sección de la cuadrícula con la altura a la que orbita la constelación Galileo (23222 kilómetros), transformándolas en coordenadas ECEF y finalmente calculando el número de estaciones de la red desde las que sería visible y dividiéndolo entre el número óptimo de estaciones esperadas, por defecto el 70% (estipulado como constante).

- **Otras funciones:** Lectura de datos según fecha, preparación de las órbitas para su conversión a json, etc.

3.1.3. Front-End

Esta sección está basada en dos páginas web construidas según el patrón MVC con HTML, CSS y Javascript. En total se tratan de dos plantillas HTML cuyo interior es completado por la API Rest implementada con Flask con la información relativa a cada sección de la aplicación; cuatro módulos Javascript encargados de gestionar la funcionalidad: uno principal que orquesta toda la funcionalidad de la página del mapa, uno relativo a acciones sobre el mapa, uno relativo a la línea de tiempo y sus interacciones y otro relativo a mensajes de error; y dos hojas de estilo CSS: una para el apartado gráfico general y otra para ajustes relativos sólo a la página del mapa.

Menú principal

Se hace uso de una misma plantilla HTML asociada a una hoja de estilo CSS y un script Javascript, para mostrar la información relativa a las secciones de menú principal, gestión de la red de estaciones y gestión de los datos orbitales, completando su interior con la información correspondiente a cada sección.

La plantilla HTML se trata de un archivo HTML normal en cuyo interior se encuentran especificados algunos elementos sin concretar mediante el siguiente formato `{{elemento_id}}`, este tipo de expresiones son identificadas por la aplicación “Flask” y sustituidas por información concreta antes de enviar este archivo tras realizarse una petición. De esta manera, esta plantilla está elaborada para que el interior del body, el título de la página y la descripción sean completados dependiendo de la página web a la que se acceda, y así de un modo sencillo conserven las tres la misma estructura.

En la página de gestión de la red de estaciones es donde se presentan las funcionalidades para la subida del fichero con la red de estaciones (sustituyendo el anterior fichero de haber sido suministrado antes) y donde se muestra la red

actual en caso de ya haber sido subida. Se proporciona la opción de subir un único fichero en formato csv, se notifica el éxito de la subida o si el proceso no ha podido completarse ya sea por presentar extensión o formato incorrectos o por algún error sucedido; y después se ofrece la opción de adicionalmente generar el mapa de calor de visibilidad a lo largo del plano terrestre en función a la red de estaciones proporcionada.

De forma análoga, para la gestión de los datos orbitales se permite la subida y almacenamiento de varios archivos de formato sp3, se comprueba la correcta extensión y nombre de estos, se muestra una lista de todos los datos hasta ahora almacenados y se ofrece la posibilidad de eliminarlos.

Mapa

Para la visualización de datos en el mapa se hace uso de una plantilla HTML asociada a una hoja de estilo CSS y un script Javascript, y cuyo interior se completa a través de la interfaz de “Flask” con la información de interés para que esta sea posteriormente gestionada por el script Javascript.

Esta página web contiene un mapa implementado a través de la librería “Leaflet” que cuenta con una capa base de una ilustración de la superficie terrestre del proyecto “OpenStreetMap” bajo licencia “Open Data Commons Open Database License” (ODbL), y con 7 capas desplegadas sobre esta:

- 5 capas para el posicionamiento de los satélites pertenecientes a cada una de los 4 principales sistemas GNSS y una adicional para los pertenecientes a cualquier otro sistema.
- La capa con el posicionamiento e información proporcionados sobre la red de estaciones.
- El mapa de calor que representa la visibilidad que un satélite en órbita presentaría desde la red de estaciones proporcionada.

Estas capas pertenecen permanentemente al mapa y mediante el panel de control pueden mostrarse u ocultarse, son los marcadores de posicionamiento de satélites en sus respectivas capas los que se van incorporando o extrayendo según sea necesario para efectuar las siguientes funcionalidades que ofrece la página:

- **Recepción de los datos desde la plantilla HTML:** Se realiza desde el script de Javascript la lectura de los datos insertados en la plantilla HTML como una variable en formato JSON mediante la siguiente etiqueta

```
<script> var data = JSON.parse('{{ data | tojson }}');</script>
```


En concreto se tratan de la fecha para la que hay datos orbitales más recientes, necesaria para posteriormente realizar una petición “get” de esos datos. Se inicializa el mapa utilizando la librería “Leaflet”, se generan y se añaden a él las capas principales: mapa de calor, red de estaciones, las cuatro constelaciones principales Galileo, GPS, Glonass y Beidou, y una capa que agrupará el resto de constelaciones; aunque estas capas aún permanecen vacías. Además, se añade un controlador para permitir mostrar y ocultar cómodamente estas capas que se han añadido. También se inicializa un diccionario de capas donde irán almacenados los marcadores de posición de los satélites divididos por constelación y por época, para facilitar la manipulación de esta información a la hora de introducirla con la interacción de la línea de tiempo.

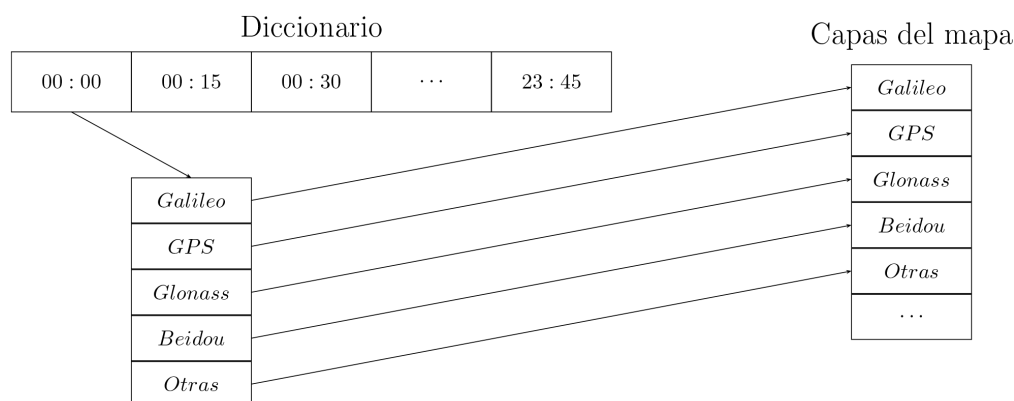


Figura 3.4: Esquema de la interacción entre el diccionario de capas y las capas del mapa

La mecánica del diccionario de capas y las capas incorporadas al mapa es la siguiente: Hay dos variables involucradas; por un lado, está el diccionario indexado de capas que tiene cinco capas para cada época en la que están divididas las 24 horas a visualizar: una para cada constelación principal de satélites y otra para agrupar otras constelaciones; y por otro lado están las capas incorporadas al mapa y gestionadas por el controlador, de la misma forma que con el diccionario: una por cada constelación principal de satélites y otra para agrupar las demás constelaciones. Las capas que están incorporadas al mapa siempre forman parte de él, ya puedan o no ocultarse sus elementos, estos siempre estarán formando parte del mapa; sin embargo, las capas del diccionario como no están asociadas al mapa se utilizan a modo de contenedores de marcadores, es decir, los elementos gráficos que representan las posiciones de los satélites. Esto lo que permite es

- Que los elementos gráficos sean generados desde el inicio y conforme cambie la época visualizada mediante la barra de tiempo sólo haya que

ir añadiendo esa capa “contenedor” y no iterar uno por uno; agilizando este proceso.

- Que las capas del mapa no estén variando todo el rato, lo cuál se vería reflejado en la leyenda del controlador que permite ver la visibilidad de las capas.

Esta es principalmente utilizada por la funcionalidad dinámica de mostrar el desplazamiento de los satélites a través de la línea de tiempo (supone una modificación rápida de lo mostrado en el mapa), se aportan más detalles al respecto posteriormente en la descripción de esta funcionalidad.

- **Obtención de la red de estaciones y el mapa de calor:** Se realiza una petición de tipo get para la obtención de cada uno de estos tipos de datos: la información y coordenadas de las estaciones configuradas y, de estar disponible, la geometría del mapa de calor. Si las peticiones salen bien, posteriormente se añaden marcadores y gráficos con esta información a su capa correspondiente (la de la red de estaciones o la del mapa de calor) a las capas que ya fueron incorporadas al mapa pero que permanecían vacías, es decir, siguen siendo las mismas capas que pueden ser manipuladas por el controlador, pero es su interior lo que se modificado.
- **Obtención de los datos orbitales de los satélites para la fecha más reciente:** A través de completar la plantilla HTML empleada, se recibe la fecha más reciente para la que se han proporcionado datos orbitales. Utilizando esta fecha se realiza una petición “get” para obtener los datos orbitales, se muestra en la página a que fecha y hora pertenecen los datos (junto a la barra de tiempo) y posteriormente estos se añaden al mapa de la misma manera descrita en la siguiente funcionalidad.
- **Selección de fecha:** Para permitir la selección de la fecha de los datos a visualizar la página presenta un selector de fecha y un botón asociado a un evento para la solicitud de los datos, este se trata de una función asíncrona que espera a la resolución de la petición get que se realiza en su interior para recibir los datos orbitales. Una vez se inicia este evento, se sacan del mapa las capas de constelaciones de la época que actualmente está siendo mostrada, se resetean las capas de todas las épocas que están siendo almacenadas y se resetea la barra de tiempo. Posteriormente se llama a la mencionada función que presenta la petición, esta realiza un “fetch” a la dirección del “get” aportando como argumento la fecha, y en caso de éxito recoge los resultados recibidos o en caso de error notifica al usuario el suceso. Finalmente se recogen los dos valores proporcionados: las coordenadas de las órbitas y la longitud en segundos de cada época, y se almacenan y grafican los datos como se explica más adelante. En caso de no haberse subido con anterioridad un fichero para dicha fecha o si sucede un error al gestionar la información, los datos de la respuesta

recogida no tienen contenido alguno a gestionar y esta situación se le notifica al usuario a través de una ventana emergente.

- **Capas de gráficos: constelación, red de estaciones y mapa de calor.** Se han creado una serie de funciones y variables de entorno para ampliar y gestionar la funcionalidad del mapa de la librería “Leaflet” y estas han sido agrupadas en el módulo relativo al mapa. Son las encargadas en su mayoría de lo explicado en esta sección.

En caso de estar dicha información disponible tras la solicitud de datos, desde el script principal de la página del mapa se genera el gráfico del **mapa de calor**:

Como las intensidades de las secciones se encuentran en el intervalo $[0, 1]$, este módulo cuenta con una función de escala lineal generada con “D3.js”, que para cualquier valor en ese intervalo devuelve el color equivalente de un degradado, en nuestro caso una progresión de rojo a amarillo a verde. Esta escala es utilizada por la función de generación del mapa de calor como estilo para graficar el formato geojson interpretando la intensidad de cada sección de la cuadrícula como una correspondiente tonalidad de color.

Uno de los problemas que surgieron con la elaboración de este mapa de calor era la gran cantidad de información en forma de polígonos que tenía que ser gestionada a la vez, lo cual ralentizaba el funcionamiento general de la página cuando se visualizaba esta capa con el mapa de calor; es por esto que este conjunto de gráficos una vez generado es renderizado como un único gráfico total para agilizar su posterior gestión por parte del mapa.

Se genera además una leyenda del mapa de calor que se añade como controlador al mapa, esta consiste en un “svg” que contiene: un rectángulo alargado con el gradiente de colores antes nombrado y la escala de valores equidistantes y en $[0, 1]$ que se convierten al dominio del número de estaciones visibles multiplicando por el valor óptimo propiedad del geojson (el valor óptimo es el número de estaciones visibles que representa la intensidad 1 en el geojson).

Posteriormente, iterando a través de la **red de estaciones** se hace uso de la correspondiente función del módulo de mapa para añadir a la capa de estaciones y por cada estación: un marcador en sus coordenadas junto a una etiqueta con su información que se muestra al colocar el cursor encima.

Cuando se solicitan las **posiciones orbitales** de los satélites para una fecha y se reciben de forma exitosa, una función del script principal se encarga de iterar en los datos orbitales recibidos: primero a través de las épocas esperadas (calculadas dividiendo el día en secciones con la longitud de la época que se recibe también) y segundo a través de cada satélite para el que se tiene datos en esa época concreta. En cada iteración se llama a una función del módulo del mapa que se encarga de añadir el satélite a la capa de su constelación para esa época; para ello añade a la capa correspondiente

un marcador de un color escogido para cada constelación y una etiqueta con el identificador único del satélite.

Tabla 3.6: Ejemplo de las equivalencias entre identificador, constelación y color

Ejemplo de identificador	Constelación a la que pertenece	Abreviatura	Color escogido
E04	Galileo	GAL	Amarillo
G16	GPS	GPS	Azul
R04	Glonass	GLO	Rojo
C11	Beidou	BDS	Verde

- Línea de tiempo** Un módulo se encarga de mediar con toda la funcionalidad de la barra de tiempo a través de una clase que encapsula el proceso. Esta clase trabaja con elementos de “D3.js” y está formada por una etiqueta `<input>` de tipo “range” `<input id=‘‘fader’’ type=‘‘range’’ min=‘‘0’’ value=‘‘0’’ aria-label=‘‘Time selector’’>` (que a partir de ahora será llamado barra), la variable que encapsula la etiqueta `<div>` en cuyo interior se escribe la época representada por el valor de la barra, el periodo (distancia en segundos) entre dos épocas y el tiempo actual en el que se encuentra la clase (dictado por el valor de la barra).

El tiempo es manejado a su vez a través de una clase que transforma los segundos totales en sus correspondientes horas, minutos y segundos y que puede devolver su equivalente como “string”.

Las funciones de esta clase tienen el objetivo de realizar ciertas modificaciones en la línea de tiempo manteniendo la concordancia entre el texto del “div”, el considerado tiempo actual de la barra y el valor real que tiene la barra. Y entre ellas destacan: modificar el periodo del sistema (cuando se realiza la lectura de datos con otra longitud de época esto supone también modificar el número y el valor de las selecciones que pueden hacerse en la barra), resetear el sistema (colocar la barra con valor 0) e incrementar la barra en una época.

Conforme se va desplazando la época actual en la línea de tiempo, se emplea esta mecánica explicada con anterioridad sobre el diccionario indexado de capas y las capas de constelaciones incorporadas en el mapa, de modo que el evento que va avanzando la época a lo largo del día lo que hace es establecer un intervalo (una función que se repite cada cierto tiempo) mediante un `setInterval(...)` en cuyo interior:

- Se retiran todos los elementos gráficos de los satélites correspondientes a la época actual de una sola vez retirando de las capas del mapa, las capas correspondientes del diccionario indexado.
- Desplazar la época representada en la barra del tiempo.

- Se añaden todos los elementos gráficos de los satélites relativos a la nueva época de una vez añadiendo directamente las capas del diccionario indexado a las correspondientes capas del mapa.

Desde el script principal, además se implementan los eventos activados al pulsar los botones “Play” y “Stop”. El primero consiste en utilizar “set_interval” para estipular que cada 120 segundos: se eliminen de las respectivas capas de constelaciones del mapa los satélites de la época actualmente mostrada, se incremente en una unidad la época de la barra, se actualice el texto de “div” con la nueva época, y finalmente se añadan a sus respectivas capas del mapa los marcadores de satélites de la nueva época, todo ello a través de las funciones de la clase de la línea de tiempo. Y el segundo evento implementado consiste en detener ese método que estaba en repetición.

- **Visibilidad** Adicionalmente, a la funcionalidad de petición de datos se le añade la posibilidad de seleccionar interactivamente un punto del plano terrestre para filtrar los datos orbitales según sean o no visibles desde esa posición. Esta funcionalidad se maneja mediante un botón de selección de la posición cuyo texto y evento desencadenado al hacer click sobre él, van cambiando en función de la fase de selección en la que se encuentre:

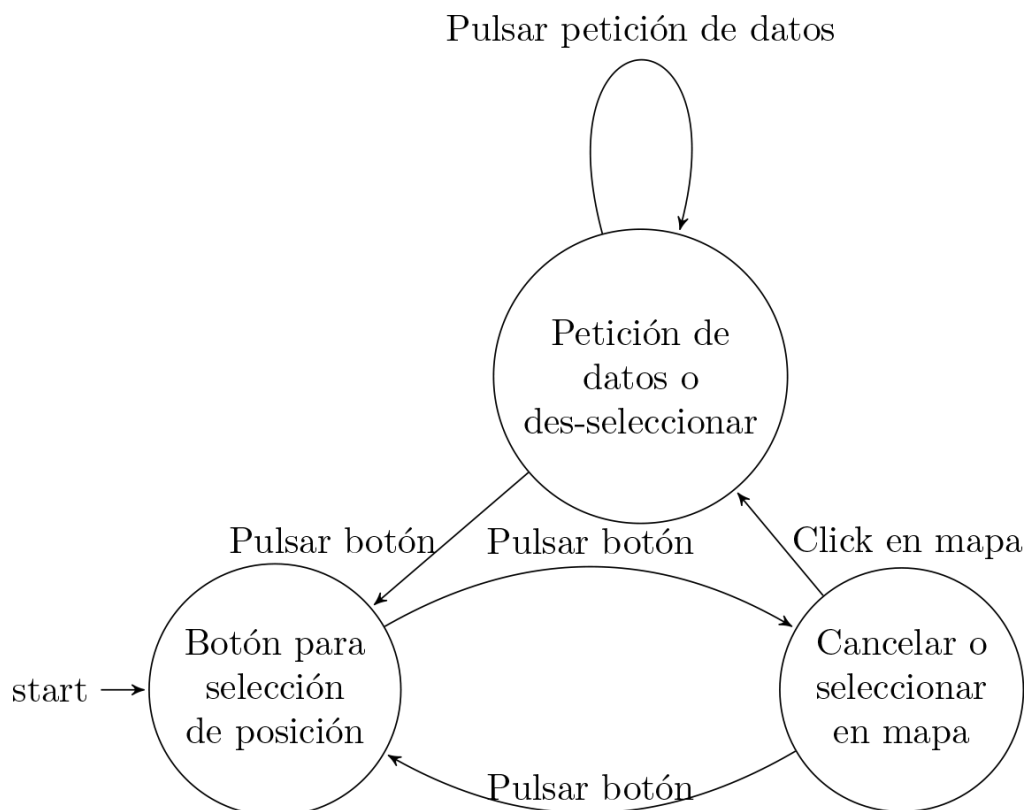


Figura 3.5: Esquema de la utilización del botón de filtrado por visibilidad desde una posición y las diferentes situaciones que se atraviesan

En un primer momento el botón servirá para comenzar a seleccionar la posición, pulsarlo hará que

- se configure el mapa para que al hacer click sobre él se ejecute una función para gestionar la selección de la posición
- y que se actualice el texto y el evento del botón para que ahora hacerlo cancele la selección de la posición, es decir, le asigna al botón un evento que desconfigura la función de click del mapa.

Si se pulsa el botón en estado anterior, ahora pasará a estar preparado para cancelar la selección de la posición, si entonces se pulsa sobre el mapa se ejecutará la función antes nombrada que:

- recibe cierta información sobre el click entre la que se encuentran su equivalencia a coordenadas geodésicas y con ello coloca un marcador en estas
- y desactiva que esta función se siga ejecutando al hacer click en el mapa, actualiza de nuevo el texto y el evento del botón para que ahora pasen a realizar una deselección de la posición y actualiza también el texto del botón de carga de datos para la fecha seleccionada.

En este momento hay una posición seleccionada en el mapa y señalada con un marcador y el estado del botón es el de des-seleccionar la posición; si ahora este es pulsado, este evento

- eliminará el marcador del mapa
- y volverá a actualizar el texto y el evento del botón para que vuelvan a realizar la selección de la posición.

Por otro lado, si en lugar de activar el botón de des-seleccionar se activa el de carga de datos para la fecha seleccionada y el input selector de fecha no está vacío, de forma similar a cómo se explica en la sección de “Selección de fecha”, se realizará la petición get proporcionando la latitud y longitud de la posición además de la fecha seleccionada por el input, y en respuesta sólo se recibirán aquellos datos orbitales visibles desde esa posición a lo largo de todas las épocas.

3.2. Funcionalidad

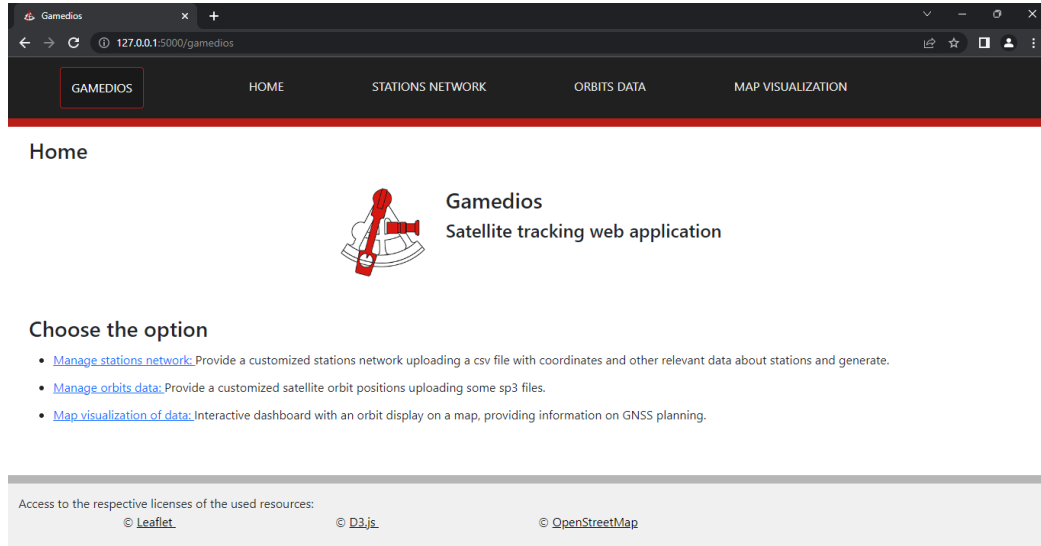


Figura 3.6: Menú principal

Un ejemplo de como este proyecto podría ser aplicado en un entorno laboral en el que se trabaje con productos de posicionamiento de satélites podría ser el de evaluar el posible desempeño de realizar correcciones utilizando una red de estaciones seleccionada para ser viable.

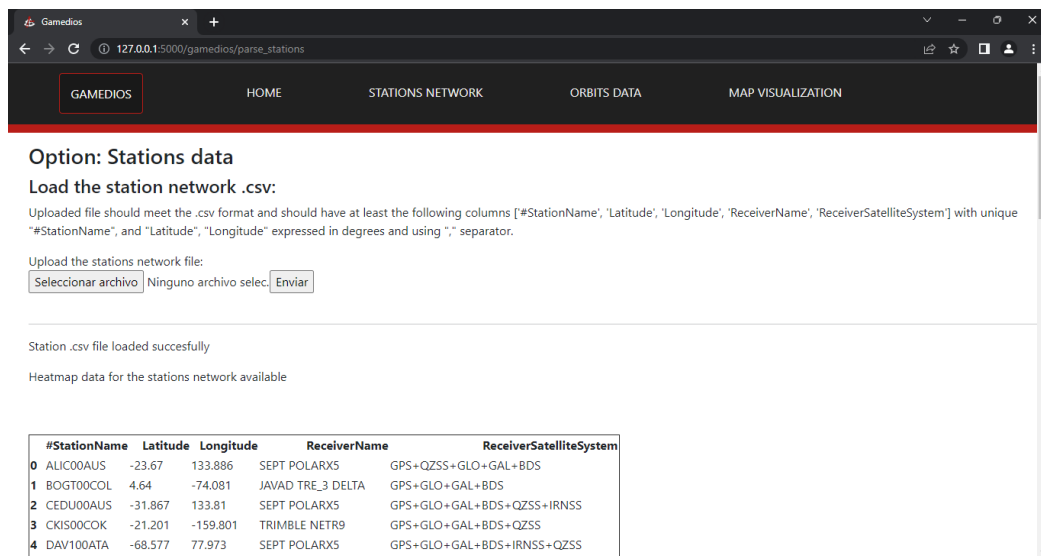


Figura 3.7: Estaciones

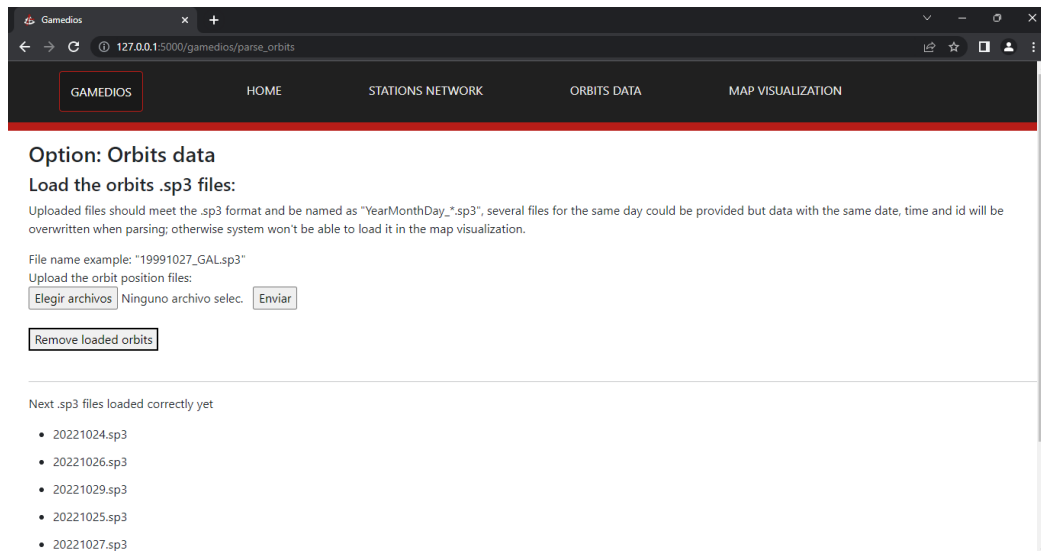


Figura 3.8: Órbitas

Lo primero sería introducir el fichero csv con el posicionamiento e información adicional de las estaciones y generar el gráfico del mapa de calor relativo a estas, e introducir los ficheros sp3 con la información de las posiciones para un cierto rango de tiempo en el que observar el movimiento completo.

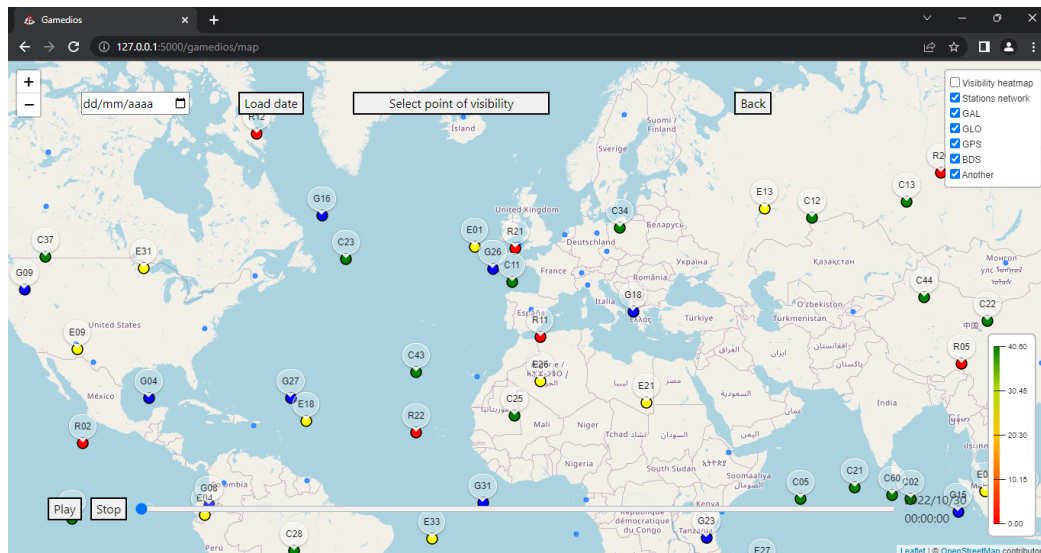


Figura 3.9: Mapa interactivo

Posteriormente desde la funcionalidad del mapa, se podría evaluar de forma visual la naturalidad del desplazamiento a lo largo del tiempo de los satélites, pudiendo detectar anomalías en este debido a errores en los cálculos o a eventos inesperados.

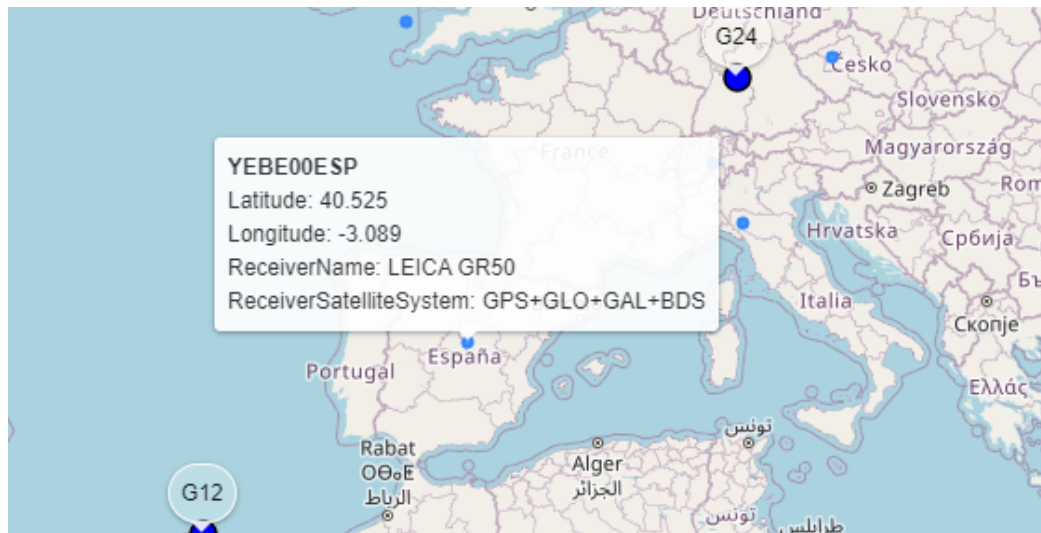


Figura 3.10: Información de una estación

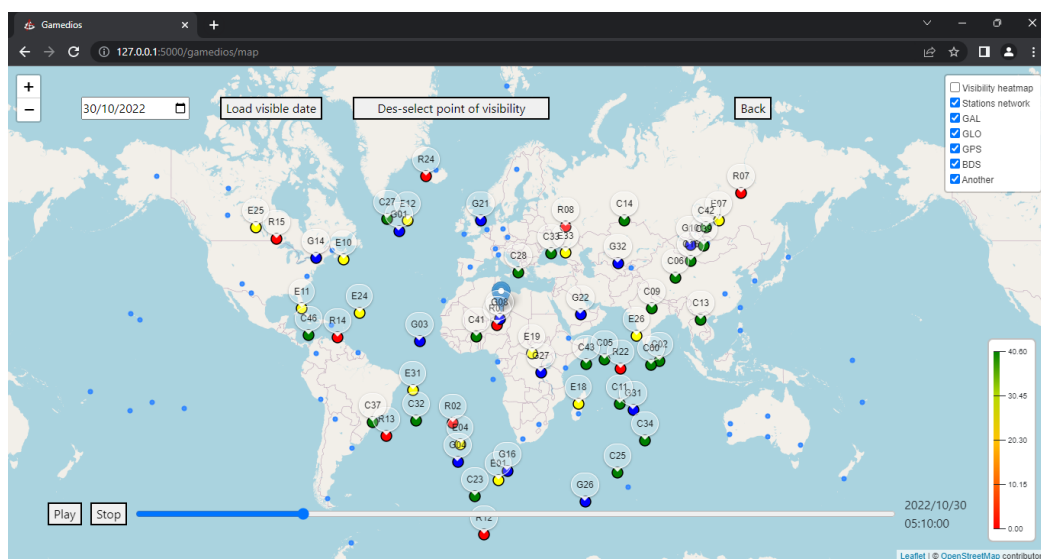


Figura 3.11: Filtrar órbitas visibles

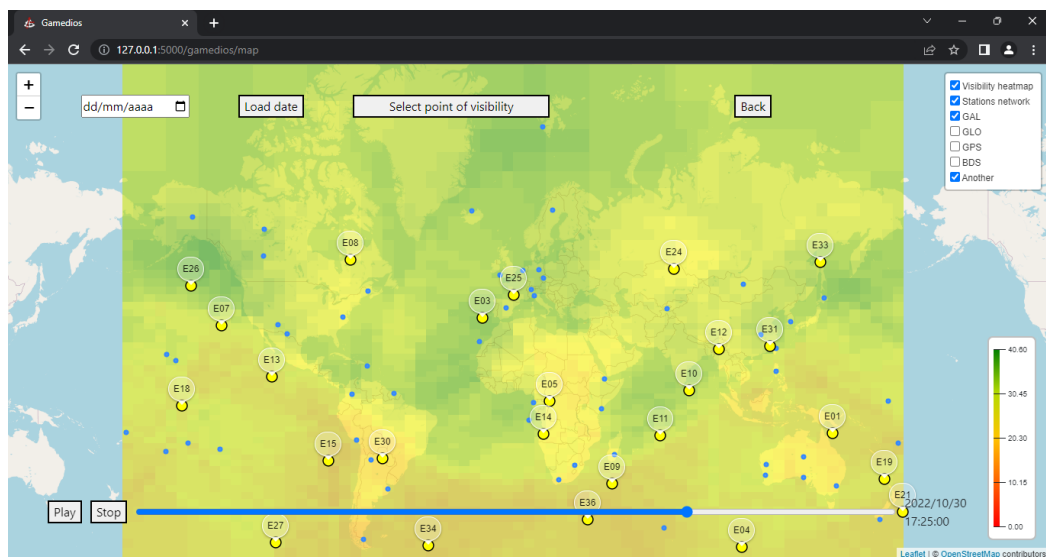


Figura 3.12: Mapa de calor

Y haciendo uso de la visualización de la red de estaciones y del mapa de calor de visibilidad, se podría evaluar la visión con la que cuentan los satélites desde un número de estaciones a lo largo de sus trayectorias para así poder identificar épocas en los que la falta de visibilidad pueda provocar imprecisiones y quizás valorar ampliar la red de estaciones en cierta región del mapa.

3.2.1. Casos de uso

1. Casos de uso de Suministrar datos necesarios

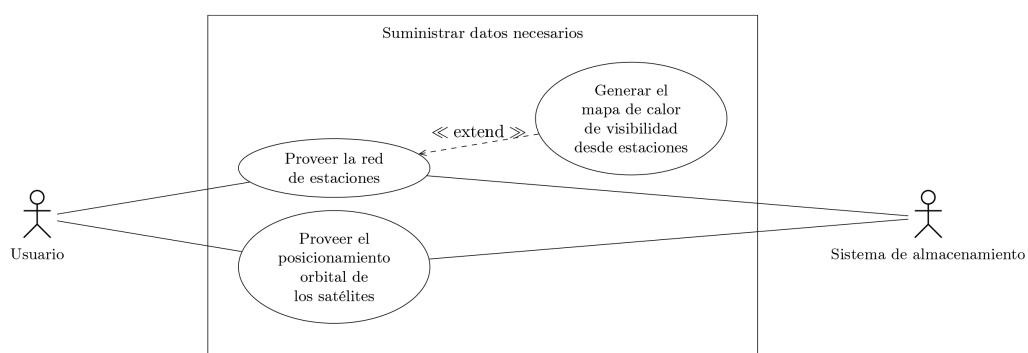


Figura 3.13: Primer caso de uso

- a) Proveer la red de estaciones
- b) Generar el mapa de calor de visibilidad desde estaciones
- c) Proveer el posicionamiento orbital de los satélites

2. Casos de uso del mapa

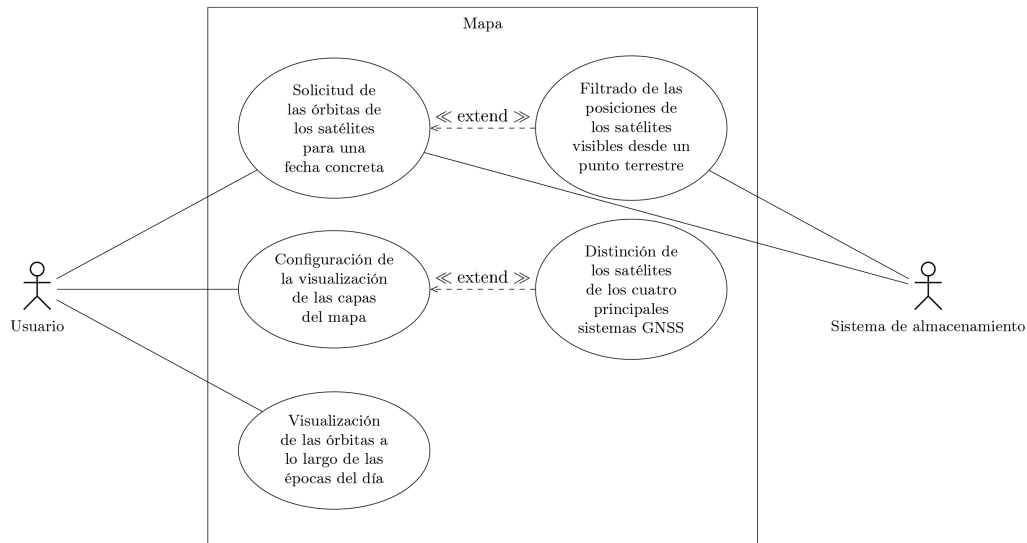


Figura 3.14: Segundo caso de uso

- a) Solicitud de las órbitas de los satélites para una fecha concreta y Filtrado de las posiciones de los satélites visibles desde un punto terrestre
- b) Configuración de la visualización de las capas del mapa y Distinción de los satélites de los cuatro principales sistemas GNSS
- c) Visualización de las órbitas a lo largo de las épocas del día

3.2.2. Despliegue

La aplicación desarrollada en el proyecto está además preparada para ser desplegada en una máquina con el sistema Ubuntu instalado o en un contenedor Docker con el mismo sistema operativo.

Cuenta con un script de python “variables.py” con los parámetros de configuración para el desempeño de la aplicación, en concreto “HOST” y “PORT” son la configuración utilizada para desplegar la aplicación.

Es necesario el acceso a internet para acceder ilustración de la superficie terrestre de “OpenStreetMap”.

Utilizando la versión 3.8 de **python** y sus siguientes requisitos de librerías:

- Flask versión 2.0.2
- numpy versión 1.22.1
- pandas versión 1.4.0

La propia aplicación ya tiene incluidas las librerías de **javascript** que emplea:

- D3.js versión 7.3.0
- leaflet versión 1.7.1

Para ejecutar la aplicación directamente en un sistema Ubuntu, es suficiente con lanzar el script principal `python3 app.py` y con ello la página es iniciada y desplegada en el host y el port de los parámetros de configuración.

Cuando la aplicación se lanza como contenedor Docker mediante el Dockerfile, la aplicación se ejecuta en el host “0.0.0.0” para permitir su acceso desde fuera del contenedor, pero debe especificarse en el comando de Docker en que host y port se quiere desplegar la aplicación y hacerlo coincidir con el port configurado en “variables.py” de la imagen. Por lo tanto el comando de Docker tiene que tener la siguiente estructura

```
docker run -p  
<host_deseado>:<port_deseado>:<port_variables.py_de_la_imagen>  
<nombre_imagen>
```

Un ejemplo de su uso para desplegar la aplicación en 127.0.0.1:4000 teniendo en cuenta que en “variables.py” se especifica PORT=5000, sería

```
docker run -p 127.0.0.1:4000:5000 gamedios
```

4

Validación

4.1. Evaluación del alcance

Haciendo referencia a los objetivos identificados al principio del proceso, los 5 puntos que componían dicha sección han sido cubiertos de la manera expuesta a continuación:

- Se ha desarrollado una herramienta preparada para que se le suministren y para soportar posiciones de satélites de las 4 principales constelaciones de satélites: Galileo, GPS, Glonass y Beidou, y adicionalmente de satélites que no pertenezcan a estas, y que las gestione agrupados según su constelación para su posterior visualización. Para su funcionamiento, al sistema también se le proporciona una red de estaciones con las posiciones y otra información adicional, que por tanto es completamente personalizable, y que tramita para realizar ciertos cálculos sobre visibilidad y su visualización más adelante.
- Utilizando la posición de las estaciones de la red proporcionada y utilizando la altura a la que se encuentran los satélites de la constelación de Galileo, se genera un mapa de calor en base al número de estaciones que tendrían visible a un satélite en esa posición del cielo de la cuadrícula generada, y esa información también se expone posteriormente en el mapa de forma visual. Además, en la visualización del mapa también se ofrece como funcionalidad el filtrado de todos los satélites visibles desde unas coordenadas a lo largo de todas las épocas del día.

- La funcionalidad principal del proyecto ha consistido en un tablero interactivo con un mapa de la superficie terrestre en el que se visualizan las posiciones de los satélites a lo largo de las épocas agrupados por capas según su constelación para poder filtrarlos y asignándole un color característico a cada constelación, para que sean distinguibles a la vista. También dentro de esta funcionalidad se visualizan formando otras capas independientes la posición e información de la red de estaciones y los cálculos de visibilidad del mapa de calor explicado en el anterior punto; además de que se ofrece la posibilidad de filtrar los satélites visibles desde un punto de la superficie terrestre a lo largo del día.
- En el desarrollo de la herramienta se han aplicado estándares utilizados en el ámbito del rastreo satelital, como la equivalencia de la letra de los nombres de los satélites con la constelación a la que pertenecen y el formato SP3 para las posiciones de los satélites, y sistemas de referencia geodésico para el uso de coordenadas. Se ha diseñado la herramienta con la pretensión de características como la flexibilidad o la capacidad de adaptación seleccionando para las distintas necesidades surgidas las más extendidas librerías en el ámbito como Pandas o Numpy, o aquellas que ofrecían un mayor abanico de posibilidades, como D3.js, y se ha procurado que todas las librerías empleadas presentasen una licencia libre.

4.2. Métricas

Con la intención de realizar un análisis dinámico de la implementación de la aplicación se ha seleccionado la herramienta “Lighthouse” (versión 9.6.7) de Google debido a su compatibilidad con las herramientas empleadas en el proyecto y la sencillez de su uso.

El reporte del análisis ofrece 5 secciones de métricas sobre la aplicación web objeto del estudio, cada una con una puntuación global de 0 a 100 sobre su desempeño.

Cada una de estas secciones está constituida por una serie de métricas, estas se basan en la obtención de un valor relativo a la aplicación sin procesar, que posteriormente es interpretado según el carácter de la métrica como un porcentaje (asociado además a un color) que representa su impacto (positivo o negativo) en el desempeño de la aplicación. Donde:

- Del 0 al 49 (rojo): Significa un valor sin procesar malo.
- Del 50 al 89 (naranja): Significa un valor sin procesar que necesita mejorar.
- Del 90 al 100 (verde): Significa un valor sin procesar bueno.

Información detallada sobre el proceso de obtención de los valores para cada métrica y la valoración de estos como porcentaje, puede consultarse a través de la documentación de la herramienta.

4.2.1. Página que utiliza la plantilla HTML del Mapa

La herramienta ha ido siendo aplicada sobre la página web con la principal funcionalidad del mapa en diversos puntos de su desarrollo para el seguimiento de sus funcionalidades, pero aquí se describirán y compararán los análisis realizados en dos puntos concretos:

- Un análisis base realizado sobre la implementación con funcionalidad completa pero no optimizada, a partir del cual se seleccionaron e implementaron ciertas modificaciones sugeridas por el análisis.
- Un análisis posterior realizado sobre la implementación funcional enriquecida con las medidas de mejora seleccionadas, es decir, la implementación final que se ha ido describiendo a lo largo de toda la sección de Diseño.

Es importante destacar que todas las funcionalidades y características del proyecto desgranadas con anterioridad a lo largo de las secciones de la memoria describen esta implementación final y definitiva mencionada de la aplicación web, en esta sección se mencionarán los planteamientos funcionales pero no optimizados que durante parte del desarrollo fueron utilizados en la aplicación pero sólo para contrastarlos con sus versiones finales.

Los dos reportes en formato PDF con los resultados de estos dos análisis pueden ser consultados para más información desde el anexo del documento.

A continuación se desglosan los resultados para cada una de las secciones de métricas obtenidos en el primer y segundo análisis realizados, a través de su relación con los cambios sugeridos implementados para la mejora de la aplicación.

Rendimiento (Performance):

Con una puntuación general de 73 (naranja), representa un balance de la experiencia extraída por un usuario en su interacción a tiempo real con la aplicación.

Contiene las siguientes métricas:

- **First Contentful Paint (FCP):** valor de 2.2s (rojo).
Mide cuanto tiempo le toma al navegador mostrar el primer elemento del contenido DOM (Document Object Mode) después de que un usuario navegue por la página.

- **Time to Interactive (Tti):** valor de 2.3s (verde).
Cuantifica el tiempo que le toma a la página volver completamente interactiva: muestre contenido útil, los controladores de eventos han sido registrados para la mayoría de los elementos visibles y la página responde a las interacciones con el usuario en menos de 50ms.
- **Speed Index:** valor de 2.2s (naranja).
Evalúa con cuanta velocidad el contenido es desplegado visualmente durante la carga de la página.
- **Total Blocking Time (TBT):** valor de 0ms (verde).
Mide la cantidad de tiempo que la página se encuentra bloqueada para no responder a la interacción con el usuario.
- **Largest Contentful Paint (LCP):** valor de 2.3s (rojo).
Tasa el tiempo que tarda la aplicación en renderizar el elemento, ya sea imagen o bloque de texto, que presenta mayor tamaño en la visualización desde el inicio de la carga.
- **Cumulative Layout Shift (CLS):** valor de 0.003 (verde).
Cuantifica la frecuencia con la que en la experiencia de un usuario con la aplicación se producen cambios de diseño inesperados que dificulten su navegación.

Como esta sección se trata de una de las que presentan menor puntuación, se ha decidido implementar algunas de las medidas sugeridas por el análisis para mejorarla. En concreto, involucradas en su cómputos son las métricas FCP, SI y LCP las que presentan menores valoraciones, por lo que se escogió realizar la siguientes variaciones relacionadas con ellas:

- Permitir la compresión de texto:
Que se ha implementado mediante la utilización de la librería “Flask-Compress”, que extiende la funcionalidad de la librería “Flask” permitiendo la compresión de texto en sus solicitudes al añadir las cabeceras necesarias para la compresión de las respuestas y de los datos de las respuestas. Está recomendada especialmente para la compresión de datos estáticos y para su desempeño con cada solicitud comprueba si se trata de uno de los tipos MIME que se pueden comprimir y si cliente y servidor coinciden en algún algoritmo de compresión.
- Realizar el acceso a toda la información de un volumen considerable a través de peticiones, en vez de completar la plantilla HTML con esos datos:
Para permitir un cargado más dinámico de la información de mayor tamaño y aprovechar al máximo la funcionalidad de compresión ofrecida por la librería antes mencionada, se modificó la implementación inicial: en la que

los datos de estaciones, el mapa de calor y los datos orbitales del día más recientes para el que había, se proporcionaban introducidos en la plantilla HTML; y esta fue sustituida posteriormente por otro esquema en la implementación final: de manera que estos datos fuesen obtenidos por la página web mediante la realización de unas petición “get” pertinentes.

Gracias a las modificaciones, se ha conseguido elevar la valoración de la métrica hasta el valor de 94 (verde), todas las métricas ahora están catalogadas con unos mejores resultados en el rango verde, salvo “Speed Index” y “Largest Contentful Paint” que han mejorado hasta el rango naranja con un valor de 1.9s y 1.2s respectivamente.

Accesibilidad (Accessibility)

Con una puntuación general de 85 (naranja), hace referencia a la capacidad de la aplicación para ser utilizada con flexibilidad en diferentes contextos y por personas con diferentes condiciones.

De forma similar a como ocurría con la anterior sección, esta presenta una puntuación notablemente menor por lo que se ha escogido implementar la siguiente selección de medidas sugeridas por el análisis:

- Añadir etiquetas asociadas a todos los elementos de formulario:
Las siguientes etiquetas `<input>` son las señaladas por al análisis al no presentar una etiqueta que corresponda a ellos: el elemento que permite introducir la fecha para la que se van a visualizar datos y la barra de desplazamiento que permite seleccionar la hora y minuto que se visualiza. Ambos elementos se ha valorado que quedan etiquetados visualmente respectivamente por el botón de carga de los datos y los elementos de la línea tiempo de su alrededor, de tal manera que sólo se les ha añadido una etiqueta de texto invisible para los lectores de pantalla mediante el argumento “aria-label”.
- Permitir escalabilidad por parte del usuario de todos los elementos:
Para realizar esto se ha permitido a estas etiquetas `<meta name="viewport">` o área visible por el usuario, el único elemento señalado en esta sugerencia, su escalabilidad por el usuario mediante el argumento `user-scalable=yes`.

También como recomendación del análisis realizado, se ha comprobado manualmente el cumplimiento de ciertas medidas que se han estimados aplicables a esta implementación:

- Los elementos interactivos pueden ser seleccionados y manipulados a través de acciones de teclado:

Se ha podido acceder a todos los elementos interactivos de esta página web: selección de fecha, petición de datos, configuración de la visibilidad de las capas, manipulación de la línea de tiempo; salvo aquella funcionalidad relativa a la selección en el mapa de un punto desde el que evaluar la visibilidad de satélites.

- Los elementos interactivos (botones y links) cuentan con indicaciones sobre su propósito, estado y modo de uso: El contexto en el que se presentan estos elementos interactivos cuenta con indicaciones que indican su utilidad, condición y utilización.

En los resultados del análisis realizado tras implementar estos cambios queda reflejada la mejoría de la evaluación de esta sección de métrica, elevándose esta hasta la máxima puntuación de 100 (verde).

Mejores prácticas (Best Practices)

Con una puntuación general de 92 (verde), manifiesta el empleo de prácticas y modelos cuyo uso supone una ventaja para la aplicación web porque atienden a estándares o resultan favorables y concretos de aplicar.

Debido a que esta es una de las secciones con una puntuación más positiva, y tras la revisión de las medidas sugeridas y de aquellas que ya estaba cumplidas, la decisión fue de no añadir a la implementación ninguna otra modificación para esta sección.

Tras los cambios implementados en relación al resto de métricas se ha comprobado que la valoración de esta sección se ha mantenido con la misma puntuación de 92 (verde) y con el cumplimiento de las mismas sugerencias que podía verse en el resultado del análisis previo.

SEO (Search Engine Optimization)

Con una puntuación de 90 (verde), refleja una valoración sobre la “Optimización en Motores de Búsqueda” de la aplicación web. Es una métrica con cierta complejidad pero que a grandes rasgos abarca la comprobación de un conjunto de prácticas enfocadas a la mejora de su posicionamiento en los resultados de motores de búsqueda.

De la misma manera que la anterior sección, esta presenta una de las mejores puntuaciones, es por esto que tras la revisión de sus sugerencias y los criterios ya cumplidos, sólo se decidió realizar el siguiente cambio adicional en la implementación:

- Añadir una descripción de la página web:
Se añade la siguiente descripción a la plantilla HTML de la página web mediante una etiqueta en la sección “head” con el siguiente formato

```
<meta name="description" content="The description">.
```

Tras el análisis posterior a las modificaciones realizadas, se puede ver como la anterior sugerencia queda cubierta con el cambio y esto logra que la puntuación de la aplicación web para esta sección de métricas asciendo hasta el valor máximo de 100 (verde).

PWA (Progressive Web App)

Con una puntuación de “ningún hito alcanzado”, esta sección valida de forma discreta los aspectos requeridos para que una aplicación web sea considerada “Aplicación web progresiva” y esta es un modelo de aplicación web que sirve como forma de aprovechar nuevas capacidades de los navegadores modernos. Para esto debe cumplir ser capaz de ser instalada y utilizada por todos los usuarios, independientemente del tamaño o tipo de entrada proporcionada.

La aplicación web desarrollada en este proyecto se ideó orientada para su integración en un sistema de archivos y es por esto que no se ha planteado su abarcamiento como “Aplicación web progresiva”; con dicho motivo, esta sección no ha sido tomada en cuenta a la hora de la implementación de mejoras sugeridas tras el análisis previo y no presenta ninguna variación en el análisis posterior.

4.2.2. Páginas que utilizan la plantilla del Menú principal

Siguiendo la dinámica antes descrita, las tres páginas web que utilizan esta plantilla: menú principal, manejo de la red de estaciones y manejo de los datos orbitales respectivamente, han sido evaluadas en dos ocasiones: una con la implementación funcional pero no optimizada, y otra con la implementación que contaba con los cambios sugeridos añadidos.

Los seis reportes en formato PDF con los resultados de los análisis pueden ser consultados también en el anexo del documento.

Como anteriormente ya se ha hecho una descripción detallada de las secciones de métricas, sus definiciones y sus métricas internas, a continuación sólo se mencionarán los valores notablemente bajos en las valoraciones de estas páginas web, las mejoras que se han implementado en relación a ellas y los cambios que han provocado en las puntuaciones.

Rendimiento (Performance)

Con una puntuación de 98 (verde) para las tres páginas web, al encontrarse esta entre los valores más altos obtenidos en las métricas no se ha decidido implementar para mejorarla ningún cambio tras examinar las medidas ya cumplidas y las aconsejado cumplir que aportó el análisis previo.

Sin embargo, tras los cambios realizados como sugerencias para otras secciones de métricas, la valoración de esta también se ha visto afectada y ha logrado ascender en las tres páginas hasta la máxima puntuación de 100 (verde).

Accesibilidad (Accessibility)

Con unas puntuaciones de 59 (naranja), 65 (naranja) y 67 (naranja) para las páginas web del menú principal, manejo de la red de estaciones y manejo de los datos orbitales respectivamente, esta es una de las secciones de métricas que presentaba una puntuación considerablemente baja y es por esto que se ha decidido añadir la misma selección de cambios que se realizaron en la plantilla HTML para la funcionalidad del mapa y adicionalmente otro cambio nuevo:

- Añadir etiquetas asociadas a todos los elementos de formulario:
En este caso se tratan de las etiquetas de tipo `<input>` utilizados en las páginas de manejo de la red de estaciones y de los datos orbitales para subir y enviar los archivos `<input type="file">` y `<input type="submit">`.
- Permitir escalabilidad por parte del usuario de todos los elementos:
De forma análoga, para este cambio se le ha añadido a las etiquetas de tipo `<meta name="viewport">` de la plantilla HTML el argumento `user-scalable=yes`.
- Añadir el suficiente contraste entre los colores del fondo y de los elementos en primer plano:
Se refiere concretamente a la etiqueta `<footer>` de esta plantilla HTML, donde el contraste entre el fondo gris y el color azul por defecto que se le aporta a las etiquetas de tipo `<a>` no es lo suficientemente claro. Para solucionar esto se ha optado por añadir a la hoja de estilo CSS el cambio de color de estas secciones `<a>` dentro del `<footer>` al color negro.
- Añadir texto alternativo descriptivo a la imagen de la aplicación web en la página web del menú principal:
En esta página se presenta una etiqueta de tipo `` para mostrar el logo de aplicación y para añadir un texto describiendo la imagen se ha utilizado el argumento `alt="Texto descriptivo"` en ella.

Las variaciones en la implementación antes mencionadas han conseguido mejorarla y esto se ha podido ver reflejado en la puntuación obtenida en el análisis

superior, donde esta métrica recibió el máximo valor de 100 (verde).

Mejores prácticas (Best Practices)

Con una puntuación de 100 (verde) en las tres páginas web, esta sección ya cuenta con la máxima puntuación posible por lo que no se ha determinado implementar medidas adicionales para su mejora.

Tras añadir el resto de cambios relativos a las otras secciones de métricas, no se observaron cambios en la evaluación de la sección y la puntuación se mantuvo en 100.

SEO (Search Engine Optimization)

Con una puntuación de 80 para la página web del menú principal y de 89 para las páginas del manejo de la red de estaciones y de los datos orbitales, esta es la otra sección que presenta una puntuación notablemente baja y para cuya mejora, igual que para la página web de la funcionalidad del mapa, se implementaron los siguientes cambios:

- Añadir una descripción alternativa a la imagen de la aplicación web en la página web del menú principal:
De nuevo este cambio es sugerido también para mejorar esta sección.
- Añadir una descripción de la página web:
Para aportar la descripción a estas páginas se añade la siguiente etiqueta a la plantilla HTML

```
<meta name="description" content="Web page with  
a header and a footer used to display information  
about {{description | safe}}.">
```

Al contrario de como ocurría en la plantilla HTML de la funcionalidad del mapa, que la descripción estaba completamente añadida a la plantilla, en este caso la etiqueta cuenta con un espacio a completar `{{description | safe}}` por la aplicación “Flask”. Este espacio es rellenado al cargar cada página y permite que estas cuenten con una descripción personalizada de su funcionalidad.

Gracias a esta modificación, la sugerencia queda cubierta y la puntuación de esta sección de métrica asciende tras el análisis posterior al máximo valor de 100 (verde).

PWA (Progressive Web App)

Con una puntuación de “ningún hito alcanzado”, como ocurría con la evaluación de la página web de la funcionalidad del mapa, por cómo está ideada la aplicación web desarrollada en este proyecto no eso uno de sus propósitos funcionar como “Aplicación web progresiva”, y por ello esta sección no ha sido tenida en cuenta para la mejora de la aplicación web.

5

Conclusiones y trabajos futuros

5.1. Cumplimiento de los objetivos

En conclusión, con este proyecto se ha cumplido con los objetivos planteados de la siguiente manera: Se ha desarrollado una **aplicación web** para **rastreo satelital** enfocada especialmente en facilitar el tratado con información de **posicionamiento** y **visibilidad** de los **satélites** de los principales sistemas GNSS y una **red de estaciones**.

Esto se ha materializado como la incorporación de un tablero interactivo con un mapa de la superficie terrestre en el que se han señalado distintas coordenadas; todos los elementos descritos a continuación conforman capas independientes que son expuestas sobre el mapa con la capacidad de configurar su visualización u ocultación.

Para **mostrar las posiciones de la red de estaciones terrestre** se han señalado estas en el mapa junto a información de utilidad sobre las estaciones. Toda esta información debe ser proporcionados por el usuario como un fichero en un formato específico de “CSV”.

Y para **mostrar las órbitas descritas por los satélites** se han utilizado las coordenadas de las proyecciones sobre la superficie de sus posiciones reales en el cielo terrestre. Estas coordenadas son señalizadas junto a la información de la constelación a la que pertenece el satélite y puede ser visualizada su variación a lo largo del tiempo (cómo se van desplazando los satélites describiendo sus órbitas) de forma dinámica mediante una línea de tiempo de 24 horas. Estos datos deben

ser proporcionados por el usuario como ficheros en formato “SP3”.

También se han implementado el **cálculo de visibilidad entre objetos posicionados en el cielo y en la superficie terrestres** mediante el criterio de que el valor de elevación de las coordenadas horizontales deba ser superior a un umbral configurado. Y esta funcionalidad se ha utilizado para el filtrado de posiciones de los satélites para sólo mostrar aquellos visibles desde una posición seleccionada por el usuario, y para generar un elemento gráfico de un mapa de calor con el número de estaciones que tendrían visible a un satélite en cada posición concreta del cielo terrestre (a la altura a la que orbitan los satélites de la constelación Galileo), y esta información se proyecta sobre la superficie para poder ser mostrada en el mapa.

5.2. Campos de desarrollo

En esta sección se proponen varias direcciones en las que se podría continuar desarrollando la aplicación web para extender la amplitud de los objetivos cumplidos en el proyecto:

- Computar las posiciones orbitales a visualizar en el mapa con una frecuencia configurable a partir de la parametrización de las órbitas de Kepler descritas por los satélites y obtenidas de ficheros con formato “TLE”.
- Proporcionar otras métricas de interés sobre el sistema, como por ejemplo el valor “DOP” (Dilution of precision) de una serie de posiciones en la superficie terrestre, relativo a la propagación de error como efecto matemático de la geometría del satélite en la precisión de la medición posicional. También se podrían generar gráficos 2D sobre el número de estaciones con visibilidad de cada satélite a lo largo del tiempo.
- Preparar el sistema para poder ser adicionalmente integrado en un servidor completamente como servicio web, por ejemplo mediante la utilización de una base de datos, lo cual por otro lado podría agilizar los procesos de petición de datos planteándolos para aprovechar este nuevo recurso.

Bibliografía

- [1] *Global Navigation Satellite System (GNSS) manual*. International Civil Aviation Organization, 2012.
- [2] S. Jin, *Global Navigation Satellite Systems: Signal, theory and applications*. InTech, 2014.
- [3] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and practice*. Springer-Verlag, 1992.
- [4] *A guide to coordinate systems in Great Britain: An introduction to mapping coordinate systems and the use of GPS datasets with Ordnance Survey*. Ordnance Survey, 2010.
- [5] P. Osborne, *The mercator projections*. Peter Osborne, 2008.
- [6] C. M. P. do and S. d. L. J. Claudio, *Geometría diferencial de Curvas y Superficies*. Alianza Editorial, 1995.
- [7] R. R. Bate, D. D. Mueller, J. E. White, and W. W. Saylor, *Fundamentals of astrodynamics*. Dover Publications, Inc., 1971.
- [8] O. Foundation, “Openstreetmap,” 2023, accedido el 05 de junio de 2023. [Online]. Available: <https://www.openstreetmap.org>
- [9] M. Bostock and Observable, Inc, “D3.js,” 2022, accedido el 05 de junio de 2023. [Online]. Available: <https://d3js.org>
- [10] V. Agafonkin, “Leaflet,” 2020, accedido el 05 de junio de 2023. [Online]. Available: <https://leafletjs.com>
- [11] NumPy Developers, “Numpy,” 2022, accedido el 05 de junio de 2023. [Online]. Available: <https://numpy.org/doc/stable/index.html>
- [12] The Pandas development team, “Pandas,” 2022, accedido el 05 de junio de 2023. [Online]. Available: <https://pandas.pydata.org/docs/index.html>
- [13] Pallets, “Flask,” 2021, accedido el 05 de junio de 2023. [Online]. Available: <https://flask.palletsprojects.com/en/2.2.x/>
- [14] T. Mansencal, “Flask-compress,” 2022, accedido el 05 de junio de 2023. [Online]. Available: <https://pypi.org/project/Flask-Compress/>
- [15] Z. Foundation and Contributors, “Waitress,” 2022, accedido el 05 de junio de 2023. [Online]. Available: <https://pypi.org/project/waitress/>
- [16] S. Hilla, National Geodetic Survey, and National Ocean Service, NOAA, “The extended standard product 3 orbit format (sp3-c),” Tech. Rep., 2010, accedido el 05 de junio de 2023. [Online]. Available: <http://www.epncb.oma.be/ftp/data/format/sp3c.txt>
- [17] Internet Engineering Task Force, H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub, and S. Hagen, “The geojson format,” Tech. Rep., 2016, accedido el 05 de junio de 2023. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc7946>

- [18] H. Moritz, "Geodetic reference system 1980," Tech. Rep., 1985.
- [19] *World Geodetic System - 1984 (WGS-84) manual*. Internat. Civil Aviation Organization, 2002.
- [20] G. Developers, "Lighthouse," 2023, accedido el 05 de junio de 2023. [Online]. Available: <https://developer.chrome.com/docs/lighthouse/overview/>

Apéndice



Análisis de métricas

A.1. Análisis base

A.1.1. Página de visualización del mapa



Performance



Accessibility



Best Practices



SEO



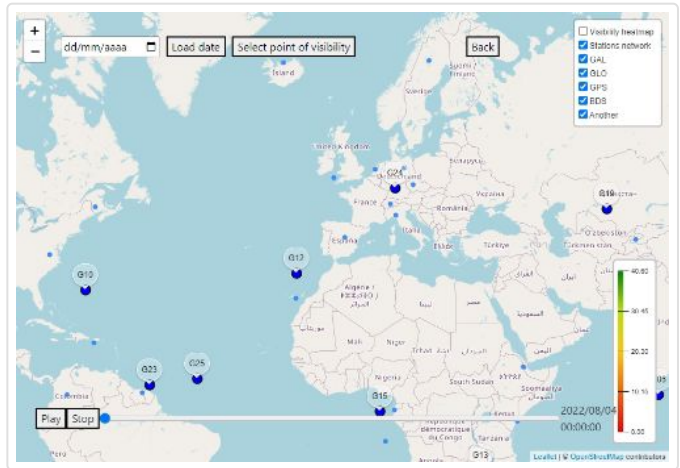
PWA



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

- ▲ 0–49
- 50–89
- 90–100



METRICS

[Expand view](#)

▲ First Contentful Paint
2.2 s

■ Speed Index
2.2 s

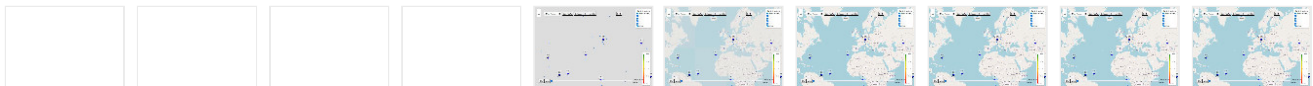
▲ Largest Contentful Paint
2.6 s

● Time to Interactive
2.3 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0.003

[View Treemap](#)



Show audits relevant to: **All** [FCP](#) [TBT](#) [LCP](#) [CLS](#)

OPPORTUNITIES

Opportunity

Estimated Savings

▲ Enable text compression **1.46 s**

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. [Learn more](#). FCP LCP

URL	Transfer Size	Potential Savings
/gamedios/map (127.0.0.1)	1,148.9 KiB	1,025.0 KiB
...dist/d3.js (127.0.0.1)	563.5 KiB	430.9 KiB
...leaflet/leaflet-src.js (127.0.0.1)	421.1 KiB	315.7 KiB
...leaflet/leaflet.css (127.0.0.1)	13.9 KiB	10.7 KiB
...js/Map.js (127.0.0.1)	10.3 KiB	7.9 KiB
...js/map_lib.js (127.0.0.1)	4.3 KiB	2.6 KiB
...js/timeline_lib.js (127.0.0.1)	2.2 KiB	1.6 KiB

▲ Eliminate render-blocking resources

1 s ^

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more](#). FCP LCP

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
...leaflet/leaflet.css (127.0.0.1)	14.2 KiB	110 ms
...css/Map.css (127.0.0.1)	1.5 KiB	110 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	340 ms
...dist/d3.js (127.0.0.1)	563.8 KiB	1,070 ms
...leaflet/leaflet-src.js (127.0.0.1)	421.4 KiB	910 ms

▲ Reduce initial server response time

0.88 s ^

Keep the server response time for the main document short because all other requests depend on it. [Learn more](#). FCP LCP

URL	Time Spent
/gamedios/map (127.0.0.1)	980 ms

■ Reduce unused JavaScript

0.42 s ^

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn more.](#) LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	563.8 KiB	443.0 KiB
...leaflet/leaflet-src.js (127.0.0.1)	421.4 KiB	148.6 KiB
...map/Map.js	55.6 KiB	23.7 KiB
...layer/vector/Canvas.js	11.1 KiB	9.4 KiB
...layer/Popup.js	14.2 KiB	8.0 KiB
...dom/Draggable.js	6.6 KiB	4.8 KiB
...layer/marker/Marker.js	8.9 KiB	4.7 KiB

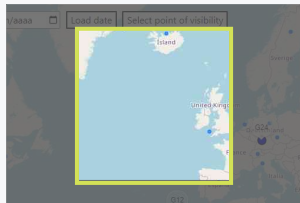
Minify JavaScript 0.4 s ^

Minifying JavaScript files can reduce payload sizes and script parse time. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...leaflet/leaflet-src.js (127.0.0.1)	421.4 KiB	231.5 KiB
...dist/d3.js (127.0.0.1)	563.8 KiB	168.3 KiB
...js/Map.js (127.0.0.1)	10.5 KiB	3.4 KiB

Preload Largest Contentful Paint image 0.25 s ^

Preload the image used by the LCP element in order to improve your LCP time. [Learn more.](#) LCP

URL	Potential Savings
<div style="display: flex; align-items: flex-start;">  <div style="font-family: monospace; font-size: 0.9em;"> <div style="margin-bottom: 5px;">div.leaflet-pane ></div> <div style="margin-bottom: 5px;">div.leaflet-layer ></div> <div style="margin-bottom: 5px;">div.leaflet-tile-container ></div> <div style="margin-bottom: 5px;">img.leaflet-tile</div> </div> <pre style="font-family: monospace; font-size: 0.8em; margin-top: 5px;"> </pre> </div> <div style="margin-left: 20px;"> <p>...3/2.png (c.tile.openstreetmap.org)</p> </div>	<p>250 ms</p>

These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

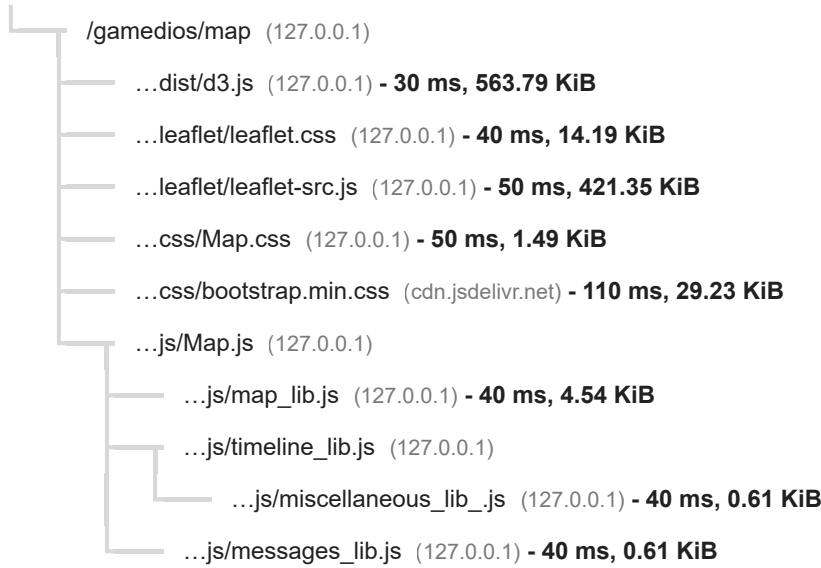
DIAGNOSTICS

○ Avoid chaining critical requests — 8 chains found ^

The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. [Learn more.](#) FCP LCP

Maximum critical path latency: **1,280 ms**

Initial Navigation



○ Keep request counts low and transfer sizes small — 28 requests • 2,341 KiB ^

To set budgets for the quantity and size of page resources, add a budget.json file. [Learn more.](#)

Resource Type	Requests	Transfer Size
Total	28	2,340.8 KiB
Document	1	1,149.0 KiB
Script	7	1,004.0 KiB
Image	16	114.9 KiB
Stylesheet	3	44.9 KiB
Other	1	28.0 KiB
Media	0	0.0 KiB
Font	0	0.0 KiB

Resource Type	Requests	Transfer Size
Third-party	17	144.1 KiB

○ Largest Contentful Paint element — 1 element found ^

This is the largest contentful element painted within the viewport. [Learn More](#) LCP

Element



```
div.leaflet-pane > div.leaflet-layer > div.leaflet-tile-container > img.leaflet-tile

```

○ Avoid large layout shifts — 1 element found ^

These DOM elements contribute most to the CLS of the page. CLS

Element	CLS Contribution
 <div style="margin-left: 20px;"> <p>Play Stop 2022/08/04 00:00:00</p> <pre><div id="botton_buttons" class="over_map"></pre> </div>	0.003

○ Avoid long main-thread tasks — 1 long task found ^

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay.

[Learn more](#) TBT

URL	Start Time	Duration
...leaflet/leaflet-src.js (127.0.0.1)	2,193 ms	91 ms

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (29)

Hide

● Properly size images ^

Serve images that are appropriately-sized to save cellular data and improve load time. [Learn more](#).

● Defer offscreen images ^

Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. [Learn more.](#)

● Minify CSS — Potential savings of 3 KiB ^

Minifying CSS files can reduce network payload sizes. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...leaflet/leaflet.css (127.0.0.1)	14.2 KiB	2.7 KiB

● Reduce unused CSS — Potential savings of 29 KiB ^

Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	28.8 KiB

● Efficiently encode images ^

Optimized images load faster and consume less cellular data. [Learn more.](#)

● Serve images in next-gen formats ^

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn more.](#)

● Preconnect to required origins — Potential savings of 80 ms ^

Consider adding `preconnect` or `dns-prefetch` resource hints to establish early connections to important third-party origins. [Learn more.](#) FCP LCP

URL	Potential Savings
https://b.tile.openstreetmap.org	80 ms

URL	Potential Savings
https://c.tile.openstreetmap.org	80 ms
https://a.tile.openstreetmap.org	80 ms

Avoid multiple page redirects ^

Redirects introduce additional delays before the page can be loaded. [Learn more.](#) FCP LCP

Preload key requests ^

Consider using ``<link rel=preload>`` to prioritize fetching resources that are currently requested later in page load. [Learn more.](#) FCP LCP

Use HTTP/2 ^

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. [Learn more.](#)

Use video formats for animated content ^

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. [Learn more](#)

LCP

Remove duplicate modules in JavaScript bundles ^

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. TBT

Avoid serving legacy JavaScript to modern browsers ^

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. [Learn More](#) TBT

Avoids enormous network payloads — Total size was 2,341 KiB ^

Large network payloads cost users real money and are highly correlated with long load times.

[Learn more.](#) LCP

Show 3rd-party resources (4)

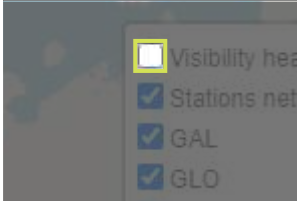
URL	Transfer Size
/gamedios/map (127.0.0.1)	1,149.0 KiB
...dist/d3.js (127.0.0.1)	563.8 KiB
...leaflet/leaflet-src.js (127.0.0.1)	421.4 KiB
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB
...images/icon192.png (127.0.0.1)	28.0 KiB
...4/2.png (a.tile.openstreetmap.org)	15.1 KiB
...leaflet/leaflet.css (127.0.0.1)	14.2 KiB
...4/3.png (b.tile.openstreetmap.org)	13.5 KiB
...js/Map.js (127.0.0.1)	10.5 KiB
...4/4.png (c.tile.openstreetmap.org)	9.7 KiB

● Uses efficient cache policy on static assets — 0 resources found ^

A long cache lifetime can speed up repeat visits to your page. [Learn more.](#)

● Avoids an excessive DOM size — 240 elements ^

A large DOM will increase memory usage, cause longer [style calculations](#), and produce costly [layout reflows](#). [Learn more.](#) TBT

Statistic	Element	Value
Total DOM Elements		240
Maximum DOM Depth	 <pre>div.leaflet-control-layers-overlays > label > div > input.leaflet-control-layers-selector <input type="checkbox" class="leaflet-control-layers- selector"></pre>	11

Statistic	Element	Value
Maximum Child Elements	 <div data-bbox="730 168 1300 280"> <p>div.leaflet-pane > div.leaflet-pane > svg.leaflet-zoom-animinated > g</p> <p><g></p> </div>	90

○ User Timing marks and measures ^

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. [Learn more.](#)

● JavaScript execution time — 0.2 s ^

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

URL	Total CPU Time	Script Evaluation	Script Parse
...leaflet/leaflet-src.js (127.0.0.1)	242 ms	157 ms	7 ms
Unattributable	182 ms	2 ms	0 ms
/gamedios/map (127.0.0.1)	143 ms	20 ms	15 ms

● Minimizes main-thread work — 0.6 s ^

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more](#) TBT

Category	Time Spent
Other	186 ms
Script Evaluation	185 ms
Style & Layout	85 ms
Parse HTML & CSS	41 ms
Garbage Collection	41 ms
Script Parsing & Compilation	34 ms
Rendering	20 ms

● All text remains visible during webfont loads ^

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading.

[Learn more.](#) FCP LCP

● Minimize third-party usage — Third-party code blocked the main thread for 0 ms ^

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading.

[Learn more.](#) TBT

Third-Party	Transfer Size	Main-Thread Blocking Time
JSDelivr CDN	29 KiB	0 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	29 KiB	0 ms

○ Lazy load third-party resources with facades ^

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. [Learn more.](#) TBT

● Largest Contentful Paint image was not lazily loaded ^

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. [Learn more.](#)

Element



```
div.leaflet-pane > div.leaflet-layer > div.leaflet-tile-container > img.leaflet-tile

```

● Uses passive listeners to improve scrolling performance ^

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. [Learn more.](#)

● Avoids `document.write()` ^

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. [Learn more.](#)

○ Avoid non-composited animations ^

Animations which are not composited can be janky and increase CLS. [Learn more](#) [CLS]

● Image elements have explicit `width` and `height` ^

Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn more](#) [CLS]

● Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). [TBT]

● Avoids `unload` event listeners ^

The `unload` event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use `pagehide` or `visibilitychange` events instead. [Learn more](#)



Accessibility

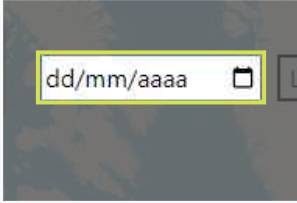
These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

NAMES AND LABELS

▲ Form elements do not have associated labels ^

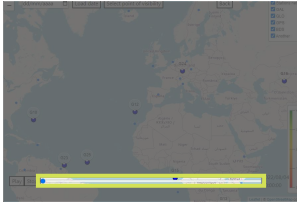
Labels ensure that form controls are announced properly by assistive technologies, like screen readers. [Learn more](#).

Failing Elements



body > div#container > div#top_left_buttons > input#date-input

```
<input id="date-input" type="date">
```



body > div#container > div#botton_buttons > input#fader

```
<input id="fader" type="range" min="0" value="0" step="900" max="85500">
```

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

BEST PRACTICES

- ▲
[user-scalable="no"] is used in the <meta name="viewport"> element or the [maximum-scale] attribute is less than 5.
^

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. [Learn more](#).

Failing Elements

head > meta

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
```

head > meta

```
<meta name="viewport" content="height=device-height, initial-scale=1.0, user-scalable=no">
```

These items highlight common accessibility best practices.

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Hide

- The page has a logical tab order
^

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more](#).

- Interactive controls are keyboard focusable
^

Custom interactive controls are keyboard focusable and display a focus indicator. [Learn more](#).

Interactive elements indicate their purpose and state ^

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn more.](#)

The user's focus is directed to new content added to the page ^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn more.](#)

User focus is not accidentally trapped in a region ^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn more.](#)

Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more.](#)

Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn more.](#)

Visual order on the page follows DOM order ^

DOM order matches the visual order, improving navigation for assistive technology. [Learn more.](#)

Offscreen content is hidden from assistive technology ^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn more.](#)

HTML5 landmark elements are used to improve navigation ^

Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more.](#)

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review.](#)

PASSED AUDITS (17) Hide

[aria-*] attributes match their roles ^

Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. [Learn more](#).

● `[aria-hidden="true"]` is not present on the document `<body>` ^

Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document ``<body>``. [Learn more](#).

● `[role]`s have all required `[aria-*`] attributes ^

Some ARIA roles have required attributes that describe the state of the element to screen readers. [Learn more](#).

● `[role]` values are valid ^

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more](#).

● `[aria-*`] attributes have valid values ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. [Learn more](#).

● `[aria-*`] attributes are valid and not misspelled ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. [Learn more](#).

● Buttons have an accessible name ^

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. [Learn more](#).

● ARIA IDs are unique ^

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. [Learn more](#).

● Image elements have `[alt]` attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

● The page contains a heading, skip link, or landmark region ^

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. [Learn more.](#)

● Background and foreground colors have a sufficient contrast ratio ^

Low-contrast text is difficult or impossible for many users to read. [Learn more.](#)

● Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

● `[id]` attributes on active, focusable elements are unique ^

All focusable elements must have a unique ``id`` to ensure that they're visible to assistive technologies. [Learn more.](#)

● `<html>` element has a `[lang]` attribute ^

If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. [Learn more.](#)

● `<html>` element has a valid value for its `[lang]` attribute ^

Specifying a valid [BCP 47 language](#) helps screen readers announce text properly. [Learn more.](#)

● Links have a discernible name ^

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn more.](#)

No element has a `[tabindex]` value greater than 0 ^

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. [Learn more.](#)

NOT APPLICABLE (25)

Hide

`[accesskey]` values are unique ^

Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. [Learn more.](#)

`button`, `link`, and `menuitem` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

`[aria-hidden="true"]` elements do not contain focusable descendents ^

Focusable descendents within an `[aria-hidden="true"]` element prevent those interactive elements from being available to users of assistive technologies like screen readers. [Learn more.](#)

ARIA input fields have accessible names ^

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `meter` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `progressbar` elements have accessible names ^

When a `progressbar` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

Elements with an ARIA `[role]` that require children to contain a specific `[role]` have all

- required children. ^

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. [Learn more.](#)

- [role]s are contained by their required parent element ^

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more.](#)

- ARIA toggle fields have accessible names ^

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

- ARIA `tooltip` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

- ARIA `treeitem` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

- `<dl>`'s contain only properly-ordered `<dt>` and `<dd>` groups, `<script>`, `<template>` or `<div>` elements. ^

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn more.](#)

- Definition list items are wrapped in `<dl>` elements ^

Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. [Learn more.](#)

- No form fields have multiple labels ^

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn more.](#)

- `<frame>` or `<iframe>` elements have a title ^

Screen reader users rely on frame titles to describe the contents of frames. [Learn more.](#)

- Heading elements appear in a sequentially-descending order ^

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. [Learn more.](#)

- `<input type="image">` elements have `[alt]` text ^

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. [Learn more.](#)

- Lists contain only `` elements and script supporting elements (`<script>` and `<template>`). ^

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. [Learn more.](#)

- List items (``) are contained within `` or `` parent elements ^

Screen readers require list items (``) to be contained within a parent `` or `` to be announced properly. [Learn more.](#)

- The document does not use `<meta http-equiv="refresh">` ^

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. [Learn more.](#)

- `<object>` elements have alternate text ^

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. [Learn more.](#)

- Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table. ^

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. [Learn more.](#)

- `<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe. ^

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. [Learn more.](#)

- `[lang]` attributes have a valid value ^

Specifying a valid [BCP 47 language](#) on elements helps ensure that text is pronounced correctly by a screen reader. [Learn more.](#)

`<video>` elements contain a `<track>` element with `[kind="captions"]` ^

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. [Learn more.](#)



Best Practices

GENERAL

Uses deprecated APIs — 1 warning found ^

Deprecated APIs will eventually be removed from the browser. [Learn more.](#)

Deprecation / Warning	Source
...	/gamedios/map:0:-1 (127.0.0.1)

Detected JavaScript libraries ^

All front-end JavaScript libraries detected on the page. [Learn more.](#)

Name	Version
Leaflet	1.6.0
D3	7.3.0

Missing source maps for large first-party JavaScript ^

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. [Learn more.](#)

URL	Map URL
...dist/d3.js (127.0.0.1)	
Large JavaScript file is missing a source map	
...leaflet/leaflet-src.js (127.0.0.1)	...leaflet/leaflet-src.js.map (127.0.0.1)

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks ^

A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. [Learn more](#)

Description	Directive	Severity
No CSP found in enforcement mode		High

PASSED AUDITS (11)

Hide

● Uses HTTPS ^

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more.](#)

● Avoids requesting the geolocation permission on page load ^

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more.](#)

● Avoids requesting the notification permission on page load ^

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more](#).

● Avoids front-end JavaScript libraries with known security vulnerabilities ^

Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. [Learn more](#).

● Allows users to paste into password fields ^

Preventing password pasting undermines good security policy. [Learn more](#).

● Displays images with correct aspect ratio ^

Image display dimensions should match natural aspect ratio. [Learn more](#).

● Serves images with appropriate resolution ^

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn more](#).

● Page has the HTML doctype ^

Specifying a doctype prevents the browser from switching to quirks-mode. [Learn more](#).

● Properly defines charset ^

A character encoding declaration is required. It can be done with a ``<meta>`` tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. [Learn more](#).

● No browser errors logged to the console ^

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. [Learn more](#)

● No issues in the [Issues](#) panel in Chrome Devtools ^

Issues logged to the `Issues` panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

NOT APPLICABLE (1)

Hide

Fonts with `font-display: optional` are preloaded ^

Preload `optional` fonts so first-time visitors may use them. [Learn more](#)



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more](#).

CONTENT BEST PRACTICES

Document does not have a meta description ^

Meta descriptions may be included in search results to concisely summarize page content. [Learn more](#).

Format your HTML in a way that enables crawlers to better understand your app's content.

ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Hide

Structured data is valid ^

Run the [Structured Data Testing Tool](#) and the [Structured Data Linter](#) to validate structured data. [Learn more.](#)

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (9)

Hide

- Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more.](#) TBT

- Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

- Page has successful HTTP status code ^

Pages with unsuccessful HTTP status codes may not be indexed properly. [Learn more.](#)

- Links have descriptive text ^

Descriptive link text helps search engines understand your content. [Learn more.](#)

- Links are crawlable ^

Search engines may use `href` attributes on links to crawl websites. Ensure that the `href` attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. [Learn More](#)

- Page isn't blocked from indexing ^

Search engines are unable to include your pages in search results if they don't have permission to crawl them. [Learn more.](#)

- Image elements have `[alt]` attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

Document has a valid `hreflang` ^

hreflang links tell search engines what version of a page they should list in search results for a given language or region. [Learn more.](#)

 Document avoids plugins ^

Search engines can't index plugin content, and many devices restrict plugins or don't support them. [Learn more.](#)

NOT APPLICABLE (4)

Hide

 robots.txt is valid ^

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. [Learn more.](#)

 Document has a valid `rel=canonical` ^

Canonical links suggest which URL to show in search results. [Learn more.](#)

 Document uses legible font sizes ^

Font sizes less than 12px are too small to be legible and require mobile visitors to “pinch to zoom” in order to read. Strive to have >60% of page text ≥ 12 px. [Learn more.](#)

 Tap targets are sized appropriately ^

Interactive elements like buttons and links should be large enough (48x48px), and have enough space around them, to be easy enough to tap without overlapping onto other elements. [Learn more.](#)





PWA

These checks validate the aspects of a Progressive Web App. [Learn more.](#)

INSTALLABLE

- Web app manifest or service worker do not meet the installability requirements — 1 reason

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more.](#)

Failure reason

Page has no manifest <link> URL

PWA OPTIMIZED

- Does not register a service worker that controls page and `start_url`

The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. [Learn more.](#)

- Is not configured for a custom splash screen **Failures: No manifest was fetched.**

A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. [Learn more.](#)

- Does not set a theme color for the address bar. **Failures: No manifest was fetched, No `<meta name="theme-color">` tag found.**

The browser address bar can be themed to match your site. [Learn more.](#)

- Content is sized correctly for the viewport

If the width of your app's content doesn't match the width of the viewport, your app might not be optimized for mobile screens. [Learn more.](#)

- Has a `<meta name="viewport">` tag with `width` or `initial-scale`

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

▲ Does not provide a valid `apple-touch-icon`

For ideal appearance on iOS when users add a progressive web app to the home screen, define an `apple-touch-icon`. It must point to a non-transparent 192px (or 180px) square PNG. [Learn More](#).

▲ Manifest doesn't have a maskable icon **No manifest was fetched**

A maskable icon ensures that the image fills the entire shape without being letterboxed when installing the app on a device. [Learn more](#).

ADDITIONAL ITEMS TO MANUALLY CHECK (3)

Hide

○ Site works cross-browser

To reach the most number of users, sites should work across every major browser. [Learn more](#).

○ Page transitions don't feel like they block on the network

Transitions should feel snappy as you tap around, even on a slow network. This experience is key to a user's perception of performance. [Learn more](#).

○ Each page has a URL

Ensure individual pages are deep linkable via URL and that URLs are unique for the purpose of shareability on social media. [Learn more](#).

These checks are required by the baseline [PWA Checklist](#) but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.

📅 Captured at Nov 20, 2022, 7:32

PM GMT+1

🕒 Initial page load

🖥️ Emulated Desktop with Lighthouse

9.6.6

🔧 Custom throttling

🌐 Single page load

🦋 Using Chromium 107.0.0.0 with devtools

A.1.2. Página del menú principal



Performance



Accessibility



Best Practices



SEO



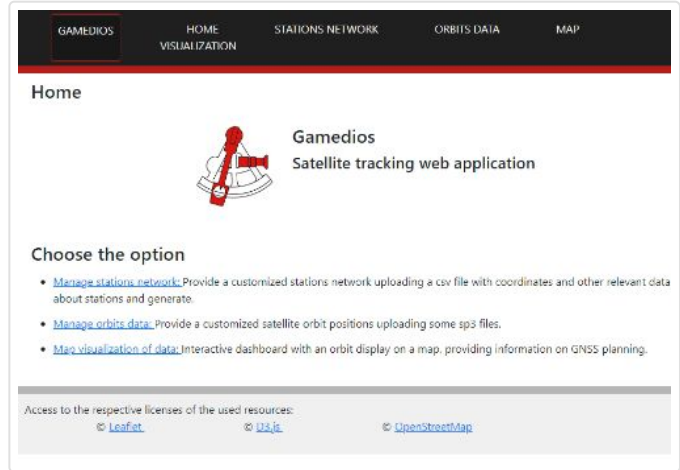
PWA



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0–49 ■ 50–89 ● 90–100



METRICS

[Expand view](#)

● First Contentful Paint
0.8 s

● Speed Index
0.8 s

● Largest Contentful Paint
0.8 s

● Time to Interactive
0.8 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0

[View Treemap](#)



Show audits relevant to: **All** [FCP](#) [TBT](#) [LCP](#) [CLS](#)

OPPORTUNITIES

Opportunity

Estimated Savings

■ Eliminate render-blocking resources

0.64 s ^

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more.](#) FCP LCP

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	27.4 KiB	270 ms
...dist/d3.js (127.0.0.1)	563.8 KiB	650 ms

Reduce unused JavaScript 0.4 s ^

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn more.](#) LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	563.8 KiB	470.7 KiB

Enable text compression 0.36 s ^

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	563.5 KiB	430.9 KiB
/gamedios (127.0.0.1)	2.7 KiB	1.5 KiB

These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

DIAGNOSTICS

Avoid chaining critical requests — 3 chains found ^

The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. [Learn more.](#) FCP LCP

Maximum critical path latency: **150 ms**

Initial Navigation

/gamedios (127.0.0.1)
...dist/d3.js (127.0.0.1) - 120 ms, 563.79 KiB
...css/Main.css (127.0.0.1) - 10 ms, 1.70 KiB
...css/bootstrap.min.css (cdn.jsdelivr.net) - 70 ms, 27.37 KiB

○ Keep request counts low and transfer sizes small — 6 requests • 624 KiB ^

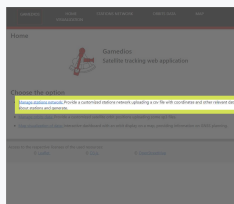
To set budgets for the quantity and size of page resources, add a budget.json file. [Learn more.](#)

Resource Type	Requests	Transfer Size
Total	6	623.8 KiB
Script	1	563.8 KiB
Stylesheet	2	29.1 KiB
Image	1	28.0 KiB
Document	1	2.8 KiB
Other	1	0.2 KiB
Media	0	0.0 KiB
Font	0	0.0 KiB
Third-party	1	27.4 KiB

○ Largest Contentful Paint element — 1 element found ^

This is the largest contentful element painted within the viewport. [Learn More](#) LCP

Element



Manage stations network: Provide a customized stations network uploading a csv ...
[](#)

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (34)

Hide

● Properly size images — Potential savings of 17 KiB ^

Serve images that are appropriately-sized to save cellular data and improve load time. [Learn more.](#)

	URL	Resource Size	Potential Savings
 <pre>div#contentainer > div.row > div.column > img#logo_icon </pre>	...images/icon192.png (127.0.0.1)	27.8 KiB	16.9 KiB

● Defer offscreen images ^

Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. [Learn more.](#)

● Minify CSS ^

Minifying CSS files can reduce network payload sizes. [Learn more.](#) FCP LCP

● Minify JavaScript — Potential savings of 168 KiB ^

Minifying JavaScript files can reduce payload sizes and script parse time. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	563.8 KiB	168.3 KiB

● Reduce unused CSS — Potential savings of 27 KiB ^

Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	27.4 KiB	26.9 KiB

● Efficiently encode images ^

Optimized images load faster and consume less cellular data. [Learn more.](#)

● Serve images in next-gen formats — Potential savings of 22 KiB ^

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn more.](#)

	URL	Resource Size	Potential Savings
 <pre> div#contentainer > div.row > div.column > img#logo_icon </pre>	...images/icon192.png (127.0.0.1)	27.8 KiB	22.0 KiB

● Preconnect to required origins ^

Consider adding `preconnect` or `dns-prefetch` resource hints to establish early connections to important third-party origins. [Learn more.](#) FCP LCP

● Initial server response time was short — Root document took 20 ms ^

Keep the server response time for the main document short because all other requests depend on it. [Learn more.](#) FCP LCP

URL	Time Spent
/gamedios (127.0.0.1)	20 ms

● Avoid multiple page redirects ^

Redirects introduce additional delays before the page can be loaded. [Learn more.](#) FCP LCP

Preload key requests ^

Consider using ``<link rel=preload>`` to prioritize fetching resources that are currently requested later in page load. [Learn more.](#) FCP LCP

Use HTTP/2 ^

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. [Learn more.](#)

Use video formats for animated content ^

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. [Learn more](#) LCP

Remove duplicate modules in JavaScript bundles ^

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. TBT

Avoid serving legacy JavaScript to modern browsers ^

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. [Learn More](#) TBT

Preload Largest Contentful Paint image ^

Preload the image used by the LCP element in order to improve your LCP time. [Learn more.](#) LCP

Avoids enormous network payloads — Total size was 624 KiB ^

Large network payloads cost users real money and are highly correlated with long load times.

[Learn more.](#) LCP

Show 3rd-party resources (1)


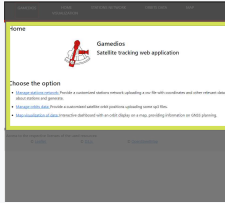
URL	Transfer Size
...dist/d3.js (127.0.0.1)	563.8 KiB
...images/icon192.png (127.0.0.1)	28.0 KiB
...css/bootstrap.min.css (cdn.jsdelivr.net)	27.4 KiB
/gamedios (127.0.0.1)	2.8 KiB
...css/Main.css (127.0.0.1)	1.7 KiB
...images/icon192.png (127.0.0.1)	0.2 KiB

● Uses efficient cache policy on static assets — 0 resources found ^

A long cache lifetime can speed up repeat visits to your page. [Learn more.](#)

● Avoids an excessive DOM size — 36 elements ^

A large DOM will increase memory usage, cause longer [style calculations](#), and produce costly [layout reflows](#). [Learn more.](#) TBT

Statistic	Element	Value
Total DOM Elements		36
Maximum DOM Depth	 <p>Leaflet </p>	6
Maximum Child Elements	 <p>Home Gamedios Satellite tracking web application Choose the option Manage ... <div id="container"></p>	7

○ User Timing marks and measures ^

Consider instrumenting your app with the User Timing API to measure your app's real-world

performance during key user experiences. [Learn more.](#)

● JavaScript execution time — 0.0 s

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

URL	Total CPU Time	Script Evaluation	Script Parse
/gamedios (127.0.0.1)	62 ms	2 ms	1 ms

● Minimizes main-thread work — 0.1 s

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more](#) TBT

Category	Time Spent
Style & Layout	36 ms
Other	26 ms
Rendering	11 ms
Script Parsing & Compilation	10 ms
Script Evaluation	9 ms
Parse HTML & CSS	4 ms

● All text remains visible during webfont loads

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading. [Learn more.](#) FCP LCP

● Minimize third-party usage — Third-party code blocked the main thread for 0 ms

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading. [Learn more.](#) TBT

Third-Party	Transfer Size	Main-Thread Blocking Time
JSDelivr CDN	27 KiB	0 ms

Third-Party	Transfer Size	Main-Thread Blocking Time
...css/bootstrap.min.css (cdn.jsdelivr.net)	27 KiB	0 ms

Lazy load third-party resources with facades ^

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. [Learn more](#). TBT

Largest Contentful Paint image was not lazily loaded ^

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. [Learn more](#).

Avoid large layout shifts ^

These DOM elements contribute most to the CLS of the page. CLS

Uses passive listeners to improve scrolling performance ^

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. [Learn more](#).

Avoids `document.write()` ^

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. [Learn more](#).

Avoid long main-thread tasks ^

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay. [Learn more](#) TBT

Avoid non-composited animations ^

Animations which are not composited can be janky and increase CLS. [Learn more](#) CLS

Image elements have explicit `width` and `height` ^

Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn more](#) CLS

- Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

- Avoids `unload` event listeners ^

The `unload` event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use `pagehide` or `visibilitychange` events instead. [Learn more](#)



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

NAMES AND LABELS

- ▲ Image elements do not have `[alt]` attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more](#).

Failing Elements



```
div#container > div.row > div.column > img#logo_icon  

```

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

BEST PRACTICES

▲ [user-scalable="no"] is used in the <meta name="viewport"> element or the [maximum-scale] attribute is less than 5. ^

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. [Learn more.](#)

Failing Elements

```
head > meta
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
```

```
head > meta
<meta name="viewport" content="height=device-height, initial-scale=1.0, user-scalable=no">
```


These items highlight common accessibility best practices.

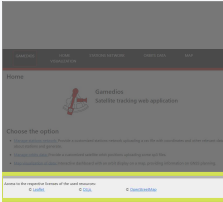
CONTRAST


▲ Background and foreground colors do not have a sufficient contrast ratio. ^

Low-contrast text is difficult or impossible for many users to read. [Learn more.](#)

Failing Elements

Leaflet


Access to the respective licenses of the used resources: © Leaflet © D3.js © Op...

<footer id="license_text">

D3.js


Access to the respective licenses of the used resources: © Leaflet © D3.js © Op...

	<pre><footer id="license_text"></pre>
	<p>OpenStreetMap</p> <pre></pre>
	<p>Access to the respective licenses of the used resources: © Leaflet © D3.js © Op...</p> <pre><footer id="license_text"></pre>

These are opportunities to improve the legibility of your content.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

<input type="radio"/> The page has a logical tab order	^
<p>Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. Learn more.</p>	
<input type="radio"/> Interactive controls are keyboard focusable	^
<p>Custom interactive controls are keyboard focusable and display a focus indicator. Learn more.</p>	
<input type="radio"/> Interactive elements indicate their purpose and state	^
<p>Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. Learn more.</p>	
<input type="radio"/> The user's focus is directed to new content added to the page	^
<p>If new content, such as a dialog, is added to the page, the user's focus is directed to it. Learn more.</p>	
<input type="radio"/> User focus is not accidentally trapped in a region	^
<p>A user can tab into and out of any control or region without accidentally trapping their focus. Learn more.</p>	
<input type="radio"/> Custom controls have associated labels	^

<p>Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. Learn more.</p>
<p><input type="radio"/> Custom controls have ARIA roles ^</p>
<p>Custom interactive controls have appropriate ARIA roles. Learn more.</p>
<p><input type="radio"/> Visual order on the page follows DOM order ^</p>
<p>DOM order matches the visual order, improving navigation for assistive technology. Learn more.</p>
<p><input type="radio"/> Offscreen content is hidden from assistive technology ^</p>
<p>Offscreen content is hidden with display: none or aria-hidden=true. Learn more.</p>
<p><input type="radio"/> HTML5 landmark elements are used to improve navigation ^</p>
<p>Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. Learn more.</p>

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review.](#)

PASSED AUDITS (9)

Hide

<p><input checked="" type="radio"/> [aria-hidden="true"] is not present on the document <body> ^</p>
<p>Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document `<code><body></code>`. Learn more.</p>
<p><input checked="" type="radio"/> The page contains a heading, skip link, or landmark region ^</p>
<p>Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. Learn more.</p>
<p><input checked="" type="radio"/> Document has a <title> element ^</p>
<p>The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. Learn more.</p>

● `<html>` element has a `[lang]` attribute ^

If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. [Learn more](#).

● `<html>` element has a valid value for its `[lang]` attribute ^

Specifying a valid [BCP 47 language](#) helps screen readers announce text properly. [Learn more](#).

● Links have a discernible name ^

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn more](#).

● Lists contain only `` elements and script supporting elements (`<script>` and `<template>`). ^

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. [Learn more](#).

● List items (``) are contained within `` or `` parent elements ^

Screen readers require list items (``) to be contained within a parent `` or `` to be announced properly. [Learn more](#).

● Heading elements appear in a sequentially-descending order ^

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. [Learn more](#).

NOT APPLICABLE (32)

Hide

○ `[accesskey]` values are unique ^

Access keys let users quickly focus a part of the page. For proper navigation, each access key

must be unique. Learn more.	
<input type="radio"/> [aria-*] attributes match their roles	^
Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. Learn more.	
<input type="radio"/> <code>button</code> , <code>link</code> , and <code>menuitem</code> elements have accessible names	^
When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more.	
<input type="radio"/> [aria-hidden="true"] elements do not contain focusable descendents	^
Focusable descendents within an `[aria-hidden="true"]` element prevent those interactive elements from being available to users of assistive technologies like screen readers. Learn more.	
<input type="radio"/> ARIA input fields have accessible names	^
When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more.	
<input type="radio"/> ARIA <code>meter</code> elements have accessible names	^
When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more.	
<input type="radio"/> ARIA <code>progressbar</code> elements have accessible names	^
When a `progressbar` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more.	
<input type="radio"/> [role]s have all required [aria-*] attributes	^
Some ARIA roles have required attributes that describe the state of the element to screen readers. Learn more.	
<input type="radio"/> Elements with an ARIA [role] that require children to contain a specific [role] have all required children.	^
Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. Learn more.	
<input type="radio"/> [role]s are contained by their required parent element	^

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more.](#)

[role] values are valid ^

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more.](#)

ARIA toggle fields have accessible names ^

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `tooltip` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `treeitem` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

[aria-*] attributes have valid values ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. [Learn more.](#)

[aria-*] attributes are valid and not misspelled ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. [Learn more.](#)

Buttons have an accessible name ^

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. [Learn more.](#)

`<dl>`'s contain only properly-ordered `<dt>` and `<dd>` groups, `<script>`, `<template>` or `<div>` elements. ^

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn more.](#)

Definition list items are wrapped in `<dl>` elements ^

Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. [Learn more.](#)

`[id]` attributes on active, focusable elements are unique ^

All focusable elements must have a unique `id` to ensure that they're visible to assistive technologies. [Learn more.](#)

ARIA IDs are unique ^

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. [Learn more.](#)

No form fields have multiple labels ^

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn more.](#)

`<frame>` or `<iframe>` elements have a title ^

Screen reader users rely on frame titles to describe the contents of frames. [Learn more.](#)

`<input type="image">` elements have `[alt]` text ^

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. [Learn more.](#)

Form elements have associated labels ^

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. [Learn more.](#)

The document does not use `<meta http-equiv="refresh">` ^

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. [Learn more.](#)

`<object>` elements have alternate text ^

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. [Learn more.](#)

<input type="radio"/> No element has a <code>[tabindex]</code> value greater than 0 ^
A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. Learn more.
<input type="radio"/> Cells in a <code><table></code> element that use the <code>[headers]</code> attribute refer to table cells within the same table. ^
Screen readers have features to make navigating tables easier. Ensuring <code><td></code> cells using the <code>[headers]</code> attribute only refer to other cells in the same table may improve the experience for screen reader users. Learn more.
<input type="radio"/> <code><th></code> elements and elements with <code>[role="columnheader"/"rowheader"]</code> have data cells they describe. ^
Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. Learn more.
<input type="radio"/> <code>[lang]</code> attributes have a valid value ^
Specifying a valid BCP 47 language on elements helps ensure that text is pronounced correctly by a screen reader. Learn more.
<input type="radio"/> <code><video></code> elements contain a <code><track></code> element with <code>[kind="captions"]</code> ^
When a video provides a caption it is easier for deaf and hearing impaired users to access its information. Learn more.



Best Practices

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks ^

A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. [Learn more](#)

Description	Directive	Severity
No CSP found in enforcement mode		High

GENERAL

○ Detected JavaScript libraries ^

All front-end JavaScript libraries detected on the page. [Learn more](#).

Name	Version
D3	7.3.0

▲ Missing source maps for large first-party JavaScript ^

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. [Learn more](#).

URL	Map URL
...dist/d3.js (127.0.0.1)	
Large JavaScript file is missing a source map	

PASSED AUDITS (12) Hide

● Uses HTTPS ^

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more](#).

● Avoids requesting the geolocation permission on page load ^

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more](#).

● Avoids requesting the notification permission on page load ^

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more](#).

● Avoids front-end JavaScript libraries with known security vulnerabilities ^

Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. [Learn more](#).

● Allows users to paste into password fields ^

Preventing password pasting undermines good security policy. [Learn more](#).

● Displays images with correct aspect ratio ^

Image display dimensions should match natural aspect ratio. [Learn more](#).

● Serves images with appropriate resolution ^

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn more](#).

● Page has the HTML doctype ^

Specifying a doctype prevents the browser from switching to quirks-mode. [Learn more](#).

● Properly defines charset ^

A character encoding declaration is required. It can be done with a `<meta>` tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. [Learn more](#).

● Avoids deprecated APIs ^

Deprecated APIs will eventually be removed from the browser. [Learn more](#).

● No browser errors logged to the console ^

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. [Learn more](#)

● No issues in the [Issues](#) panel in Chrome Devtools ^

Issues logged to the `Issues` panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

NOT APPLICABLE (1)

Hide

○ Fonts with `font-display: optional` are preloaded ^

Preload `optional` fonts so first-time visitors may use them. [Learn more](#)



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more](#).

CONTENT BEST PRACTICES

▲ Document does not have a meta description ^

Meta descriptions may be included in search results to concisely summarize page content. [Learn more.](#)

▲ Image elements do not have [alt] attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

Failing Elements

	<pre>div#container > div.row > div.column > img#logo_icon </pre>
-----------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Format your HTML in a way that enables crawlers to better understand your app's content.

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Hide

Structured data is valid ^

Run the [Structured Data Testing Tool](#) and the [Structured Data Linter](#) to validate structured data. [Learn more.](#)

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (8) Hide

Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more.](#) TBT

Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

Page has successful HTTP status code ^

Pages with unsuccessful HTTP status codes may not be indexed properly. [Learn more.](#)

Links have descriptive text ^

Descriptive link text helps search engines understand your content. [Learn more](#).

Links are crawlable ^

Search engines may use `href` attributes on links to crawl websites. Ensure that the `href` attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. [Learn More](#)

Page isn't blocked from indexing ^

Search engines are unable to include your pages in search results if they don't have permission to crawl them. [Learn more](#).

Document has a valid `hreflang` ^

hreflang links tell search engines what version of a page they should list in search results for a given language or region. [Learn more](#).

Document avoids plugins ^

Search engines can't index plugin content, and many devices restrict plugins or don't support them. [Learn more](#).

NOT APPLICABLE (4)

Hide

robots.txt is valid ^

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. [Learn more](#).

Document has a valid `rel=canonical` ^

Canonical links suggest which URL to show in search results. [Learn more](#).

Document uses legible font sizes ^

Font sizes less than 12px are too small to be legible and require mobile visitors to “pinch to zoom”

in order to read. Strive to have >60% of page text ≥ 12 px. [Learn more.](#)

○ Tap targets are sized appropriately ^

Interactive elements like buttons and links should be large enough (48x48px), and have enough space around them, to be easy enough to tap without overlapping onto other elements. [Learn more.](#)



PWA

These checks validate the aspects of a Progressive Web App. [Learn more.](#)

⊕ INSTALLABLE

▲ Web app manifest or service worker do not meet the installability requirements — 1 reason ^

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more.](#)

Failure reason

Page has no manifest <link> URL

★ PWA OPTIMIZED

▲ Does not register a service worker that controls page and `start_url` ^

The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. [Learn more.](#)

- ▲ Is not configured for a custom splash screen **Failures: No manifest was fetched.** ^

A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. [Learn more.](#)

- ▲ Does not set a theme color for the address bar. **Failures: No manifest was fetched, No ``<meta name="theme-color">`` tag found.** ^

The browser address bar can be themed to match your site. [Learn more.](#)

- Content is sized correctly for the viewport ^

If the width of your app's content doesn't match the width of the viewport, your app might not be optimized for mobile screens. [Learn more.](#)

- Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input.](#) [Learn more.](#) TBT

- ▲ Does not provide a valid `apple-touch-icon` ^

For ideal appearance on iOS when users add a progressive web app to the home screen, define an `apple-touch-icon`. It must point to a non-transparent 192px (or 180px) square PNG. [Learn More.](#)

- ▲ Manifest doesn't have a maskable icon **No manifest was fetched** ^

A maskable icon ensures that the image fills the entire shape without being letterboxed when installing the app on a device. [Learn more.](#)

ADDITIONAL ITEMS TO MANUALLY CHECK (3) Hide

- Site works cross-browser ^

To reach the most number of users, sites should work across every major browser. [Learn more.](#)

- Page transitions don't feel like they block on the network ^

Transitions should feel snappy as you tap around, even on a slow network. This experience is key to a user's perception of performance. [Learn more.](#)


- Each page has a URL ^

Ensure individual pages are deep linkable via URL and that URLs are unique for the purpose of shareability on social media. [Learn more](#).

These checks are required by the baseline [PWA Checklist](#) but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.

 Captured at Dec 27, 2022, 1:03

PM GMT+1


 Initial page load

 Emulated Desktop with Lighthouse

9.6.6

 Custom throttling

 Single page load

 Using Chromium 108.0.0.0 with devtools

Generated by **Lighthouse 9.6.6** | [File an issue](#)

A.1.3. Página de gestión de la red de estaciones



Performance



Accessibility



Best Practices



SEO



PWA



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

- ▲ 0–49
- 50–89
- 90–100

Option: Stations data

Load the station network .csv:

Uploaded file should meet the .csv format and should have at least the following columns [#StationName, 'Latitude', 'Longitude', 'ReceiverName', 'ReceiverSatelliteSystem'] with unique '#StationName', and 'Latitude', 'Longitude' expressed in degrees and using ',' separator.

Seleccionar archivo: Ninguno archivo seleccionado Enviar

Station .csv file loaded successfully

Heatmap data for the stations network available

#StationName	Latitude	Longitude	ReceiverName	ReceiverSatelliteSystem
0 ALIC00AUS	-23.67	133.896	SEPT POLARX5	GPS+QZSS+GLO+GAL+BDS
1 ROGT00COL	4.64	-74.081	JAVAD TRE-3 DELTA	GPS+GLO+GAL+BDS
2 CEDU00AUS	-31.067	133.81	SEPT POLARX5	GPS+GLO+GAL+BDS+QZSS+IRNSS
3 CKIS00COK	-21.201	-158.001	TRIMBLE NETR9	GPS+GLO+GAL+BDS+QZSS
4 DAV100AIA	-68.577	77.973	SEPT POLARX5	GPS+GLO+GAL+BDS+IRNSS+QZSS
5

METRICS

Expand view

● First Contentful Paint
0.9 s

● Time to Interactive
0.9 s

● Speed Index
0.9 s

● Total Blocking Time
0 ms

● Largest Contentful Paint
0.9 s

● Cumulative Layout Shift
0

[View Treemap](#)



Show audits relevant to: **All** [FCP](#) [TBT](#) [LCP](#) [CLS](#)

OPPORTUNITIES

Opportunity

Estimated Savings

■ Eliminate render-blocking resources

0.66 s ^

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more.](#) FCP LCP

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	300 ms
...dist/d3.js (127.0.0.1)	563.8 KiB	650 ms

Reduce unused JavaScript 0.44 s ^

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn more.](#) LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	563.8 KiB	470.7 KiB

Enable text compression 0.4 s ^

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	563.5 KiB	430.9 KiB
/gamedios/parse_stations (127.0.0.1)	11.6 KiB	8.8 KiB

Minify JavaScript 0.16 s ^

Minifying JavaScript files can reduce payload sizes and script parse time. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	563.8 KiB	168.3 KiB

These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

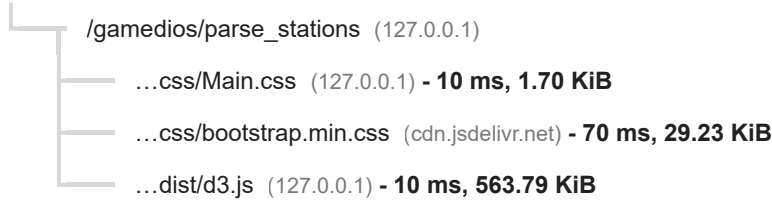
DIAGNOSTICS

○ Avoid chaining critical requests — 3 chains found

The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. [Learn more.](#) FCP LCP

Maximum critical path latency: **100 ms**

Initial Navigation



○ Keep request counts low and transfer sizes small — 5 requests • 634 KiB

To set budgets for the quantity and size of page resources, add a budget.json file. [Learn more.](#)

Resource Type	Requests	Transfer Size
Total	5	634.4 KiB
Script	1	563.8 KiB
Stylesheet	2	30.9 KiB
Other	1	28.0 KiB
Document	1	11.7 KiB
Image	0	0.0 KiB
Media	0	0.0 KiB
Font	0	0.0 KiB
Third-party	1	29.2 KiB

○ Largest Contentful Paint element — 1 element found

This is the largest contentful element painted within the viewport. [Learn More](#) LCP

Element

Uploaded file should meet the .csv format and should have at least the followin...
<p>

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (33)

Hide

● Properly size images ^

Serve images that are appropriately-sized to save cellular data and improve load time. [Learn more.](#)

● Defer offscreen images ^

Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. [Learn more.](#)

● Minify CSS ^

Minifying CSS files can reduce network payload sizes. [Learn more.](#) FCP LCP

● Reduce unused CSS — Potential savings of 29 KiB ^

Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	28.7 KiB

● Efficiently encode images ^

Optimized images load faster and consume less cellular data. [Learn more.](#)

● Serve images in next-gen formats ^

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn more.](#)

● Preconnect to required origins ^

Consider adding `preconnect` or `dns-prefetch` resource hints to establish early connections to important third-party origins. [Learn more.](#) FCP LCP

● Initial server response time was short — Root document took 20 ms ^

Keep the server response time for the main document short because all other requests depend on it. [Learn more.](#) FCP LCP

URL	Time Spent
/gamedios/parse_stations (127.0.0.1)	20 ms

● Avoid multiple page redirects ^

Redirects introduce additional delays before the page can be loaded. [Learn more.](#) FCP LCP

○ Preload key requests ^

Consider using ``<link rel=preload>`` to prioritize fetching resources that are currently requested later in page load. [Learn more.](#) FCP LCP

● Use HTTP/2 ^

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. [Learn more.](#)

● Use video formats for animated content ^

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. [Learn more](#)

LCP

● Remove duplicate modules in JavaScript bundles ^

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. TBT

● Avoid serving legacy JavaScript to modern browsers ^

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. [Learn More](#) TBT

● Preload Largest Contentful Paint image ^

Preload the image used by the LCP element in order to improve your LCP time. [Learn more.](#) LCP

● Avoids enormous network payloads — Total size was 634 KiB ^

Large network payloads cost users real money and are highly correlated with long load times. [Learn more.](#) LCP

Show 3rd-party resources (1)

URL	Transfer Size
...dist/d3.js (127.0.0.1)	563.8 KiB
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB
...images/icon192.png (127.0.0.1)	28.0 KiB
/gamedios/parse_stations (127.0.0.1)	11.7 KiB
...css/Main.css (127.0.0.1)	1.7 KiB

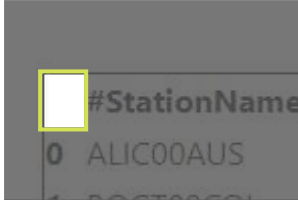

● Uses efficient cache policy on static assets — 0 resources found ^

A long cache lifetime can speed up repeat visits to your page. [Learn more.](#)

● Avoids an excessive DOM size — 447 elements ^

A large DOM will increase memory usage, cause longer [style calculations](#), and produce costly [layout reflows](#). [Learn more.](#) TBT

Statistic	Element	Value
Total DOM Elements		447

Statistic	Element	Value
Maximum DOM Depth	 <pre>table.dataframe > thead > tr > th <th></pre>	7
Maximum Child Elements	 <pre>0 ALIC00AUS -23.67 133.886 SEPT POLARX5 GPS+QZSS+GLO+GAL+BDS 1 BOGT00COL 4.64 -... <tbody></pre>	58

User Timing marks and measures ^

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. [Learn more.](#)

JavaScript execution time — 0.0 s ^

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

URL	Total CPU Time	Script Evaluation	Script Parse
/gamedios/parse_stations (127.0.0.1)	101 ms	3 ms	1 ms

Minimizes main-thread work — 0.2 s ^

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

Category	Time Spent
Style & Layout	74 ms
Other	51 ms
Script Parsing & Compilation	13 ms
Script Evaluation	12 ms
Rendering	10 ms
Parse HTML & CSS	7 ms

● All text remains visible during webfont loads ^

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading.

[Learn more.](#) FCP LCP

● Minimize third-party usage — Third-party code blocked the main thread for 0 ms ^

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading.

[Learn more.](#) TBT

Third-Party	Transfer Size	Main-Thread Blocking Time
JSDelivr CDN	29 KiB	0 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	29 KiB	0 ms

○ Lazy load third-party resources with facades ^

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. [Learn more.](#) TBT

○ Largest Contentful Paint image was not lazily loaded ^

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. [Learn more.](#)

○ Avoid large layout shifts ^

These DOM elements contribute most to the CLS of the page. CLS

● Uses passive listeners to improve scrolling performance ^

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. [Learn more.](#)

● Avoids `document.write()` ^

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. [Learn more.](#)

○ Avoid long main-thread tasks ^

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay.

[Learn more](#) TBT

Avoid non-composited animations ^

Animations which are not composited can be janky and increase CLS. [Learn more](#) CLS

Image elements have explicit `width` and `height` ^

Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn more](#) CLS

Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

Avoids `unload` event listeners ^

The `unload` event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use `pagehide` or `visibilitychange` events instead. [Learn more](#)



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

NAMES AND LABELS

Form elements do not have associated labels ^

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. [Learn more.](#)

Failing Elements



body > div#container > form > input#uploadFile

```
<input type="file" id="uploadFile" accept=".csv">
```

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

BEST PRACTICES



[\[user-scalable="no"\]](#) is used in the `<meta name="viewport">` element or the [\[maximum-scale\]](#) attribute is less than 5. ^

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. [Learn more.](#)

Failing Elements

head > meta

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
```

head > meta

```
<meta name="viewport" content="height=device-height, initial-scale=1.0, user-scalable=no">
```

These items highlight common accessibility best practices.

CONTRAST



Background and foreground colors do not have a sufficient contrast ratio. ^

Low-contrast text is difficult or impossible for many users to read. [Learn more.](#)

Failing Elements

Leaflet

```
<a href="https://github.com/Leaflet/Leaflet/blob/main/LICENSE">
```

Access to the respective licenses of the used resources: © Leaflet © D3.js © Op...

```
<footer id="license_text">
```

Failing Elements

D3.js

```
<a href="https://github.com/d3/d3/blob/main/LICENSE">
```

Access to the respective licenses of the used resources: © Leaflet © D3.js © Op...

```
<footer id="license_text">
```

OpenStreetMap

```
<a href="https://www.openstreetmap.org/copyright">
```

Access to the respective licenses of the used resources: © Leaflet © D3.js © Op...

```
<footer id="license_text">
```

These are opportunities to improve the legibility of your content.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

- The page has a logical tab order ^

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more.](#)

- Interactive controls are keyboard focusable ^

Custom interactive controls are keyboard focusable and display a focus indicator. [Learn more.](#)

- Interactive elements indicate their purpose and state ^

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn more.](#)

- The user's focus is directed to new content added to the page ^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn more.](#)

- User focus is not accidentally trapped in a region ^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn more.](#)

- Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more.](#)

○ Custom controls have ARIA roles



Custom interactive controls have appropriate ARIA roles. [Learn more.](#)

○ Visual order on the page follows DOM order



DOM order matches the visual order, improving navigation for assistive technology. [Learn more.](#)

○ Offscreen content is hidden from assistive technology



Offscreen content is hidden with display: none or aria-hidden=true. [Learn more.](#)

○ HTML5 landmark elements are used to improve navigation



Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more.](#)

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review.](#)

PASSED AUDITS (12)

Hide

● [aria-hidden="true"] is not present on the document <body>



Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document ``<body>``. [Learn more.](#)

● The page contains a heading, skip link, or landmark region



Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. [Learn more.](#)

● Document has a <title> element



The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

● [id] attributes on active, focusable elements are unique



All focusable elements must have a unique `id` to ensure that they're visible to assistive technologies. [Learn more.](#)

● `<html>` element has a `[lang]` attribute ^

If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. [Learn more.](#)

● `<html>` element has a valid value for its `[lang]` attribute ^

Specifying a valid [BCP 47 language](#) helps screen readers announce text properly. [Learn more.](#)

● Links have a discernible name ^

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn more.](#)

● Lists contain only `` elements and script supporting elements (`<script>` and `<template>`). ^

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. [Learn more.](#)

● List items (``) are contained within `` or `` parent elements ^

Screen readers require list items (``) to be contained within a parent `` or `` to be announced properly. [Learn more.](#)

● Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table. ^

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. [Learn more.](#)

● `<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe. ^

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. [Learn more.](#)

Heading elements appear in a sequentially-descending order ^

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. [Learn more.](#)

NOT APPLICABLE (29)

Hide

[\[accesskey\]](#) values are unique ^

Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. [Learn more.](#)

[\[aria-*\]](#) attributes match their roles ^

Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. [Learn more.](#)

[button](#), [link](#), and [menuitem](#) elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

[\[aria-hidden="true"\]](#) elements do not contain focusable descendents ^

Focusable descendents within an `[aria-hidden="true"]` element prevent those interactive elements from being available to users of assistive technologies like screen readers. [Learn more.](#)

ARIA input fields have accessible names ^

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA [meter](#) elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA [progressbar](#) elements have accessible names ^

When a `progressbar` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

[role]s have all required [aria-*] attributes ^

Some ARIA roles have required attributes that describe the state of the element to screen readers. [Learn more.](#)

Elements with an ARIA [role] that require children to contain a specific [role] have all required children. ^

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. [Learn more.](#)

[role]s are contained by their required parent element ^

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more.](#)

[role] values are valid ^

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more.](#)

ARIA toggle fields have accessible names ^

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA tooltip elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA treeitem elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

[aria-*] attributes have valid values ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. [Learn more.](#)

[aria-*] attributes are valid and not misspelled ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. [Learn more.](#)

Buttons have an accessible name ^

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. [Learn more.](#)

<dl>'s contain only properly-ordered <dt> and <dd> groups, <script>, <template> or <div> elements. ^

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn more.](#)

Definition list items are wrapped in <dl> elements ^

Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. [Learn more.](#)

ARIA IDs are unique ^

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. [Learn more.](#)

No form fields have multiple labels ^

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn more.](#)

<frame> or <iframe> elements have a title ^

Screen reader users rely on frame titles to describe the contents of frames. [Learn more.](#)

Image elements have [alt] attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

<input type="image"> elements have [alt] text ^

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. [Learn more.](#)

The document does not use `<meta http-equiv="refresh">` ^

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. [Learn more.](#)

`<object>` elements have alternate text ^

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. [Learn more.](#)

No element has a `[tabindex]` value greater than 0 ^

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. [Learn more.](#)

`[lang]` attributes have a valid value ^

Specifying a valid [BCP 47 language](#) on elements helps ensure that text is pronounced correctly by a screen reader. [Learn more.](#)

`<video>` elements contain a `<track>` element with `[kind="captions"]` ^

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. [Learn more.](#)



Best Practices

TRUST AND SAFETY

Ensure CSP is effective against XSS attacks ^

A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. [Learn more](#)

Description	Directive	Severity
No CSP found in enforcement mode		High

GENERAL

○ Detected JavaScript libraries ^

All front-end JavaScript libraries detected on the page. [Learn more.](#)

Name	Version
D3	7.3.0

▲ Missing source maps for large first-party JavaScript ^

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. [Learn more.](#)

URL	Map URL
...dist/d3.js (127.0.0.1)	
Large JavaScript file is missing a source map	

PASSED AUDITS (12) Hide

● Uses HTTPS ^

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more.](#)

● Avoids requesting the geolocation permission on page load ^

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more.](#)

● Avoids requesting the notification permission on page load ^

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more](#).

● Avoids front-end JavaScript libraries with known security vulnerabilities ^

Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. [Learn more](#).

● Allows users to paste into password fields ^

Preventing password pasting undermines good security policy. [Learn more](#).

● Displays images with correct aspect ratio ^

Image display dimensions should match natural aspect ratio. [Learn more](#).

● Serves images with appropriate resolution ^

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn more](#).

● Page has the HTML doctype ^

Specifying a doctype prevents the browser from switching to quirks-mode. [Learn more](#).

● Properly defines charset ^

A character encoding declaration is required. It can be done with a `` tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. [Learn more](#).

● Avoids deprecated APIs ^

Deprecated APIs will eventually be removed from the browser. [Learn more](#).

● No browser errors logged to the console ^

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. [Learn more](#)

● No issues in the [Issues](#) panel in Chrome Devtools ^

Issues logged to the `Issues` panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

NOT APPLICABLE (1) Hide

○ Fonts with `font-display: optional` are preloaded ^

Preload `optional` fonts so first-time visitors may use them. [Learn more](#)



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more](#).

CONTENT BEST PRACTICES

▲ Document does not have a meta description ^

Meta descriptions may be included in search results to concisely summarize page content. [Learn more](#).

Format your HTML in a way that enables crawlers to better understand your app's content.

ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Hide

Structured data is valid ^

Run the [Structured Data Testing Tool](#) and the [Structured Data Linter](#) to validate structured data. [Learn more.](#)

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (8)

Hide

Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more.](#) TBT

Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

Page has successful HTTP status code ^

Pages with unsuccessful HTTP status codes may not be indexed properly. [Learn more.](#)

Links have descriptive text ^

Descriptive link text helps search engines understand your content. [Learn more.](#)

Links are crawlable ^

Search engines may use `href` attributes on links to crawl websites. Ensure that the `href` attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. [Learn More](#)

Page isn't blocked from indexing ^

Search engines are unable to include your pages in search results if they don't have permission to crawl them. [Learn more.](#)

Document has a valid `hreflang` ^

hreflang links tell search engines what version of a page they should list in search results for a given language or region. [Learn more.](#)

Document avoids plugins ^

Search engines can't index plugin content, and many devices restrict plugins or don't support them. [Learn more.](#)

NOT APPLICABLE (5) Hide

robots.txt is valid ^

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. [Learn more.](#)

Image elements have `[alt]` attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

Document has a valid `rel=canonical` ^

Canonical links suggest which URL to show in search results. [Learn more.](#)

Document uses legible font sizes ^

Font sizes less than 12px are too small to be legible and require mobile visitors to “pinch to zoom” in order to read. Strive to have >60% of page text ≥12px. [Learn more.](#)

Tap targets are sized appropriately ^

Interactive elements like buttons and links should be large enough (48x48px), and have enough space around them, to be easy enough to tap without overlapping onto other elements. [Learn more.](#)



PWA

These checks validate the aspects of a Progressive Web App. [Learn more.](#)

INSTALLABLE

-  Web app manifest or service worker do not meet the installability requirements — 1 reason 

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more.](#)

Failure reason

Page has no manifest <link> URL



PWA OPTIMIZED

-  Does not register a service worker that controls page and `start_url` 

The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. [Learn more.](#)

-  Is not configured for a custom splash screen **Failures: No manifest was fetched.** 

A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. [Learn more.](#)

-  Does not set a theme color for the address bar. **Failures: No manifest was fetched, No ``<meta name="theme-color">`` tag found.** 

The browser address bar can be themed to match your site. [Learn more.](#)

-  Content is sized correctly for the viewport 

If the width of your app's content doesn't match the width of the viewport, your app might not be optimized for mobile screens. [Learn more.](#)

● Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more.](#) TBT

▲ Does not provide a valid `apple-touch-icon` ^

For ideal appearance on iOS when users add a progressive web app to the home screen, define an `apple-touch-icon`. It must point to a non-transparent 192px (or 180px) square PNG. [Learn More.](#)

▲ Manifest doesn't have a maskable icon **No manifest was fetched** ^

A maskable icon ensures that the image fills the entire shape without being letterboxed when installing the app on a device. [Learn more.](#)

ADDITIONAL ITEMS TO MANUALLY CHECK (3)

Hide

Site works cross-browser ^

To reach the most number of users, sites should work across every major browser. [Learn more.](#)

Page transitions don't feel like they block on the network ^

Transitions should feel snappy as you tap around, even on a slow network. This experience is key to a user's perception of performance. [Learn more.](#)

Each page has a URL ^

Ensure individual pages are deep linkable via URL and that URLs are unique for the purpose of shareability on social media. [Learn more.](#)

These checks are required by the baseline [PWA Checklist](#) but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.

🕒 Initial page load

🔧 Custom throttling

🖥️ Using Chromium 107.0.0.0 with devtools

Generated by **Lighthouse** 9.6.6 | [File an issue](#)

A.1.4. Página de gestión de los datos orbitales



Performance



Accessibility



Best Practices



SEO



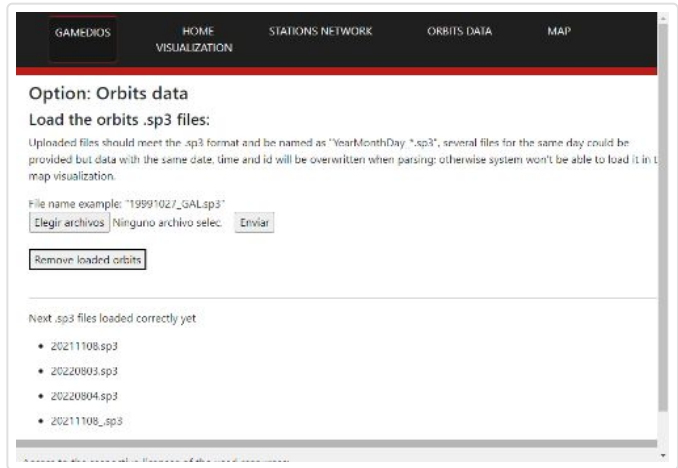
PWA



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0-49 ■ 50-89 ● 90-100



METRICS

[Expand view](#)

● First Contentful Paint
0.9 s

● Time to Interactive
0.9 s

● Speed Index
0.9 s

● Total Blocking Time
0 ms

● Largest Contentful Paint
0.9 s

● Cumulative Layout Shift
0

[View Treemap](#)



Show audits relevant to: **All** [FCP](#) [TBT](#) [LCP](#) [CLS](#)

OPPORTUNITIES

Opportunity

Estimated Savings

■ Eliminate render-blocking resources

0.69 s ^

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more](#). FCP LCP

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
...css/Main.css (127.0.0.1)	1.7 KiB	70 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	280 ms
...dist/d3.js (127.0.0.1)	563.8 KiB	670 ms

Reduce unused JavaScript 0.4 s ^

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn more](#). LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	563.8 KiB	470.7 KiB

Enable text compression 0.36 s ^

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. [Learn more](#). FCP LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	563.5 KiB	430.9 KiB
/gamedios/parse_orbits (127.0.0.1)	2.7 KiB	1.4 KiB

Minify JavaScript 0.16 s ^

Minifying JavaScript files can reduce payload sizes and script parse time. [Learn more](#). FCP LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	563.8 KiB	168.3 KiB

These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

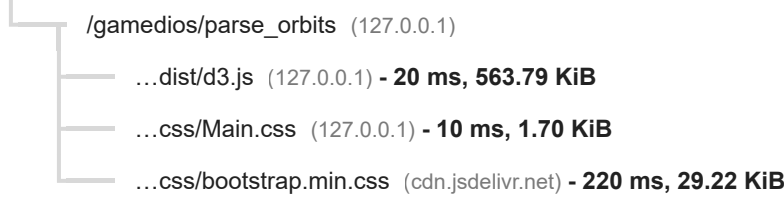
DIAGNOSTICS

○ Avoid chaining critical requests — 3 chains found ^

The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. [Learn more.](#) FCP LCP

Maximum critical path latency: **240 ms**

Initial Navigation



○ Keep request counts low and transfer sizes small — 5 requests • 625 KiB ^

To set budgets for the quantity and size of page resources, add a budget.json file. [Learn more.](#)

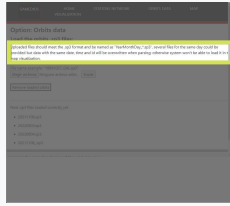
Resource Type	Requests	Transfer Size
Total	5	625.5 KiB
Script	1	563.8 KiB
Stylesheet	2	30.9 KiB
Other	1	28.0 KiB
Document	1	2.8 KiB
Image	0	0.0 KiB
Media	0	0.0 KiB
Font	0	0.0 KiB
Third-party	1	29.2 KiB

○ Largest Contentful Paint element — 1 element found ^

This is the largest contentful element painted within the viewport. [Learn More](#) LCP

Element

Element



Uploaded files should meet the .sp3 format and be named as "YearMonthDay_*.sp3"...

<p>

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (33)

Hide

● Properly size images



Serve images that are appropriately-sized to save cellular data and improve load time. [Learn more](#).

● Defer offscreen images



Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. [Learn more](#).

● Minify CSS



Minifying CSS files can reduce network payload sizes. [Learn more](#). FCP LCP

● Reduce unused CSS — Potential savings of 29 KiB



Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. [Learn more](#). FCP LCP

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	28.7 KiB

● Efficiently encode images



Optimized images load faster and consume less cellular data. [Learn more](#).

● Serve images in next-gen formats



Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn more.](#)

Preconnect to required origins ^

Consider adding `preconnect` or `dns-prefetch` resource hints to establish early connections to important third-party origins. [Learn more.](#) FCP LCP

Initial server response time was short — Root document took 0 ms ^

Keep the server response time for the main document short because all other requests depend on it. [Learn more.](#) FCP LCP

URL	Time Spent
/gamedios/parse_orbits (127.0.0.1)	0 ms

Avoid multiple page redirects ^

Redirects introduce additional delays before the page can be loaded. [Learn more.](#) FCP LCP

Preload key requests ^

Consider using ``<link rel=preload>`` to prioritize fetching resources that are currently requested later in page load. [Learn more.](#) FCP LCP

Use HTTP/2 ^

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. [Learn more.](#)

Use video formats for animated content ^

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. [Learn more](#) LCP

Remove duplicate modules in JavaScript bundles ^

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. TBT

● Avoid serving legacy JavaScript to modern browsers ^

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. [Learn More](#) TBT

● Preload Largest Contentful Paint image ^

Preload the image used by the LCP element in order to improve your LCP time. [Learn more.](#) LCP

● Avoids enormous network payloads — Total size was 625 KiB ^

Large network payloads cost users real money and are highly correlated with long load times. [Learn more.](#) LCP

Show 3rd-party resources (1)

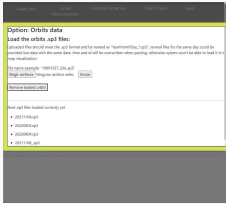
URL	Transfer Size
...dist/d3.js (127.0.0.1)	563.8 KiB
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB
...images/icon192.png (127.0.0.1)	28.0 KiB
/gamedios/parse_orbits (127.0.0.1)	2.8 KiB
...css/Main.css (127.0.0.1)	1.7 KiB

● Uses efficient cache policy on static assets — 0 resources found ^

A long cache lifetime can speed up repeat visits to your page. [Learn more.](#)

● Avoids an excessive DOM size — 35 elements ^

A large DOM will increase memory usage, cause longer [style calculations](#), and produce costly [layout reflows](#). [Learn more.](#) TBT

Statistic	Element	Value
Total DOM Elements		35
Maximum DOM Depth	Leaflet 	6
Maximum Child Elements	 <p>Option: Orbits data Load the orbits .sp3 files: Uploaded files should meet the... <div id="container"></p>	8

○ User Timing marks and measures ^

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. [Learn more.](#)

● JavaScript execution time — 0.0 s ^

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

URL	Total CPU Time	Script Evaluation	Script Parse
/gamedios/parse_orbits (127.0.0.1)	78 ms	2 ms	1 ms

● Minimizes main-thread work — 0.1 s ^

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more](#) TBT

Category	Time Spent
Style & Layout	53 ms
Other	47 ms
Parse HTML & CSS	15 ms
Script Parsing & Compilation	12 ms
Script Evaluation	11 ms
Rendering	1 ms

All text remains visible during webfont loads ^

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading. [Learn more.](#) FCP LCP

Minimize third-party usage — **Third-party code blocked the main thread for 0 ms** ^

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading. [Learn more.](#) TBT

Third-Party	Transfer Size	Main-Thread Blocking Time
JSDelivr CDN	29 KiB	0 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	29 KiB	0 ms

Lazy load third-party resources with facades ^

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. [Learn more.](#) TBT

Largest Contentful Paint image was not lazily loaded ^

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. [Learn more.](#)

Avoid large layout shifts ^

These DOM elements contribute most to the CLS of the page. CLS

Uses passive listeners to improve scrolling performance ^

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. [Learn more.](#)

Avoids `document.write()` ^

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. [Learn more.](#)

Avoid long main-thread tasks ^

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay.

[Learn more](#) TBT

Avoid non-composited animations ^

Animations which are not composited can be janky and increase CLS. [Learn more](#) CLS

Image elements have explicit `width` and `height` ^

Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn more](#) CLS

Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

Avoids `unload` event listeners ^

The `unload` event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use `pagehide` or `visibilitychange` events instead. [Learn more](#)



Accessibility

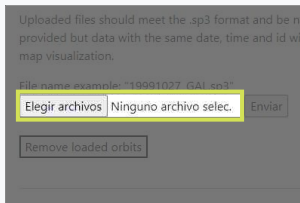
These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

NAMES AND LABELS

Form elements do not have associated labels ^

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. [Learn more.](#)

Failing Elements



div#container > div > form > input#uploadFile

```
<input type="file" id="uploadFile" accept=".sp3" multiple="">
```

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

BEST PRACTICES



`[user-scalable="no"]` is used in the `<meta name="viewport">` element or the `[maximum-scale]` attribute is less than 5. ^

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. [Learn more.](#)

Failing Elements

head > meta

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=no">
```

head > meta

```
<meta name="viewport" content="height=device-height, initial-scale=1.0, user-scalable=no">
```

These items highlight common accessibility best practices.

CONTRAST



Background and foreground colors do not have a sufficient contrast ratio. ^

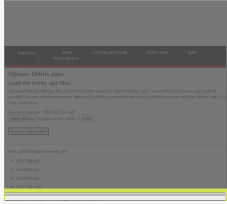
Low-contrast text is difficult or impossible for many users to read. [Learn more.](#)

Failing Elements

Leaflet

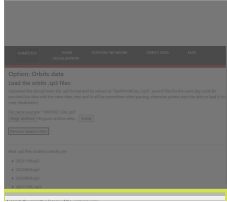
```
<a href="https://github.com/Leaflet/Leaflet/blob/main/LICENSE">
```

Failing Elements



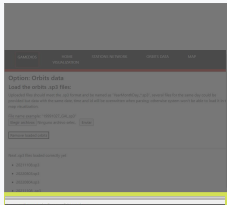
Access to the respective licenses of the used resources: © Leaflet © D3.js © Op...
<footer id="license_text">

D3.js



Access to the respective licenses of the used resources: © Leaflet © D3.js © Op...
<footer id="license_text">

OpenStreetMap



Access to the respective licenses of the used resources: © Leaflet © D3.js © Op...
<footer id="license_text">

These are opportunities to improve the legibility of your content.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

- The page has a logical tab order



Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more.](#)

- Interactive controls are keyboard focusable



Custom interactive controls are keyboard focusable and display a focus indicator. [Learn more.](#)

- Interactive elements indicate their purpose and state



Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn more.](#)

- The user's focus is directed to new content added to the page



If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn more.](#)

User focus is not accidentally trapped in a region ^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn more.](#)

Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more.](#)

Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn more.](#)

Visual order on the page follows DOM order ^

DOM order matches the visual order, improving navigation for assistive technology. [Learn more.](#)

Offscreen content is hidden from assistive technology ^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn more.](#)

HTML5 landmark elements are used to improve navigation ^

Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more.](#)

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review.](#)

PASSED AUDITS (11)

Hide

[aria-hidden="true"] is not present on the document <body> ^

Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document ``<body>``. [Learn more.](#)

Buttons have an accessible name ^

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. [Learn more.](#)

● The page contains a heading, skip link, or landmark region ^

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. [Learn more.](#)

● Document has a <title> element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

● [id] attributes on active, focusable elements are unique ^

All focusable elements must have a unique `id` to ensure that they're visible to assistive technologies. [Learn more.](#)

● <html> element has a [lang] attribute ^

If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. [Learn more.](#)

● <html> element has a valid value for its [lang] attribute ^

Specifying a valid [BCP 47 language](#) helps screen readers announce text properly. [Learn more.](#)

● Links have a discernible name ^

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn more.](#)

● Lists contain only elements and script supporting elements (<script> and <template>). ^

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. [Learn more.](#)

● List items () are contained within or parent elements ^

Screen readers require list items (``) to be contained within a parent `` or `` to be announced properly. [Learn more.](#)

Heading elements appear in a sequentially-descending order ^

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. [Learn more.](#)

NOT APPLICABLE (30)

Hide

`[accesskey]` values are unique ^

Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. [Learn more.](#)

`[aria-*]` attributes match their roles ^

Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. [Learn more.](#)

`button`, `link`, and `menuitem` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

`[aria-hidden="true"]` elements do not contain focusable descendents ^

Focusable descendents within an `[aria-hidden="true"]` element prevent those interactive elements from being available to users of assistive technologies like screen readers. [Learn more.](#)

ARIA input fields have accessible names ^

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `meter` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `progressbar` elements have accessible names ^

When a `progressbar` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

[role]s have all required [aria-*] attributes ^

Some ARIA roles have required attributes that describe the state of the element to screen readers. [Learn more.](#)

Elements with an ARIA [role] that require children to contain a specific [role] have all required children. ^

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. [Learn more.](#)

[role]s are contained by their required parent element ^

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more.](#)

[role] values are valid ^

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more.](#)

ARIA toggle fields have accessible names ^

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA tooltip elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA treeitem elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

[aria-*] attributes have valid values ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. [Learn more.](#)

[aria-*] attributes are valid and not misspelled ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. [Learn more.](#)

<dl>'s contain only properly-ordered <dt> and <dd> groups, <script>, <template> or <div> elements. ^

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn more.](#)

Definition list items are wrapped in <dl> elements ^

Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. [Learn more.](#)

ARIA IDs are unique ^

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. [Learn more.](#)

No form fields have multiple labels ^

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn more.](#)

<frame> or <iframe> elements have a title ^

Screen reader users rely on frame titles to describe the contents of frames. [Learn more.](#)

Image elements have [alt] attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

<input type="image"> elements have [alt] text ^

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. [Learn more.](#)

The document does not use <meta http-equiv="refresh"> ^

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. [Learn more.](#)

`<object>` elements have alternate text ^

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. [Learn more.](#)

No element has a `[tabindex]` value greater than 0 ^

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. [Learn more.](#)

Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table. ^

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. [Learn more.](#)

`<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe. ^

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. [Learn more.](#)

`[lang]` attributes have a valid value ^

Specifying a valid [BCP 47 language](#) on elements helps ensure that text is pronounced correctly by a screen reader. [Learn more.](#)

`<video>` elements contain a `<track>` element with `[kind="captions"]` ^

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. [Learn more.](#)



Best Practices

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks ^

A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. [Learn more](#)

Description	Directive	Severity
No CSP found in enforcement mode		High

GENERAL

○ Detected JavaScript libraries ^

All front-end JavaScript libraries detected on the page. [Learn more](#).

Name	Version
D3	7.3.0

▲ Missing source maps for large first-party JavaScript ^

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. [Learn more](#).

URL	Map URL
...dist/d3.js (127.0.0.1)	
Large JavaScript file is missing a source map	

PASSED AUDITS (12)

Hide

● Uses HTTPS ^

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more](#).

● Avoids requesting the geolocation permission on page load ^

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more](#).

● Avoids requesting the notification permission on page load ^

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more](#).

● Avoids front-end JavaScript libraries with known security vulnerabilities ^

Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. [Learn more](#).

● Allows users to paste into password fields ^

Preventing password pasting undermines good security policy. [Learn more](#).

● Displays images with correct aspect ratio ^

Image display dimensions should match natural aspect ratio. [Learn more](#).

● Serves images with appropriate resolution ^

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn more](#).

● Page has the HTML doctype ^

Specifying a doctype prevents the browser from switching to quirks-mode. [Learn more](#).

● Properly defines charset ^

A character encoding declaration is required. It can be done with a `` tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. [Learn more](#).

● Avoids deprecated APIs ^

Deprecated APIs will eventually be removed from the browser. [Learn more.](#)

● No browser errors logged to the console ^

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. [Learn more](#)

● No issues in the [Issues](#) panel in Chrome Devtools ^

Issues logged to the `Issues` panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

NOT APPLICABLE (1)

Hide

Fonts with `font-display: optional` are preloaded ^


Preload `optional` fonts so first-time visitors may use them. [Learn more](#)



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more.](#)

CONTENT BEST PRACTICES

 Document does not have a meta description ^

Meta descriptions may be included in search results to concisely summarize page content. [Learn more](#).

Format your HTML in a way that enables crawlers to better understand your app's content.

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Hide

Structured data is valid ^

Run the [Structured Data Testing Tool](#) and the [Structured Data Linter](#) to validate structured data. [Learn more](#).

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (8) Hide

Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more](#).

Page has successful HTTP status code ^

Pages with unsuccessful HTTP status codes may not be indexed properly. [Learn more](#).

Links have descriptive text ^

Descriptive link text helps search engines understand your content. [Learn more](#).

Links are crawlable ^

Search engines may use `href` attributes on links to crawl websites. Ensure that the `href` attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. [Learn More](#)

Page isn't blocked from indexing ^

Search engines are unable to include your pages in search results if they don't have permission to crawl them. [Learn more.](#)

Document has a valid `hreflang` ^

hreflang links tell search engines what version of a page they should list in search results for a given language or region. [Learn more.](#)

Document avoids plugins ^

Search engines can't index plugin content, and many devices restrict plugins or don't support them. [Learn more.](#)

NOT APPLICABLE (5)

Hide

robots.txt is valid ^

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. [Learn more.](#)

Image elements have `[alt]` attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

Document has a valid `rel=canonical` ^

Canonical links suggest which URL to show in search results. [Learn more.](#)

Document uses legible font sizes ^

Font sizes less than 12px are too small to be legible and require mobile visitors to “pinch to zoom” in order to read. Strive to have >60% of page text ≥ 12 px. [Learn more.](#)

Tap targets are sized appropriately ^

Interactive elements like buttons and links should be large enough (48x48px), and have enough space around them, to be easy enough to tap without overlapping onto other elements. [Learn more](#).



PWA

These checks validate the aspects of a Progressive Web App. [Learn more](#).

INSTALLABLE

 Web app manifest or service worker do not meet the installability requirements — 1 reason 

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more](#).

Failure reason

Page has no manifest <link> URL

PWA OPTIMIZED

 Does not register a service worker that controls page and `start_url` 

The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. [Learn more](#).

 Is not configured for a custom splash screen **Failures: No manifest was fetched.** 

A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. [Learn more](#).

- ▲ Does not set a theme color for the address bar. ^
Failures: No manifest was fetched, No ``<meta name="theme-color">`` tag found.

The browser address bar can be themed to match your site. [Learn more.](#)

- Content is sized correctly for the viewport ^

If the width of your app's content doesn't match the width of the viewport, your app might not be optimized for mobile screens. [Learn more.](#)

- Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more.](#) TBT

- ▲ Does not provide a valid `apple-touch-icon` ^

For ideal appearance on iOS when users add a progressive web app to the home screen, define an `apple-touch-icon`. It must point to a non-transparent 192px (or 180px) square PNG. [Learn More.](#)

- ▲ Manifest doesn't have a maskable icon **No manifest was fetched** ^

A maskable icon ensures that the image fills the entire shape without being letterboxed when installing the app on a device. [Learn more.](#)

ADDITIONAL ITEMS TO MANUALLY CHECK (3)

Hide

- Site works cross-browser ^

To reach the most number of users, sites should work across every major browser. [Learn more.](#)







- Page transitions don't feel like they block on the network ^

Transitions should feel snappy as you tap around, even on a slow network. This experience is key to a user's perception of performance. [Learn more.](#)

- Each page has a URL ^

Ensure individual pages are deep linkable via URL and that URLs are unique for the purpose of shareability on social media. [Learn more.](#)

These checks are required by the baseline [PWA Checklist](#) but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.

 Captured at Nov 20, 2022, 7:28 PM GMT+1	 Emulated Desktop with Lighthouse 9.6.6	 Single page load
 Initial page load	 Custom throttling	 Using Chromium 107.0.0.0 with devtools

Generated by **Lighthouse 9.6.6** | [File an issue](#)

A.2. Análisis posterior

A.2.1. Página de visualización del mapa



Performance



Accessibility



Best Practices



SEO



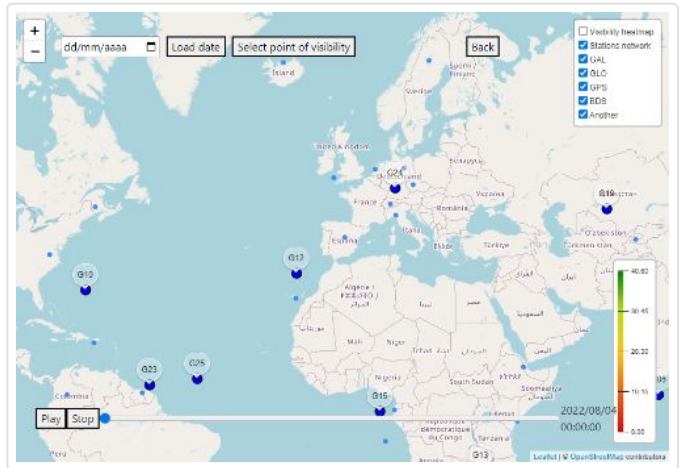
PWA



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

- ▲ 0–49
- 50–89
- 90–100



METRICS

[Collapse view](#)

- First Contentful Paint

0.7 s

First Contentful Paint marks the time at which the first text or image is painted. [Learn more.](#)

- Time to Interactive

0.9 s

Time to interactive is the amount of time it takes for the page to become fully interactive. [Learn more.](#)

- Speed Index

1.9 s

Speed Index shows how quickly the contents of a page are visibly populated. [Learn more.](#)

- Total Blocking Time

30 ms

Sum of all time periods between FCP and Time to Interactive, when task length exceeded 50ms, expressed in milliseconds. [Learn more.](#)

- Largest Contentful Paint

1.2 s

Largest Contentful Paint marks the time at which the largest text or image is painted. [Learn more](#)

- Cumulative Layout Shift

0.003

Cumulative Layout Shift measures the movement of visible elements within the viewport. [Learn more.](#)

[View Treemap](#)



Show audits relevant to: **All** [FCP](#) [TBT](#) [LCP](#) [CLS](#)

OPPORTUNITIES

Opportunity

Estimated Savings

▲ Reduce initial server response time 0.69 s ^

Keep the server response time for the main document short because all other requests depend on it. [Learn more.](#) [FCP](#) [LCP](#)

URL	Time Spent
/gamedios/map (127.0.0.1)	790 ms

■ Eliminate render-blocking resources 0.49 s ^

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more.](#) [FCP](#) [LCP](#)

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
...leaflet/leaflet.css (127.0.0.1)	3.6 KiB	60 ms
...css/Map.css (127.0.0.1)	0.8 KiB	100 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	27.4 KiB	290 ms
...dist/d3.js (127.0.0.1)	136.0 KiB	420 ms
...leaflet/leaflet-src.js (127.0.0.1)	107.2 KiB	380 ms

■ Reduce unused JavaScript 0.16 s ^

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn more.](#) [LCP](#)

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	136.0 KiB	106.8 KiB

URL	Transfer Size	Potential Savings
...leaflet/leaflet-src.js (127.0.0.1)	107.2 KiB	37.8 KiB
...map/Map.js	14.1 KiB	6.0 KiB
...layer/vector/Canvas.js	2.8 KiB	2.4 KiB
...layer/Popup.js	3.6 KiB	2.0 KiB
...dom/Draggable.js	1.7 KiB	1.2 KiB
...layer/marker/Marker.js	2.3 KiB	1.2 KiB

These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

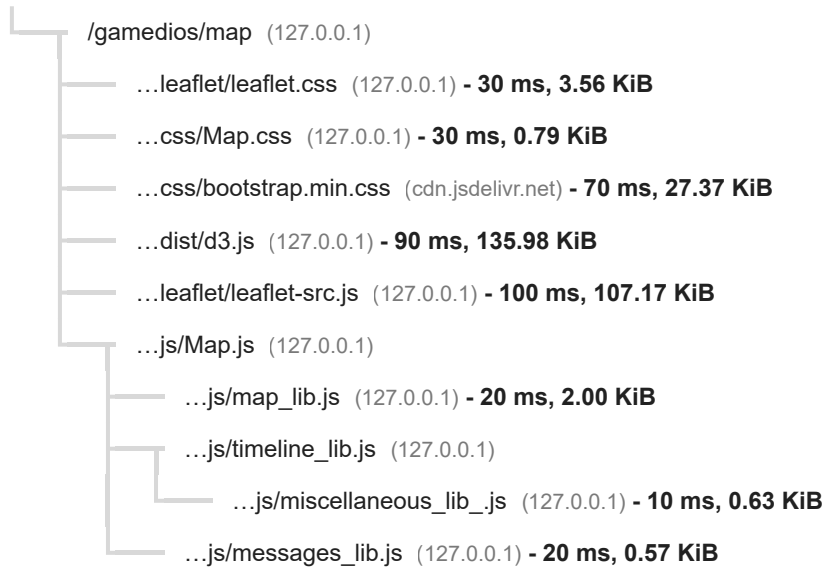
DIAGNOSTICS

○ Avoid chaining critical requests — 8 chains found ^

The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. [Learn more.](#) FCP LCP

Maximum critical path latency: **1,030 ms**

Initial Navigation



○ Keep request counts low and transfer sizes small — 31 requests • 545 KiB ^

To set budgets for the quantity and size of page resources, add a budget.json file. [Learn more.](#)

Resource Type	Requests	Transfer Size
Total	31	544.5 KiB

Resource Type	Requests	Transfer Size
Script	7	250.2 KiB
Other	4	146.6 KiB
Image	16	114.9 KiB
Stylesheet	3	31.7 KiB
Document	1	1.1 KiB
Media	0	0.0 KiB
Font	0	0.0 KiB
Third-party	17	142.3 KiB

Largest Contentful Paint element — 1 element found

This is the largest contentful element painted within the viewport. [Learn More](#) LCP

Element



```
div.leaflet-pane > div.leaflet-layer > div.leaflet-tile-container > img.leaflet-tile

```

Avoid large layout shifts — 1 element found

These DOM elements contribute most to the CLS of the page. CLS

Element	CLS Contribution
 <pre>Play Stop 2022/08/04 00:00:00 <div id="botton_buttons" class="over_map"></pre>	0.003

Avoid long main-thread tasks — 1 long task found

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay.

[Learn more](#) TBT

URL	Start Time	Duration
-----	------------	----------

URL	Start Time	Duration
...leaflet/leaflet-src.js (127.0.0.1)	900 ms	108 ms

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (32)

Hide

● Properly size images



Serve images that are appropriately-sized to save cellular data and improve load time. [Learn more](#).

● Defer offscreen images



Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. [Learn more](#).

● Minify CSS



Minifying CSS files can reduce network payload sizes. [Learn more](#). FCP LCP

● Minify JavaScript — Potential savings of 99 KiB



Minifying JavaScript files can reduce payload sizes and script parse time. [Learn more](#). FCP LCP

URL	Transfer Size	Potential Savings
...leaflet/leaflet-src.js (127.0.0.1)	107.2 KiB	58.9 KiB
...dist/d3.js (127.0.0.1)	136.0 KiB	40.6 KiB

● Reduce unused CSS — Potential savings of 27 KiB



Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. [Learn more](#). FCP LCP

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	27.4 KiB	26.9 KiB

Efficiently encode images ^

Optimized images load faster and consume less cellular data. [Learn more.](#)

Serve images in next-gen formats ^

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn more.](#)

Enable text compression ^

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. [Learn more.](#) FCP LCP

Preconnect to required origins ^

Consider adding `preconnect` or `dns-prefetch` resource hints to establish early connections to important third-party origins. [Learn more.](#) FCP LCP

Avoid multiple page redirects ^

Redirects introduce additional delays before the page can be loaded. [Learn more.](#) FCP LCP

Preload key requests ^

Consider using ``<link rel=preload>`` to prioritize fetching resources that are currently requested later in page load. [Learn more.](#) FCP LCP

Use HTTP/2 ^

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. [Learn more.](#)

Use video formats for animated content ^

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. [Learn more](#)

LCP

● Remove duplicate modules in JavaScript bundles

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. TBT

● Avoid serving legacy JavaScript to modern browsers

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. Learn More TBT

● Preload Largest Contentful Paint image

Preload the image used by the LCP element in order to improve your LCP time. Learn more. LCP

	URL	Potential Savings
 <pre> div.leaflet-pane > div.leaflet-layer > div.leaflet-tile- container > img.leaflet-tile </pre> <p></p>	...4/3.png (b.tile.openstreetmap.org)	0 ms

● Avoids enormous network payloads — Total size was 545 KiB

Large network payloads cost users real money and are highly correlated with long load times. Learn more. LCP

Show 3rd-party resources (5)

URL	Transfer Size
...dist/d3.js (127.0.0.1)	136.0 KiB

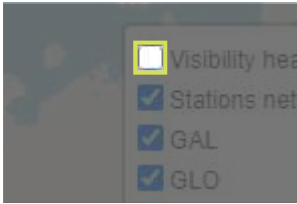

URL	Transfer Size
...leaflet/leaflet-src.js (127.0.0.1)	107.2 KiB
/gamedios/get_data_from_date?date=2022-08-04 (127.0.0.1)	67.3 KiB
/gamedios/get_heatmap (127.0.0.1)	49.8 KiB
...images/icon192.png (127.0.0.1)	28.0 KiB
...css/bootstrap.min.css (cdn.jsdelivr.net)	27.4 KiB
...4/2.png (a.tile.openstreetmap.org)	15.1 KiB
...4/3.png (b.tile.openstreetmap.org)	13.5 KiB
...4/4.png (c.tile.openstreetmap.org)	9.7 KiB
...2/2.png (b.tile.openstreetmap.org)	9.7 KiB

● Uses efficient cache policy on static assets — 0 resources found ^

A long cache lifetime can speed up repeat visits to your page. [Learn more.](#)

● Avoids an excessive DOM size — 240 elements ^

A large DOM will increase memory usage, cause longer [style calculations](#), and produce costly [layout reflows](#). [Learn more.](#) TBT

Statistic	Element	Value
Total DOM Elements		240
Maximum DOM Depth	 <pre>div.leaflet-control-layers-overlays > label > div > input.leaflet-control-layers-selector <input type="checkbox" class="leaflet-control-layers-selector"></pre>	11
Maximum Child Elements	 <pre>div.leaflet-pane > div.leaflet-pane > svg.leaflet-zoom- animated > g <g></pre>	90

○ User Timing marks and measures ^

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. [Learn more.](#)

● JavaScript execution time — 0.2 s ^

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

URL	Total CPU Time	Script Evaluation	Script Parse
...leaflet/leaflet-src.js (127.0.0.1)	276 ms	179 ms	7 ms
Unattributable	120 ms	5 ms	0 ms
/gamedios/map (127.0.0.1)	92 ms	2 ms	1 ms

● Minimizes main-thread work — 0.5 s ^

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

Category	Time Spent
Script Evaluation	191 ms
Other	139 ms
Style & Layout	94 ms
Rendering	28 ms
Garbage Collection	24 ms
Script Parsing & Compilation	19 ms
Parse HTML & CSS	12 ms

● All text remains visible during webfont loads ^

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading.

[Learn more.](#) FCP LCP

● Minimize third-party usage — Third-party code blocked the main thread for 0 ms ^

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading.

[Learn more.](#) TBT

Third-Party	Transfer Size	Main-Thread Blocking Time
JSDelivr CDN	27 KiB	0 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	27 KiB	0 ms

Lazy load third-party resources with facades ^

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. [Learn more.](#) TBT

Largest Contentful Paint image was not lazily loaded ^

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. [Learn more.](#)

Element



```
div.leaflet-pane > div.leaflet-layer > div.leaflet-tile-container > img.leaflet-tile

```

Uses passive listeners to improve scrolling performance ^

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. [Learn more.](#)

Avoids `document.write()` ^

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. [Learn more.](#)

Avoid non-composited animations ^

Animations which are not composited can be janky and increase CLS. [Learn more](#) CLS

Image elements have explicit `width` and `height` ^

Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn more](#) CLS

- Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

- Avoids `unload` event listeners ^

The `unload` event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use `pagehide` or `visibilitychange` events instead. [Learn more](#)



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

- The page has a logical tab order ^

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more](#).

- Interactive controls are keyboard focusable ^

Custom interactive controls are keyboard focusable and display a focus indicator. [Learn more](#).

- Interactive elements indicate their purpose and state ^

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn more.](#)

The user's focus is directed to new content added to the page ^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn more.](#)

User focus is not accidentally trapped in a region ^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn more.](#)

Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more.](#)

Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn more.](#)

Visual order on the page follows DOM order ^

DOM order matches the visual order, improving navigation for assistive technology. [Learn more.](#)

Offscreen content is hidden from assistive technology ^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn more.](#)

HTML5 landmark elements are used to improve navigation ^

Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more.](#)

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review.](#)

PASSED AUDITS (19)

Hide

[aria-*] attributes match their roles ^

Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. [Learn more.](#)

● [aria-hidden="true"] is not present on the document <body> ^

Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document ``<body>``. [Learn more.](#)

● [role]s have all required [aria-*] attributes ^

Some ARIA roles have required attributes that describe the state of the element to screen readers. [Learn more.](#)

● [role] values are valid ^

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more.](#)

● [aria-*] attributes have valid values ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. [Learn more.](#)

● [aria-*] attributes are valid and not misspelled ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. [Learn more.](#)

● Buttons have an accessible name ^

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. [Learn more.](#)

● ARIA IDs are unique ^

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. [Learn more.](#)

● Image elements have [alt] attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

● Form elements have associated labels ^

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. [Learn more.](#)

● `[user-scalable="no"]` is not used in the `<meta name="viewport">` element and the `[maximum-scale]` attribute is not less than 5. ^

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. [Learn more.](#)

● The page contains a heading, skip link, or landmark region ^

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. [Learn more.](#)

● Background and foreground colors have a sufficient contrast ratio ^

Low-contrast text is difficult or impossible for many users to read. [Learn more.](#)

● Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

● `[id]` attributes on active, focusable elements are unique ^

All focusable elements must have a unique `id` to ensure that they're visible to assistive technologies. [Learn more.](#)

● `<html>` element has a `[lang]` attribute ^

If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the

default language, then the screen reader might not announce the page's text correctly. [Learn more](#).

● <html> element has a valid value for its [lang] attribute ^

Specifying a valid [BCP 47 language](#) helps screen readers announce text properly. [Learn more](#).

● Links have a discernible name ^

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn more](#).

● No element has a [tabindex] value greater than 0 ^

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. [Learn more](#).

NOT APPLICABLE (25)

Hide

○ [accesskey] values are unique ^

Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. [Learn more](#).

○ button, link, and menuitem elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more](#).

○ [aria-hidden="true"] elements do not contain focusable descendents ^

Focusable descendents within an `[aria-hidden="true"]` element prevent those interactive elements from being available to users of assistive technologies like screen readers. [Learn more](#).

○ ARIA input fields have accessible names ^

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more](#).

ARIA `meter` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `progressbar` elements have accessible names ^

When a `progressbar` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

Elements with an ARIA `[role]` that require children to contain a specific `[role]` have all required children. ^

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. [Learn more.](#)

`[role]`s are contained by their required parent element ^

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more.](#)

ARIA toggle fields have accessible names ^

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `tooltip` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `treeitem` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

`<dl>`'s contain only properly-ordered `<dt>` and `<dd>` groups, `<script>`, `<template>` or `<div>` elements. ^

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn more.](#)

Definition list items are wrapped in `<dl>` elements ^

Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. [Learn more.](#)

No form fields have multiple labels ^

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn more.](#)

`<frame>` or `<iframe>` elements have a title ^

Screen reader users rely on frame titles to describe the contents of frames. [Learn more.](#)

Heading elements appear in a sequentially-descending order ^

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. [Learn more.](#)

`<input type="image">` elements have `[alt]` text ^

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. [Learn more.](#)

Lists contain only `` elements and script supporting elements (`<script>` and `<template>`). ^

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. [Learn more.](#)

List items (``) are contained within `` or `` parent elements ^

Screen readers require list items (``) to be contained within a parent `` or `` to be announced properly. [Learn more.](#)

The document does not use `<meta http-equiv="refresh">` ^

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. [Learn more.](#)

`<object>` elements have alternate text ^

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. [Learn more.](#)

Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table. ^

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. [Learn more.](#)

- `<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe. ^

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. [Learn more.](#)

- `[lang]` attributes have a valid value ^

Specifying a valid [BCP 47 language](#) on elements helps ensure that text is pronounced correctly by a screen reader. [Learn more.](#)

- `<video>` elements contain a `<track>` element with `[kind="captions"]` ^

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. [Learn more.](#)



Best Practices

GENERAL

- ▲ Uses deprecated APIs — 1 warning found ^

Deprecated APIs will eventually be removed from the browser. [Learn more.](#)

Deprecation / Warning	Source
...	/gamedios/map:0:-1 (127.0.0.1)

- Detected JavaScript libraries ^

All front-end JavaScript libraries detected on the page. [Learn more.](#)

Name	Version
Leaflet	1.6.0
D3	7.3.0

▲ Missing source maps for large first-party JavaScript

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. [Learn more.](#)

URL	Map URL
...dist/d3.js (127.0.0.1)	
Large JavaScript file is missing a source map	
...leaflet/leaflet-src.js (127.0.0.1)	...leaflet/leaflet-src.js.map (127.0.0.1)

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks

A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. [Learn more](#)

Description	Directive	Severity
No CSP found in enforcement mode		High

PASSED AUDITS (11)

Hide

● Uses HTTPS

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more.](#)

● Avoids requesting the geolocation permission on page load ^

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more.](#)

● Avoids requesting the notification permission on page load ^

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more.](#)

● Avoids front-end JavaScript libraries with known security vulnerabilities ^

Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. [Learn more.](#)

● Allows users to paste into password fields ^

Preventing password pasting undermines good security policy. [Learn more.](#)

● Displays images with correct aspect ratio ^

Image display dimensions should match natural aspect ratio. [Learn more.](#)

● Serves images with appropriate resolution ^

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn more.](#)

● Page has the HTML doctype ^

Specifying a doctype prevents the browser from switching to quirks-mode. [Learn more.](#)

● Properly defines charset ^

A character encoding declaration is required. It can be done with a `` tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. [Learn more.](#)

No browser errors logged to the console ^

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. [Learn more](#)

No issues in the [Issues](#) panel in Chrome Devtools ^

Issues logged to the `Issues` panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

NOT APPLICABLE (1)

Hide

Fonts with `font-display: optional` are preloaded ^

Preload `optional` fonts so first-time visitors may use them. [Learn more](#)



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more](#).

ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Hide

Structured data is valid ^

Run the [Structured Data Testing Tool](#) and the [Structured Data Linter](#) to validate structured data. [Learn more](#).

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (10)

Hide

<p>● Has a <code><meta name="viewport"></code> tag with <code>width</code> or <code>initial-scale</code> ^</p> <p>A <code><meta name="viewport"></code> not only optimizes your app for mobile screen sizes, but also prevents a 300 millisecond delay to user input. Learn more. TBT</p>
<p>● Document has a <code><title></code> element ^</p> <p>The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. Learn more.</p>
<p>● Document has a meta description ^</p> <p>Meta descriptions may be included in search results to concisely summarize page content. Learn more.</p>
<p>● Page has successful HTTP status code ^</p> <p>Pages with unsuccessful HTTP status codes may not be indexed properly. Learn more.</p>
<p>● Links have descriptive text ^</p> <p>Descriptive link text helps search engines understand your content. Learn more.</p>
<p>● Links are crawlable ^</p> <p>Search engines may use <code>href</code> attributes on links to crawl websites. Ensure that the <code>href</code> attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. Learn More</p>
<p>● Page isn't blocked from indexing ^</p> <p>Search engines are unable to include your pages in search results if they don't have permission to crawl them. Learn more.</p>
<p>● Image elements have <code>[alt]</code> attributes ^</p>

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

Document has a valid hreflang ^

hreflang links tell search engines what version of a page they should list in search results for a given language or region. [Learn more.](#)

Document avoids plugins ^

Search engines can't index plugin content, and many devices restrict plugins or don't support them. [Learn more.](#)

NOT APPLICABLE (4)

Hide

robots.txt is valid ^

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. [Learn more.](#)

Document has a valid rel=canonical ^

Canonical links suggest which URL to show in search results. [Learn more.](#)

Document uses legible font sizes ^

Font sizes less than 12px are too small to be legible and require mobile visitors to “pinch to zoom” in order to read. Strive to have >60% of page text ≥12px. [Learn more.](#)

Tap targets are sized appropriately ^

Interactive elements like buttons and links should be large enough (48x48px), and have enough space around them, to be easy enough to tap without overlapping onto other elements. [Learn more.](#)



PWA

These checks validate the aspects of a Progressive Web App. [Learn more.](#)

INSTALLABLE

Web app manifest or service worker do not meet the installability requirements — 1 reason

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more.](#)

Failure reason

Page has no manifest <link> URL

PWA OPTIMIZED

Does not register a service worker that controls page and `start_url`

The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. [Learn more.](#)

Is not configured for a custom splash screen **Failures: No manifest was fetched.**

A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. [Learn more.](#)

Does not set a theme color for the address bar. **Failures: No manifest was fetched, No ``<meta name="theme-color">`` tag found.**

The browser address bar can be themed to match your site. [Learn more.](#)

Content is sized correctly for the viewport

If the width of your app's content doesn't match the width of the viewport, your app might not be optimized for mobile screens. [Learn more.](#)

● Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more.](#) TBT

▲ Does not provide a valid `apple-touch-icon` ^

For ideal appearance on iOS when users add a progressive web app to the home screen, define an `apple-touch-icon`. It must point to a non-transparent 192px (or 180px) square PNG. [Learn More.](#)

▲ Manifest doesn't have a maskable icon No manifest was fetched ^

A maskable icon ensures that the image fills the entire shape without being letterboxed when installing the app on a device. [Learn more.](#)

ADDITIONAL ITEMS TO MANUALLY CHECK (3)

Hide

Site works cross-browser ^

To reach the most number of users, sites should work across every major browser. [Learn more.](#)

Page transitions don't feel like they block on the network ^

Transitions should feel snappy as you tap around, even on a slow network. This experience is key to a user's perception of performance. [Learn more.](#)

Each page has a URL ^

Ensure individual pages are deep linkable via URL and that URLs are unique for the purpose of shareability on social media. [Learn more.](#)

These checks are required by the baseline [PWA Checklist](#) but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.

🕒 Initial page load

🔧 Custom throttling

🖥️ Using Chromium 107.0.0.0 with devtools

Generated by **Lighthouse** 9.6.6 | [File an issue](#)

A.2.2. Página del menú principal



Performance



Accessibility



Best Practices



SEO



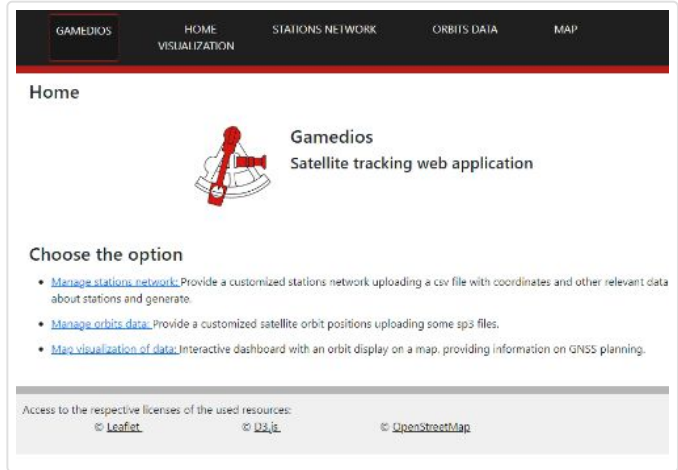
PWA



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

▲ 0–49 ■ 50–89 ● 90–100



METRICS

[Expand view](#)

● First Contentful Paint
0.5 s

● Speed Index
0.5 s

● Largest Contentful Paint
0.5 s

● Time to Interactive
0.5 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0

[View Treemap](#)



Show audits relevant to: **All** [FCP](#) [TBT](#) [LCP](#) [CLS](#)

OPPORTUNITIES

Opportunity

Estimated Savings

■ Eliminate render-blocking resources

0.35 s ^

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more](#). FCP LCP

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
...css/Main.css (127.0.0.1)	0.8 KiB	60 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	330 ms
...dist/d3.js (127.0.0.1)	136.0 KiB	340 ms

Reduce unused JavaScript

0.18 s ^

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn more](#). LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	136.0 KiB	113.5 KiB

These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

DIAGNOSTICS

Avoid chaining critical requests — 3 chains found ^

The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. [Learn more](#). FCP LCP

Maximum critical path latency: **220 ms**

Initial Navigation



Keep request counts low and transfer sizes small — 6 requests • 196 KiB ^

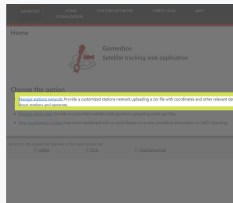
To set budgets for the quantity and size of page resources, add a budget.json file. [Learn more](#).

Resource Type	Requests	Transfer Size
Total	6	195.6 KiB
Script	1	136.0 KiB
Stylesheet	2	30.0 KiB
Image	1	28.0 KiB
Document	1	1.3 KiB
Other	1	0.2 KiB
Media	0	0.0 KiB
Font	0	0.0 KiB
Third-party	1	29.2 KiB

○ Largest Contentful Paint element — 1 element found ^

This is the largest contentful element painted within the viewport. [Learn More](#) LCP

Element



Manage stations network: Provide a customized stations network uploading a csv ...

``


More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (35)

Hide

● Properly size images — Potential savings of 17 KiB ^

Serve images that are appropriately-sized to save cellular data and improve load time. [Learn more](#).

	URL	Resource Size	Potential Savings
	sexta nt logo <code><img id="</code>	...images/icon192.png (127.0.0.1) 27.8 KiB	16.9 KiB

URL	Resource Size	Potential Savings
<pre>logo_icon" src="/static/images/icon192.png" style="width:120px;height:120px;" alt="sextant logo"></pre>		

● Defer offscreen images ^

Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. [Learn more.](#)

● Minify CSS ^

Minifying CSS files can reduce network payload sizes. [Learn more.](#) FCP LCP

● Minify JavaScript — Potential savings of 41 KiB ^

Minifying JavaScript files can reduce payload sizes and script parse time. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	136.0 KiB	40.6 KiB

● Reduce unused CSS — Potential savings of 29 KiB ^

Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. [Learn more.](#) FCP LCP


URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	28.8 KiB

● Efficiently encode images ^

Optimized images load faster and consume less cellular data. [Learn more.](#)

● Serve images in next-gen formats — Potential savings of 22 KiB ^

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn more.](#)

	URL	Resource Size	Potential Savings
 <pre> sextant logo </pre>	...images/icon192.png (127.0.0.1)	27.8 KiB	22.0 KiB

● Enable text compression ^

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. [Learn more.](#) FCP LCP

● Preconnect to required origins ^

Consider adding `preconnect` or `dns-prefetch` resource hints to establish early connections to important third-party origins. [Learn more.](#) FCP LCP

● Initial server response time was short — Root document took 20 ms ^

Keep the server response time for the main document short because all other requests depend on it. [Learn more.](#) FCP LCP

URL	Time Spent
/gamedios (127.0.0.1)	20 ms

● Avoid multiple page redirects ^

Redirects introduce additional delays before the page can be loaded. [Learn more.](#) FCP LCP

○ Preload key requests ^

Consider using ``<link rel=preload>`` to prioritize fetching resources that are currently requested later in page load. [Learn more.](#) FCP LCP

● Use HTTP/2 ^

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. [Learn more.](#)

● Use video formats for animated content ^

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. [Learn more](#) LCP

● Remove duplicate modules in JavaScript bundles ^

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. TBT

● Avoid serving legacy JavaScript to modern browsers ^

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. [Learn More](#) TBT

● Preload Largest Contentful Paint image ^

Preload the image used by the LCP element in order to improve your LCP time. [Learn more.](#) LCP

● Avoids enormous network payloads — Total size was 196 KiB ^

Large network payloads cost users real money and are highly correlated with long load times. [Learn more.](#) LCP

Show 3rd-party resources (1)


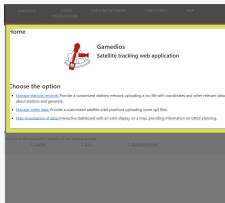
URL	Transfer Size
...dist/d3.js (127.0.0.1)	136.0 KiB
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB
...images/icon192.png (127.0.0.1)	28.0 KiB
/gamedios (127.0.0.1)	1.3 KiB
...css/Main.css (127.0.0.1)	0.8 KiB
...images/icon192.png (127.0.0.1)	0.2 KiB

● Uses efficient cache policy on static assets — 0 resources found ^

A long cache lifetime can speed up repeat visits to your page. [Learn more.](#)

● Avoids an excessive DOM size — 36 elements ^

A large DOM will increase memory usage, cause longer [style calculations](#), and produce costly [layout reflows](#). [Learn more.](#) TBT

Statistic	Element	Value
Total DOM Elements		36
Maximum DOM Depth	 <p>Leaflet </p>	6
Maximum Child Elements	 <p>Home Gamedios Satellite tracking web application Choose the option Manage ... <div id="container"></p>	7

○ User Timing marks and measures ^

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. [Learn more.](#)

● JavaScript execution time — 0.0 s ^

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

URL	Total CPU Time	Script Evaluation	Script Parse
/gamedios (127.0.0.1)	65 ms	2 ms	1 ms

● Minimizes main-thread work — 0.1 s ^

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

Category	Time Spent
Style & Layout	38 ms
Other	28 ms
Script Parsing & Compilation	13 ms
Rendering	12 ms
Script Evaluation	10 ms
Parse HTML & CSS	5 ms

● All text remains visible during webfont loads ^

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading. [Learn more.](#) FCP LCP

● Minimize third-party usage — Third-party code blocked the main thread for 0 ms ^

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading. [Learn more.](#) TBT

Third-Party	Transfer Size	Main-Thread Blocking Time
JSDelivr CDN	29 KiB	0 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	29 KiB	0 ms

○ Lazy load third-party resources with facades ^

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. [Learn more](#). TBT

Largest Contentful Paint image was not lazily loaded ^

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. [Learn more](#).

Avoid large layout shifts ^

These DOM elements contribute most to the CLS of the page. CLS

Uses passive listeners to improve scrolling performance ^

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. [Learn more](#).

Avoids `document.write()` ^

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. [Learn more](#).

Avoid long main-thread tasks ^

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay. [Learn more](#) TBT

Avoid non-composited animations ^

Animations which are not composited can be janky and increase CLS. [Learn more](#) CLS

Image elements have explicit `width` and `height` ^

Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn more](#) CLS

Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

● Avoids `unload` event listeners ^

The `unload` event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use `pagehide` or `visibilitychange` events instead. [Learn more](#)



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

<input type="radio"/> The page has a logical tab order ^
Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. Learn more .
<input type="radio"/> Interactive controls are keyboard focusable ^
Custom interactive controls are keyboard focusable and display a focus indicator. Learn more .
<input type="radio"/> Interactive elements indicate their purpose and state ^
Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. Learn more .
<input type="radio"/> The user's focus is directed to new content added to the page ^
If new content, such as a dialog, is added to the page, the user's focus is directed to it. Learn more .

<input type="radio"/> User focus is not accidentally trapped in a region ^
A user can tab into and out of any control or region without accidentally trapping their focus. Learn more.
<input type="radio"/> Custom controls have associated labels ^
Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. Learn more.
<input type="radio"/> Custom controls have ARIA roles ^
Custom interactive controls have appropriate ARIA roles. Learn more.
<input type="radio"/> Visual order on the page follows DOM order ^
DOM order matches the visual order, improving navigation for assistive technology. Learn more.
<input type="radio"/> Offscreen content is hidden from assistive technology ^
Offscreen content is hidden with display: none or aria-hidden=true. Learn more.
<input type="radio"/> HTML5 landmark elements are used to improve navigation ^
Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. Learn more.

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review.](#)

PASSED AUDITS (12)

Hide

<input checked="" type="radio"/> [aria-hidden="true"] is not present on the document <body> ^
Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document ` <code><body></code> `. Learn more.
<input checked="" type="radio"/> Image elements have [alt] attributes ^
Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. Learn more.

- `[user-scalable="no"]` is not used in the `<meta name="viewport">` element and the `[maximum-scale]` attribute is not less than 5. ^

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. [Learn more.](#)

- The page contains a heading, skip link, or landmark region ^

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. [Learn more.](#)

- Background and foreground colors have a sufficient contrast ratio ^

Low-contrast text is difficult or impossible for many users to read. [Learn more.](#)

- Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

- `<html>` element has a `[lang]` attribute ^

If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. [Learn more.](#)

- `<html>` element has a valid value for its `[lang]` attribute ^

Specifying a valid [BCP 47 language](#) helps screen readers announce text properly. [Learn more.](#)

- Links have a discernible name ^

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn more.](#)

- Lists contain only `` elements and script supporting elements (`<script>` and `<template>`). ^

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. Learn more.	
<input checked="" type="radio"/> List items (<code></code>) are contained within <code></code> or <code></code> parent elements ^	
Screen readers require list items (<code></code>) to be contained within a parent <code></code> or <code></code> to be announced properly. Learn more.	
<input checked="" type="radio"/> Heading elements appear in a sequentially-descending order ^	
Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. Learn more.	

NOT APPLICABLE (32)

Hide

<input type="radio"/> <code>[accesskey]</code> values are unique ^	
Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. Learn more.	
<input type="radio"/> <code>[aria-*)</code> attributes match their roles ^	
Each ARIA <code>role</code> supports a specific subset of <code>aria-*</code> attributes. Mismatching these invalidates the <code>aria-*</code> attributes. Learn more.	
<input type="radio"/> <code>button</code> , <code>link</code> , and <code>menuitem</code> elements have accessible names ^	
When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more.	
<input type="radio"/> <code>[aria-hidden="true"]</code> elements do not contain focusable descendents ^	
Focusable descendents within an <code>[aria-hidden="true"]</code> element prevent those interactive elements from being available to users of assistive technologies like screen readers. Learn more.	
<input type="radio"/> ARIA input fields have accessible names ^	
When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. Learn more.	

ARIA `meter` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `progressbar` elements have accessible names ^

When a `progressbar` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

`[role]`s have all required `[aria-*`] attributes ^

Some ARIA roles have required attributes that describe the state of the element to screen readers. [Learn more.](#)

Elements with an ARIA `[role]` that require children to contain a specific `[role]` have all required children. ^

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. [Learn more.](#)

`[role]`s are contained by their required parent element ^

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more.](#)

`[role]` values are valid ^

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more.](#)

ARIA toggle fields have accessible names ^

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `tooltip` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `treeitem` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic

<p>name, making it unusable for users who rely on screen readers. Learn more.</p>	
<p><input type="radio"/> [aria-*] attributes have valid values ^</p>	
<p>Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. Learn more.</p>	
<p><input type="radio"/> [aria-*] attributes are valid and not misspelled ^</p>	
<p>Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. Learn more.</p>	
<p><input type="radio"/> Buttons have an accessible name ^</p>	
<p>When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. Learn more.</p>	
<p><input type="radio"/> <dl>'s contain only properly-ordered <dt> and <dd> groups, <script>, <template> or <div> elements. ^</p>	
<p>When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. Learn more.</p>	
<p><input type="radio"/> Definition list items are wrapped in <dl> elements ^</p>	
<p>Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. Learn more.</p>	
<p><input type="radio"/> [id] attributes on active, focusable elements are unique ^</p>	
<p>All focusable elements must have a unique `id` to ensure that they're visible to assistive technologies. Learn more.</p>	
<p><input type="radio"/> ARIA IDs are unique ^</p>	
<p>The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. Learn more.</p>	
<p><input type="radio"/> No form fields have multiple labels ^</p>	
<p>Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. Learn more.</p>	
<p><input type="radio"/> <frame> or <iframe> elements have a title ^</p>	

Screen reader users rely on frame titles to describe the contents of frames. [Learn more.](#)

`<input type="image">` elements have `[alt]` text ^

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. [Learn more.](#)

Form elements have associated labels ^

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. [Learn more.](#)

The document does not use `<meta http-equiv="refresh">` ^

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. [Learn more.](#)

`<object>` elements have alternate text ^

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. [Learn more.](#)

No element has a `[tabindex]` value greater than 0 ^

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. [Learn more.](#)

Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table. ^

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. [Learn more.](#)

`<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe. ^

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. [Learn more.](#)

`[lang]` attributes have a valid value ^

Specifying a valid [BCP 47 language](#) on elements helps ensure that text is pronounced correctly by a screen reader. [Learn more.](#)

○ `<video>` elements contain a `<track>` element with `[kind="captions"]` ^

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. [Learn more](#).



Best Practices

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks ^

A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. [Learn more](#)

Description	Directive	Severity
No CSP found in enforcement mode		High

GENERAL

○ Detected JavaScript libraries ^

All front-end JavaScript libraries detected on the page. [Learn more](#).

Name	Version
D3	7.3.0

▲ Missing source maps for large first-party JavaScript ^

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. [Learn more](#).

URL	Map URL
...dist/d3.js (127.0.0.1)	
Large JavaScript file is missing a source map	

PASSED AUDITS (12)

Hide

 Uses HTTPS 

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more](#).

 Avoids requesting the geolocation permission on page load 

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more](#).

 Avoids requesting the notification permission on page load 

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more](#).

 Avoids front-end JavaScript libraries with known security vulnerabilities 

Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. [Learn more](#).

 Allows users to paste into password fields 

Preventing password pasting undermines good security policy. [Learn more](#).



 Displays images with correct aspect ratio 

Image display dimensions should match natural aspect ratio. [Learn more.](#)

● Serves images with appropriate resolution ^

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn more.](#)

● Page has the HTML doctype ^

Specifying a doctype prevents the browser from switching to quirks-mode. [Learn more.](#)

● Properly defines charset ^

A character encoding declaration is required. It can be done with a `` tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. [Learn more.](#)

● Avoids deprecated APIs ^

Deprecated APIs will eventually be removed from the browser. [Learn more.](#)

● No browser errors logged to the console ^

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. [Learn more](#)

● No issues in the [Issues](#) panel in Chrome Devtools ^

Issues logged to the `Issues` panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

NOT APPLICABLE (1)

Hide

○ Fonts with `font-display: optional` are preloaded ^

Preload `optional` fonts so first-time visitors may use them. [Learn more](#)



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more](#).

ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Hide

Structured data is valid ^

Run the [Structured Data Testing Tool](#) and the [Structured Data Linter](#) to validate structured data. [Learn more](#).

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (10)

Hide

Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more](#).

Document has a meta description ^

Meta descriptions may be included in search results to concisely summarize page content. [Learn more](#).

● Page has successful HTTP status code ^

Pages with unsuccessful HTTP status codes may not be indexed properly. [Learn more.](#)

● Links have descriptive text ^

Descriptive link text helps search engines understand your content. [Learn more.](#)

● Links are crawlable ^

Search engines may use `href` attributes on links to crawl websites. Ensure that the `href` attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. [Learn More](#)

● Page isn't blocked from indexing ^

Search engines are unable to include your pages in search results if they don't have permission to crawl them. [Learn more.](#)

● Image elements have [alt] attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

● Document has a valid hreflang ^

hreflang links tell search engines what version of a page they should list in search results for a given language or region. [Learn more.](#)

● Document avoids plugins ^

Search engines can't index plugin content, and many devices restrict plugins or don't support them. [Learn more.](#)

NOT APPLICABLE (4)

Hide

○ robots.txt is valid ^

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. [Learn more.](#)

○ Document has a valid rel=canonical ^

Canonical links suggest which URL to show in search results. [Learn more.](#)

○ Document uses legible font sizes ^

Font sizes less than 12px are too small to be legible and require mobile visitors to “pinch to zoom” in order to read. Strive to have >60% of page text ≥12px. [Learn more.](#)

○ Tap targets are sized appropriately ^

Interactive elements like buttons and links should be large enough (48x48px), and have enough space around them, to be easy enough to tap without overlapping onto other elements. [Learn more.](#)



PWA

These checks validate the aspects of a Progressive Web App. [Learn more.](#)

⊕ INSTALLABLE

▲ Web app manifest or service worker do not meet the installability requirements — 1 reason ^

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more.](#)

Failure reason

Page has no manifest <link> URL

★ PWA OPTIMIZED

▲ Does not register a service worker that controls page and `start_url` ^

The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. [Learn more](#).

▲ Is not configured for a custom splash screen **Failures: No manifest was fetched.** ^

A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. [Learn more](#).

▲ Does not set a theme color for the address bar.
Failures: No manifest was fetched, No ``<meta name="theme-color">`` tag found. ^

The browser address bar can be themed to match your site. [Learn more](#).

○ Content is sized correctly for the viewport ^

If the width of your app's content doesn't match the width of the viewport, your app might not be optimized for mobile screens. [Learn more](#).

● Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

▲ Does not provide a valid `apple-touch-icon` ^

For ideal appearance on iOS when users add a progressive web app to the home screen, define an `apple-touch-icon`. It must point to a non-transparent 192px (or 180px) square PNG. [Learn More](#).

▲ Manifest doesn't have a maskable icon **No manifest was fetched** ^

A maskable icon ensures that the image fills the entire shape without being letterboxed when installing the app on a device. [Learn more](#).

ADDITIONAL ITEMS TO MANUALLY CHECK (3)

Hide

 Site works cross-browser ^

To reach the most number of users, sites should work across every major browser. [Learn more](#).

 Page transitions don't feel like they block on the network ^

Transitions should feel snappy as you tap around, even on a slow network. This experience is key to a user's perception of performance. [Learn more](#).


 Each page has a URL ^

Ensure individual pages are deep linkable via URL and that URLs are unique for the purpose of shareability on social media. [Learn more](#).


These checks are required by the baseline [PWA Checklist](#) but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.

 Captured at Dec 27, 2022, 1:05

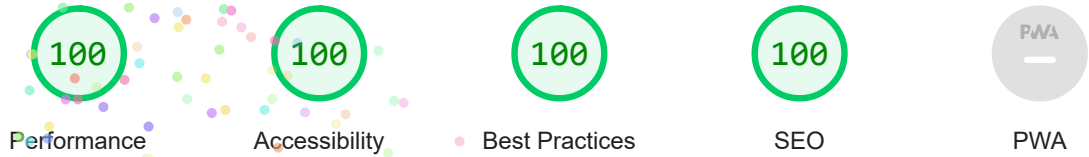
PM GMT+1

 Initial page load Emulated Desktop with Lighthouse

9.6.6

 Custom throttling Single page load Using Chromium 108.0.0.0 with devtoolsGenerated by **Lighthouse** 9.6.6 | [File an issue](#)

A.2.3. Página de gestión de la red de estaciones



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)



Option: Stations data
 Load the station network .csv:
 Uploaded file should meet the .csv format and should have at least the following columns [#StationName, 'Latitude', 'Longitude', 'ReceiverName', 'ReceiverSatelliteSystem'] with unique '#StationName', and 'Latitude', 'Longitude' expressed in degrees and using ',' separator.
 Upload the stations network file:
 Ninguno archivo seleccionado
 Station .csv file loaded successfully
 Heatmap data for the stations network available

#StationName	Latitude	Longitude	ReceiverName	ReceiverSatelliteSystem
0 AHC00AUS	23.67	133.886	SEPT POLARIS	GPS+QZSS+GLO+GAL+BDS
1 BOGT00COL	4.64	-74.081	JAWAD TRE_3 DELTA	GPS+GLO+GAL+BDS
2 CEDU00AUS	-31.887	133.81	SEPT POLARIS	GPS+GLO+GAL+BDS+QZSS+IRNSS
3 CKIS00CCK	21.201	159.801	TRIMBLE NETR9	GPS+GLO+GAL+BDS+QZSS
4 PLAV00ATA	68.477	77.073	SEPT POLARIS	GPS+GLO+GAL+BDS+QZSS

METRICS

Expand view

● First Contentful Paint
0.5 s

● Time to Interactive
0.5 s

● Speed Index
0.5 s

● Total Blocking Time
0 ms

● Largest Contentful Paint
0.5 s

● Cumulative Layout Shift
0

[View Treemap](#)



Show audits relevant to: **All** [FCP](#) [TBT](#) [LCP](#) [CLS](#)

OPPORTUNITIES

Opportunity

Estimated Savings

■ Eliminate render-blocking resources **0.32 s** ^

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more](#). FCP LCP

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
...css/Main.css (127.0.0.1)	0.8 KiB	70 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	270 ms
...dist/d3.js (127.0.0.1)	136.0 KiB	310 ms

Reduce unused JavaScript

0.16 s ^

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn more](#). LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	136.0 KiB	113.5 KiB

These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

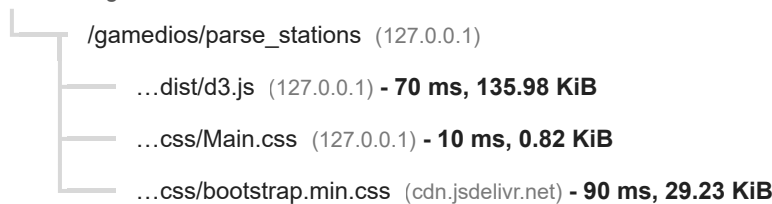
DIAGNOSTICS

Avoid chaining critical requests — 3 chains found ^

The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. [Learn more](#). FCP LCP

Maximum critical path latency: **140 ms**

Initial Navigation



Keep request counts low and transfer sizes small — 5 requests • 197 KiB ^

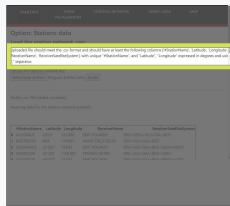
To set budgets for the quantity and size of page resources, add a budget.json file. [Learn more](#).

Resource Type	Requests	Transfer Size
Total	5	197.0 KiB
Script	1	136.0 KiB
Stylesheet	2	30.0 KiB
Other	1	28.0 KiB
Document	1	3.0 KiB
Image	0	0.0 KiB
Media	0	0.0 KiB
Font	0	0.0 KiB
Third-party	1	29.2 KiB

○ Largest Contentful Paint element — 1 element found ^

This is the largest contentful element painted within the viewport. [Learn More](#) LCP

Element



Uploaded file should meet the .csv format and should have at least the followin...

<p>

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (35)

Hide

● Properly size images ^

Serve images that are appropriately-sized to save cellular data and improve load time. [Learn more](#).

● Defer offscreen images ^

Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. [Learn more](#).

● Minify CSS ^

Minifying CSS files can reduce network payload sizes. [Learn more.](#) FCP LCP

● Minify JavaScript — Potential savings of 41 KiB ^

Minifying JavaScript files can reduce payload sizes and script parse time. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	136.0 KiB	40.6 KiB

● Reduce unused CSS — Potential savings of 29 KiB ^

Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	28.7 KiB

● Efficiently encode images ^

Optimized images load faster and consume less cellular data. [Learn more.](#)

● Serve images in next-gen formats ^

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn more.](#)

● Enable text compression ^

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. [Learn more.](#) FCP LCP

● Preconnect to required origins ^

Consider adding `preconnect` or `dns-prefetch` resource hints to establish early connections to important third-party origins. [Learn more.](#) FCP LCP

● Initial server response time was short — Root document took 30 ms ^

Keep the server response time for the main document short because all other requests depend on it. [Learn more](#). FCP LCP

URL	Time Spent
/gamedios/parse_stations (127.0.0.1)	30 ms

● Avoid multiple page redirects ^

Redirects introduce additional delays before the page can be loaded. [Learn more](#). FCP LCP

○ Preload key requests ^

Consider using ``<link rel=preload>`` to prioritize fetching resources that are currently requested later in page load. [Learn more](#). FCP LCP

● Use HTTP/2 ^

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. [Learn more](#).

● Use video formats for animated content ^

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. [Learn more](#) LCP

● Remove duplicate modules in JavaScript bundles ^

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. TBT

● Avoid serving legacy JavaScript to modern browsers ^

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script

deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. [Learn More](#) TBT

● Preload Largest Contentful Paint image ^

Preload the image used by the LCP element in order to improve your LCP time. [Learn more.](#) LCP

● Avoids enormous network payloads — Total size was 197 KiB ^

Large network payloads cost users real money and are highly correlated with long load times. [Learn more.](#) LCP

Show 3rd-party resources (1)

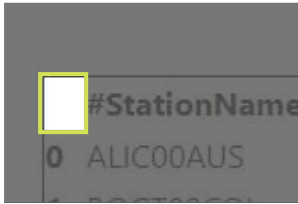
URL	Transfer Size
...dist/d3.js (127.0.0.1)	136.0 KiB
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB
...images/icon192.png (127.0.0.1)	28.0 KiB
/gamedios/parse_stations (127.0.0.1)	3.0 KiB
...css/Main.css (127.0.0.1)	0.8 KiB


● Uses efficient cache policy on static assets — 0 resources found ^

A long cache lifetime can speed up repeat visits to your page. [Learn more.](#)

● Avoids an excessive DOM size — 449 elements ^

A large DOM will increase memory usage, cause longer [style calculations](#), and produce costly [layout reflows](#). [Learn more.](#) TBT

Statistic	Element	Value
Total DOM Elements		449
Maximum DOM Depth	 <pre>table.dataframe > thead > tr > th <th></pre>	7

Statistic	Element	Value
Maximum Child Elements	 <pre>0 ALIC00AUS -23.67 133.886 SEPT POLARX5 GPS+QZSS+GLO+GAL+BDS 1 BOGT00COL 4.64 -... <tbody></pre>	58

User Timing marks and measures ^

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. [Learn more.](#)

JavaScript execution time — 0.0 s ^

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

URL	Total CPU Time	Script Evaluation	Script Parse
/gamedios/parse_stations (127.0.0.1)	97 ms	2 ms	0 ms

Minimizes main-thread work — 0.2 s ^

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more](#) TBT

Category	Time Spent
Style & Layout	79 ms
Other	38 ms
Script Evaluation	11 ms
Script Parsing & Compilation	10 ms
Parse HTML & CSS	7 ms
Rendering	6 ms

All text remains visible during webfont loads ^

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading. [Learn more.](#) FCP LCP

Minimize third-party usage — **Third-party code blocked the main thread for 0 ms** ^

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading.

[Learn more.](#) TBT

Third-Party	Transfer Size	Main-Thread Blocking Time
JSDelivr CDN	29 KiB	0 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	29 KiB	0 ms

Lazy load third-party resources with facades ^

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. [Learn more.](#) TBT

Largest Contentful Paint image was not lazily loaded ^

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. [Learn more.](#)

Avoid large layout shifts ^

These DOM elements contribute most to the CLS of the page. CLS

Uses passive listeners to improve scrolling performance ^

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. [Learn more.](#)

Avoids `document.write()` ^

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. [Learn more.](#)

Avoid long main-thread tasks ^

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay.

[Learn more](#) TBT

Avoid non-composited animations ^

Animations which are not composited can be janky and increase CLS. [Learn more](#) CLS

Image elements have explicit `width` and `height` ^

Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn more](#) CLS

Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

Avoids `unload` event listeners ^

The `unload` event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use `pagehide` or `visibilitychange` events instead. [Learn more](#)



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

The page has a logical tab order ^

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more](#).

- Interactive controls are keyboard focusable ^

Custom interactive controls are keyboard focusable and display a focus indicator. [Learn more.](#)
- Interactive elements indicate their purpose and state ^

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn more.](#)
- The user's focus is directed to new content added to the page ^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn more.](#)
- User focus is not accidentally trapped in a region ^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn more.](#)
- Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more.](#)
- Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn more.](#)
- Visual order on the page follows DOM order ^

DOM order matches the visual order, improving navigation for assistive technology. [Learn more.](#)
- Offscreen content is hidden from assistive technology ^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn more.](#)
- HTML5 landmark elements are used to improve navigation ^

Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more.](#)

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review.](#)

● [aria-*] attributes match their roles ^

Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. [Learn more.](#)

● [aria-hidden="true"] is not present on the document <body> ^

Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document ``<body>``. [Learn more.](#)

● [aria-*] attributes have valid values ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. [Learn more.](#)

● [aria-*] attributes are valid and not misspelled ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. [Learn more.](#)

● ARIA IDs are unique ^

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. [Learn more.](#)

● Form elements have associated labels ^

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. [Learn more.](#)

● [user-scalable="no"] is not used in the <meta name="viewport"> element and the [maximum-scale] attribute is not less than 5. ^

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. [Learn more.](#)

- The page contains a heading, skip link, or landmark region ^

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. [Learn more.](#)

- Background and foreground colors have a sufficient contrast ratio ^

Low-contrast text is difficult or impossible for many users to read. [Learn more.](#)

- Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

- `<html>` element has a `[lang]` attribute ^

If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. [Learn more.](#)

- `<html>` element has a valid value for its `[lang]` attribute ^

Specifying a valid [BCP 47 language](#) helps screen readers announce text properly. [Learn more.](#)

- Links have a discernible name ^

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn more.](#)

- Lists contain only `` elements and script supporting elements (`<script>` and `<template>`). ^

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. [Learn more.](#)

- List items (``) are contained within `` or `` parent elements ^

Screen readers require list items (``) to be contained within a parent `` or `` to be announced properly. [Learn more.](#)

- Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table. ^

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. [Learn more.](#)

- `<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe. ^

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. [Learn more.](#)

- Heading elements appear in a sequentially-descending order ^

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. [Learn more.](#)

NOT APPLICABLE (26)

Hide

- `[accesskey]` values are unique ^

Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. [Learn more.](#)

- `button`, `link`, and `menuitem` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

- `[aria-hidden="true"]` elements do not contain focusable descendents ^

Focusable descendents within an `[aria-hidden="true"]` element prevent those interactive elements from being available to users of assistive technologies like screen readers. [Learn more.](#)

- ARIA input fields have accessible names ^

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `meter` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `progressbar` elements have accessible names ^

When a `progressbar` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

`[role]`s have all required `[aria-*`] attributes ^

Some ARIA roles have required attributes that describe the state of the element to screen readers. [Learn more.](#)

Elements with an ARIA `[role]` that require children to contain a specific `[role]` have all required children. ^

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. [Learn more.](#)

`[role]`s are contained by their required parent element ^

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more.](#)

`[role]` values are valid ^

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more.](#)

ARIA toggle fields have accessible names ^

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `tooltip` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `treeitem` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

Buttons have an accessible name ^

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. [Learn more.](#)

`<dl>`'s contain only properly-ordered `<dt>` and `<dd>` groups, `<script>`, `<template>` or `<div>` elements. ^

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn more.](#)

Definition list items are wrapped in `<dl>` elements ^

Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. [Learn more.](#)

`[id]` attributes on active, focusable elements are unique ^

All focusable elements must have a unique `id` to ensure that they're visible to assistive technologies. [Learn more.](#)

No form fields have multiple labels ^

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn more.](#)

`<frame>` or `<iframe>` elements have a title ^

Screen reader users rely on frame titles to describe the contents of frames. [Learn more.](#)

Image elements have `[alt]` attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

`<input type="image">` elements have `[alt]` text ^

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. [Learn more.](#)

- The document does not use `<meta http-equiv="refresh">` ^

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. [Learn more.](#)

- `<object>` elements have alternate text ^

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. [Learn more.](#)

- No element has a `[tabindex]` value greater than 0 ^

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. [Learn more.](#)

- `[lang]` attributes have a valid value ^

Specifying a valid [BCP 47 language](#) on elements helps ensure that text is pronounced correctly by a screen reader. [Learn more.](#)

- `<video>` elements contain a `<track>` element with `[kind="captions"]` ^

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. [Learn more.](#)



Best Practices

TRUST AND SAFETY

- Ensure CSP is effective against XSS attacks ^

A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. [Learn more](#)

Description	Directive	Severity
No CSP found in enforcement mode		High

GENERAL

○ Detected JavaScript libraries ^

All front-end JavaScript libraries detected on the page. [Learn more.](#)

Name	Version
D3	7.3.0

▲ Missing source maps for large first-party JavaScript ^

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. [Learn more.](#)

URL	Map URL
...dist/d3.js (127.0.0.1)	
Large JavaScript file is missing a source map	

PASSED AUDITS (12)

Hide

● Uses HTTPS ^

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more.](#)

● Avoids requesting the geolocation permission on page load ^

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more.](#)

● Avoids requesting the notification permission on page load ^

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more](#).

● Avoids front-end JavaScript libraries with known security vulnerabilities ^

Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. [Learn more](#).

● Allows users to paste into password fields ^

Preventing password pasting undermines good security policy. [Learn more](#).

● Displays images with correct aspect ratio ^

Image display dimensions should match natural aspect ratio. [Learn more](#).

● Serves images with appropriate resolution ^

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn more](#).

● Page has the HTML doctype ^

Specifying a doctype prevents the browser from switching to quirks-mode. [Learn more](#).

● Properly defines charset ^

A character encoding declaration is required. It can be done with a ``<meta>`` tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. [Learn more](#).

● Avoids deprecated APIs ^

Deprecated APIs will eventually be removed from the browser. [Learn more](#).

● No browser errors logged to the console ^

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. [Learn more](#)

No issues in the **Issues** panel in Chrome Devtools ^

Issues logged to the `Issues` panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

NOT APPLICABLE (1) Hide

Fonts with **font-display: optional** are preloaded ^

Preload `optional` fonts so first-time visitors may use them. [Learn more](#)



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more](#).

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Hide

Structured data is valid ^

Run the [Structured Data Testing Tool](#) and the [Structured Data Linter](#) to validate structured data. [Learn more](#).

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (9)

Hide

● Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

● Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more](#).

● Document has a meta description ^

Meta descriptions may be included in search results to concisely summarize page content. [Learn more](#).

● Page has successful HTTP status code ^

Pages with unsuccessful HTTP status codes may not be indexed properly. [Learn more](#).

● Links have descriptive text ^

Descriptive link text helps search engines understand your content. [Learn more](#).

● Links are crawlable ^

Search engines may use `href` attributes on links to crawl websites. Ensure that the `href` attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. [Learn More](#)

● Page isn't blocked from indexing ^

Search engines are unable to include your pages in search results if they don't have permission to crawl them. [Learn more](#).

● Document has a valid `hreflang` ^

hreflang links tell search engines what version of a page they should list in search results for a given language or region. [Learn more](#).

Document avoids plugins ^

Search engines can't index plugin content, and many devices restrict plugins or don't support them. [Learn more](#).

NOT APPLICABLE (5)

Hide

robots.txt is valid ^

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. [Learn more](#).

Image elements have [alt] attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more](#).

Document has a valid rel=canonical ^

Canonical links suggest which URL to show in search results. [Learn more](#).

Document uses legible font sizes ^

Font sizes less than 12px are too small to be legible and require mobile visitors to “pinch to zoom” in order to read. Strive to have >60% of page text ≥12px. [Learn more](#).

Tap targets are sized appropriately ^

Interactive elements like buttons and links should be large enough (48x48px), and have enough space around them, to be easy enough to tap without overlapping onto other elements. [Learn more](#).



PWA

These checks validate the aspects of a Progressive Web App. [Learn more.](#)

INSTALLABLE



-  Web app manifest or service worker do not meet the installability requirements — 1 reason 

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more.](#)

Failure reason

Page has no manifest <link> URL



PWA OPTIMIZED

-  Does not register a service worker that controls page and `start_url` 

The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. [Learn more.](#)

-  Is not configured for a custom splash screen **Failures: No manifest was fetched.** 


A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. [Learn more.](#)

-  Does not set a theme color for the address bar. 
Failures: No manifest was fetched, No `<meta name="theme-color">` tag found.

The browser address bar can be themed to match your site. [Learn more.](#)

- Content is sized correctly for the viewport 

If the width of your app's content doesn't match the width of the viewport, your app might not be optimized for mobile screens. [Learn more.](#)

- Has a `<meta name="viewport">` tag with `width` or `initial-scale` 

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

▲ Does not provide a valid `apple-touch-icon` ^

For ideal appearance on iOS when users add a progressive web app to the home screen, define an `apple-touch-icon`. It must point to a non-transparent 192px (or 180px) square PNG. [Learn More](#).

▲ Manifest doesn't have a maskable icon **No manifest was fetched** ^

A maskable icon ensures that the image fills the entire shape without being letterboxed when installing the app on a device. [Learn more](#).

ADDITIONAL ITEMS TO MANUALLY CHECK (3)

Hide

○ Site works cross-browser ^

To reach the most number of users, sites should work across every major browser. [Learn more](#).

○ Page transitions don't feel like they block on the network ^

Transitions should feel snappy as you tap around, even on a slow network. This experience is key to a user's perception of performance. [Learn more](#).

○ Each page has a URL ^

Ensure individual pages are deep linkable via URL and that URLs are unique for the purpose of shareability on social media. [Learn more](#).

These checks are required by the baseline [PWA Checklist](#) but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.

Captured at Nov 20, 2022, 7:26

PM GMT+1

Initial page load

Emulated Desktop with Lighthouse

9.6.6

Custom throttling

Single page load

Using Chromium 107.0.0.0 with devtools

A.2.4. Página de gestión de los datos orbitales



Performance



Accessibility



Best Practices



SEO



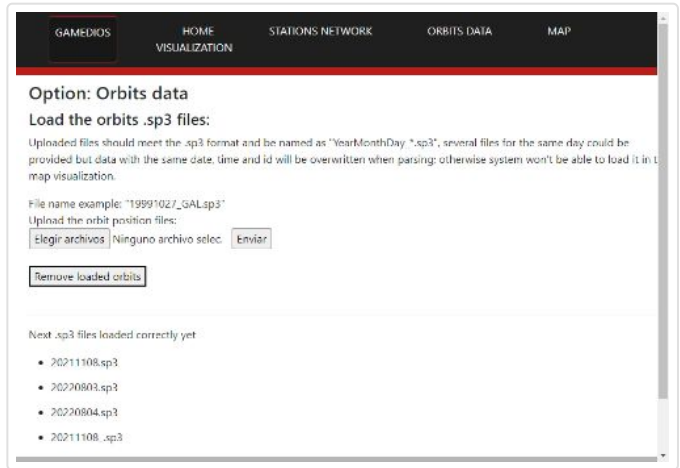
PWA



Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)

- ▲ 0–49
- 50–89
- 90–100



METRICS

Expand view

● First Contentful Paint
0.5 s

● Speed Index
0.5 s

● Largest Contentful Paint
0.5 s

● Time to Interactive
0.5 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0

[View Treemap](#)



Show audits relevant to: **All** [FCP](#) [TBT](#) [LCP](#) [CLS](#)

OPPORTUNITIES

Opportunity

Estimated Savings

■ Eliminate render-blocking resources

0.32 s ^

Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles. [Learn more](#). FCP LCP

Show 3rd-party resources (1)

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	320 ms
...dist/d3.js (127.0.0.1)	136.0 KiB	330 ms

These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

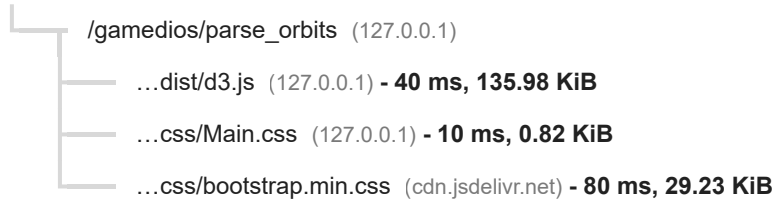
DIAGNOSTICS

Avoid chaining critical requests — 3 chains found ^

The Critical Request Chains below show you what resources are loaded with a high priority. Consider reducing the length of chains, reducing the download size of resources, or deferring the download of unnecessary resources to improve page load. [Learn more](#). FCP LCP

Maximum critical path latency: **100 ms**

Initial Navigation



Keep request counts low and transfer sizes small — 5 requests • 195 KiB ^

To set budgets for the quantity and size of page resources, add a budget.json file. [Learn more](#).

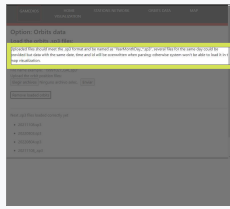
Resource Type	Requests	Transfer Size
Total	5	195.5 KiB
Script	1	136.0 KiB
Stylesheet	2	30.0 KiB
Other	1	28.0 KiB
Document	1	1.4 KiB
Image	0	0.0 KiB

Resource Type	Requests	Transfer Size
Media	0	0.0 KiB
Font	0	0.0 KiB
Third-party	1	29.2 KiB

○ Largest Contentful Paint element — 1 element found ^

This is the largest contentful element painted within the viewport. [Learn More](#) LCP

Element



Uploaded files should meet the .sp3 format and be named as "YearMonthDay_*.sp3"...

<p>

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

PASSED AUDITS (36)

Hide

● Properly size images ^

Serve images that are appropriately-sized to save cellular data and improve load time. [Learn more](#).

● Defer offscreen images ^

Consider lazy-loading offscreen and hidden images after all critical resources have finished loading to lower time to interactive. [Learn more](#).

● Minify CSS ^

Minifying CSS files can reduce network payload sizes. [Learn more](#) FCP LCP

● Minify JavaScript — Potential savings of 41 KiB ^

Minifying JavaScript files can reduce payload sizes and script parse time. [Learn more](#) FCP LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	136.0 KiB	40.6 KiB

● Reduce unused CSS — Potential savings of 29 KiB ^

Reduce unused rules from stylesheets and defer CSS not used for above-the-fold content to decrease bytes consumed by network activity. [Learn more.](#) FCP LCP

URL	Transfer Size	Potential Savings
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB	28.7 KiB

● Reduce unused JavaScript — Potential savings of 114 KiB ^

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. [Learn more.](#) LCP

URL	Transfer Size	Potential Savings
...dist/d3.js (127.0.0.1)	136.0 KiB	113.5 KiB

● Efficiently encode images ^

Optimized images load faster and consume less cellular data. [Learn more.](#)

● Serve images in next-gen formats ^

Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn more.](#)

● Enable text compression ^

Text-based resources should be served with compression (gzip, deflate or brotli) to minimize total network bytes. [Learn more.](#) FCP LCP

● Preconnect to required origins ^

Consider adding `preconnect` or `dns-prefetch` resource hints to establish early connections to important third-party origins. [Learn more.](#) FCP LCP

● Initial server response time was short — Root document took 10 ms ^

Keep the server response time for the main document short because all other requests depend on it. [Learn more.](#) FCP LCP

URL	Time Spent
/gamedios/parse_orbits (127.0.0.1)	10 ms

● Avoid multiple page redirects ^

Redirects introduce additional delays before the page can be loaded. [Learn more.](#) FCP LCP

○ Preload key requests ^

Consider using `` to prioritize fetching resources that are currently requested later in page load. [Learn more.](#) FCP LCP

● Use HTTP/2 ^

HTTP/2 offers many benefits over HTTP/1.1, including binary headers and multiplexing. [Learn more.](#)

● Use video formats for animated content ^

Large GIFs are inefficient for delivering animated content. Consider using MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes. [Learn more](#) LCP

● Remove duplicate modules in JavaScript bundles ^

Remove large, duplicate JavaScript modules from bundles to reduce unnecessary bytes consumed by network activity. TBT

● Avoid serving legacy JavaScript to modern browsers ^

Polyfills and transforms enable legacy browsers to use new JavaScript features. However, many aren't necessary for modern browsers. For your bundled JavaScript, adopt a modern script deployment strategy using module/nomodule feature detection to reduce the amount of code shipped to modern browsers, while retaining support for legacy browsers. [Learn More](#) TBT

● Preload Largest Contentful Paint image ^

Preload the image used by the LCP element in order to improve your LCP time. [Learn more.](#) LCP

● Avoids enormous network payloads — Total size was 195 KiB ^

Large network payloads cost users real money and are highly correlated with long load times. [Learn more.](#) LCP

Show 3rd-party resources (1)

URL	Transfer Size
...dist/d3.js (127.0.0.1)	136.0 KiB
...css/bootstrap.min.css (cdn.jsdelivr.net)	29.2 KiB
...images/icon192.png (127.0.0.1)	28.0 KiB
/gamedios/parse_orbits (127.0.0.1)	1.4 KiB
...css/Main.css (127.0.0.1)	0.8 KiB

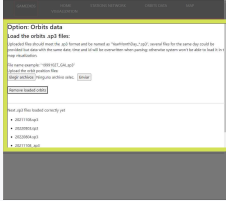
● Uses efficient cache policy on static assets — 0 resources found ^

A long cache lifetime can speed up repeat visits to your page. [Learn more.](#)

● Avoids an excessive DOM size — 37 elements ^

A large DOM will increase memory usage, cause longer [style calculations](#), and produce costly [layout reflows](#). [Learn more.](#) TBT

Statistic	Element	Value
Total DOM Elements		37
Maximum DOM Depth	Leaflet 	6

Statistic	Element	Value
Maximum Child Elements		Option: Orbits data Load the orbits .sp3 files: Uploaded files should meet the... <div id="container">

○ User Timing marks and measures ^

Consider instrumenting your app with the User Timing API to measure your app's real-world performance during key user experiences. [Learn more.](#)

● JavaScript execution time — 0.0 s ^

Consider reducing the time spent parsing, compiling, and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more.](#) TBT

URL	Total CPU Time	Script Evaluation	Script Parse
/gamedios/parse_orbits (127.0.0.1)	66 ms	4 ms	1 ms

● Minimizes main-thread work — 0.1 s ^

Consider reducing the time spent parsing, compiling and executing JS. You may find delivering smaller JS payloads helps with this. [Learn more](#) TBT

Category	Time Spent
Style & Layout	49 ms
Other	42 ms
Script Evaluation	13 ms
Script Parsing & Compilation	10 ms
Parse HTML & CSS	7 ms
Rendering	7 ms

● All text remains visible during webfont loads ^

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading. [Learn more.](#) FCP LCP

Minimize third-party usage — **Third-party code blocked the main thread for 0 ms** ^

Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading.

[Learn more.](#) TBT

Third-Party	Transfer Size	Main-Thread Blocking Time
JSDelivr CDN	29 KiB	0 ms
...css/bootstrap.min.css (cdn.jsdelivr.net)	29 KiB	0 ms

Lazy load third-party resources with facades ^

Some third-party embeds can be lazy loaded. Consider replacing them with a facade until they are required. [Learn more.](#) TBT

Largest Contentful Paint image was not lazily loaded ^

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. [Learn more.](#)

Avoid large layout shifts ^

These DOM elements contribute most to the CLS of the page. CLS

Uses passive listeners to improve scrolling performance ^

Consider marking your touch and wheel event listeners as `passive` to improve your page's scroll performance. [Learn more.](#)

Avoids `document.write()` ^

For users on slow connections, external scripts dynamically injected via `document.write()` can delay page load by tens of seconds. [Learn more.](#)

Avoid long main-thread tasks ^

Lists the longest tasks on the main thread, useful for identifying worst contributors to input delay.

[Learn more](#) TBT

Avoid non-composited animations ^

Animations which are not composited can be janky and increase CLS. [Learn more](#) CLS

Image elements have explicit `width` and `height` ^

Set an explicit width and height on image elements to reduce layout shifts and improve CLS. [Learn more](#) CLS

Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more](#). TBT

Avoids `unload` event listeners ^

The `unload` event does not fire reliably and listening for it can prevent browser optimizations like the Back-Forward Cache. Use `pagehide` or `visibilitychange` events instead. [Learn more](#)



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

The page has a logical tab order ^

Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more](#).

- Interactive controls are keyboard focusable ^

Custom interactive controls are keyboard focusable and display a focus indicator. [Learn more.](#)
- Interactive elements indicate their purpose and state ^

Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn more.](#)
- The user's focus is directed to new content added to the page ^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn more.](#)
- User focus is not accidentally trapped in a region ^

A user can tab into and out of any control or region without accidentally trapping their focus. [Learn more.](#)
- Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more.](#)
- Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn more.](#)
- Visual order on the page follows DOM order ^

DOM order matches the visual order, improving navigation for assistive technology. [Learn more.](#)
- Offscreen content is hidden from assistive technology ^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn more.](#)
- HTML5 landmark elements are used to improve navigation ^

Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more.](#)

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review.](#)

● [aria-*] attributes match their roles ^

Each ARIA `role` supports a specific subset of `aria-*` attributes. Mismatching these invalidates the `aria-*` attributes. [Learn more.](#)

● [aria-hidden="true"] is not present on the document <body> ^

Assistive technologies, like screen readers, work inconsistently when `aria-hidden="true"` is set on the document ``<body>``. [Learn more.](#)

● [aria-*] attributes have valid values ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. [Learn more.](#)

● [aria-*] attributes are valid and not misspelled ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. [Learn more.](#)

● Buttons have an accessible name ^

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. [Learn more.](#)

● ARIA IDs are unique ^

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. [Learn more.](#)

● Form elements have associated labels ^

Labels ensure that form controls are announced properly by assistive technologies, like screen readers. [Learn more.](#)

● [user-scalable="no"] is not used in the <meta name="viewport"> element and the [maximum-

`scale]` attribute is not less than 5.

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. [Learn more.](#)

● The page contains a heading, skip link, or landmark region ^

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. [Learn more.](#)

● Background and foreground colors have a sufficient contrast ratio ^

Low-contrast text is difficult or impossible for many users to read. [Learn more.](#)

● Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

● `<html>` element has a `[lang]` attribute ^

If a page doesn't specify a lang attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. [Learn more.](#)

● `<html>` element has a valid value for its `[lang]` attribute ^

Specifying a valid [BCP 47 language](#) helps screen readers announce text properly. [Learn more.](#)

● Links have a discernible name ^

Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn more.](#)

● Lists contain only `` elements and script supporting elements (`<script>` and `<template>`). ^

Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. [Learn more.](#)

● List items (``) are contained within `` or `` parent elements ^

Screen readers require list items (``) to be contained within a parent `` or `` to be announced properly. [Learn more.](#)

● Heading elements appear in a sequentially-descending order ^

Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. [Learn more.](#)

NOT APPLICABLE (27)

Hide

○ [\[accesskey\]](#) values are unique ^

Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. [Learn more.](#)

○ `button`, `link`, and `menuitem` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

○ `[aria-hidden="true"]` elements do not contain focusable descendents ^

Focusable descendents within an `[aria-hidden="true"]` element prevent those interactive elements from being available to users of assistive technologies like screen readers. [Learn more.](#)

○ ARIA input fields have accessible names ^

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

○ ARIA `meter` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `progressbar` elements have accessible names ^

When a `progressbar` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

`[role]`s have all required `[aria-*`] attributes ^

Some ARIA roles have required attributes that describe the state of the element to screen readers. [Learn more.](#)

Elements with an ARIA `[role]` that require children to contain a specific `[role]` have all required children. ^

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. [Learn more.](#)

`[role]`s are contained by their required parent element ^

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more.](#)

`[role]` values are valid ^

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more.](#)

ARIA toggle fields have accessible names ^

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `tooltip` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

ARIA `treeitem` elements have accessible names ^

When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more.](#)

`<dl>`'s contain only properly-ordered `<dt>` and `<dd>` groups, `<script>`, `<template>` or `<div>` elements. ^

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn more.](#)

Definition list items are wrapped in `<dl>` elements ^

Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<dl>` element to ensure that screen readers can properly announce them. [Learn more.](#)

`[id]` attributes on active, focusable elements are unique ^

All focusable elements must have a unique `id` to ensure that they're visible to assistive technologies. [Learn more.](#)

No form fields have multiple labels ^

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn more.](#)

`<frame>` or `<iframe>` elements have a title ^

Screen reader users rely on frame titles to describe the contents of frames. [Learn more.](#)

Image elements have `[alt]` attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

`<input type="image">` elements have `[alt]` text ^

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. [Learn more.](#)

The document does not use `<meta http-equiv="refresh">` ^

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. [Learn more.](#)

`<object>` elements have alternate text ^

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. [Learn more.](#)

No element has a `[tabindex]` value greater than 0 ^

A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. [Learn more.](#)

- Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table. ^

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. [Learn more.](#)

- `<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe. ^

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. [Learn more.](#)

- `[lang]` attributes have a valid value ^

Specifying a valid [BCP 47 language](#) on elements helps ensure that text is pronounced correctly by a screen reader. [Learn more.](#)

- `<video>` elements contain a `<track>` element with `[kind="captions"]` ^

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. [Learn more.](#)



Best Practices

TRUST AND SAFETY

- Ensure CSP is effective against XSS attacks ^

A strong Content Security Policy (CSP) significantly reduces the risk of cross-site scripting (XSS) attacks. [Learn more](#)

Description	Directive	Severity
No CSP found in enforcement mode		High

GENERAL

○ Detected JavaScript libraries ^

All front-end JavaScript libraries detected on the page. [Learn more](#).

Name	Version
D3	7.3.0

▲ Missing source maps for large first-party JavaScript ^

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. [Learn more](#).

URL	Map URL
...dist/d3.js (127.0.0.1)	
Large JavaScript file is missing a source map	

PASSED AUDITS (12)

Hide

● Uses HTTPS ^

All sites should be protected with HTTPS, even ones that don't handle sensitive data. This includes avoiding [mixed content](#), where some resources are loaded over HTTP despite the initial request being served over HTTPS. HTTPS prevents intruders from tampering with or passively listening in on the communications between your app and your users, and is a prerequisite for HTTP/2 and many new web platform APIs. [Learn more](#).

● Avoids requesting the geolocation permission on page load ^

Users are mistrustful of or confused by sites that request their location without context. Consider tying the request to a user action instead. [Learn more](#).

● Avoids requesting the notification permission on page load ^

Users are mistrustful of or confused by sites that request to send notifications without context. Consider tying the request to user gestures instead. [Learn more](#).

● Avoids front-end JavaScript libraries with known security vulnerabilities ^

Some third-party scripts may contain known security vulnerabilities that are easily identified and exploited by attackers. [Learn more](#).

● Allows users to paste into password fields ^

Preventing password pasting undermines good security policy. [Learn more](#).

● Displays images with correct aspect ratio ^

Image display dimensions should match natural aspect ratio. [Learn more](#).

● Serves images with appropriate resolution ^

Image natural dimensions should be proportional to the display size and the pixel ratio to maximize image clarity. [Learn more](#).

● Page has the HTML doctype ^

Specifying a doctype prevents the browser from switching to quirks-mode. [Learn more](#).

● Properly defines charset ^

A character encoding declaration is required. It can be done with a ``<meta>`` tag in the first 1024 bytes of the HTML or in the Content-Type HTTP response header. [Learn more](#).

● Avoids deprecated APIs ^

Deprecated APIs will eventually be removed from the browser. [Learn more](#).

No browser errors logged to the console ^

Errors logged to the console indicate unresolved problems. They can come from network request failures and other browser concerns. [Learn more](#)

No issues in the [Issues](#) panel in Chrome Devtools ^

Issues logged to the `Issues` panel in Chrome Devtools indicate unresolved problems. They can come from network request failures, insufficient security controls, and other browser concerns. Open up the Issues panel in Chrome DevTools for more details on each issue.

NOT APPLICABLE (1)

Hide

Fonts with `font-display: optional` are preloaded ^

Preload `optional` fonts so first-time visitors may use them. [Learn more](#)



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more](#).

ADDITIONAL ITEMS TO MANUALLY CHECK (1)

Hide

Structured data is valid ^

Run the [Structured Data Testing Tool](#) and the [Structured Data Linter](#) to validate structured data. [Learn more.](#)

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (9)

Hide

● Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more.](#) TBT

● Document has a `<title>` element ^

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more.](#)

● Document has a meta description ^

Meta descriptions may be included in search results to concisely summarize page content. [Learn more.](#)

● Page has successful HTTP status code ^

Pages with unsuccessful HTTP status codes may not be indexed properly. [Learn more.](#)

● Links have descriptive text ^

Descriptive link text helps search engines understand your content. [Learn more.](#)

● Links are crawlable ^

Search engines may use `href` attributes on links to crawl websites. Ensure that the `href` attribute of anchor elements links to an appropriate destination, so more pages of the site can be discovered. [Learn More](#)

● Page isn't blocked from indexing ^

Search engines are unable to include your pages in search results if they don't have permission to crawl them. [Learn more.](#)

Document has a valid `hreflang` ^

hreflang links tell search engines what version of a page they should list in search results for a given language or region. [Learn more.](#)

Document avoids plugins ^

Search engines can't index plugin content, and many devices restrict plugins or don't support them. [Learn more.](#)

NOT APPLICABLE (5) Hide

robots.txt is valid ^

If your robots.txt file is malformed, crawlers may not be able to understand how you want your website to be crawled or indexed. [Learn more.](#)

Image elements have `[alt]` attributes ^

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more.](#)

Document has a valid `rel=canonical` ^

Canonical links suggest which URL to show in search results. [Learn more.](#)

Document uses legible font sizes ^

Font sizes less than 12px are too small to be legible and require mobile visitors to “pinch to zoom” in order to read. Strive to have >60% of page text ≥ 12 px. [Learn more.](#)

Tap targets are sized appropriately ^

Interactive elements like buttons and links should be large enough (48x48px), and have enough space around them, to be easy enough to tap without overlapping onto other elements. [Learn more.](#)



PWA

These checks validate the aspects of a Progressive Web App. [Learn more.](#)

INSTALLABLE


-  Web app manifest or service worker do not meet the installability requirements — 1 reason 

Service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. With proper service worker and manifest implementations, browsers can proactively prompt users to add your app to their homescreen, which can lead to higher engagement. [Learn more.](#)

Failure reason

Page has no manifest <link> URL



PWA OPTIMIZED

-  Does not register a service worker that controls page and `start_url` 

The service worker is the technology that enables your app to use many Progressive Web App features, such as offline, add to homescreen, and push notifications. [Learn more.](#)

-  Is not configured for a custom splash screen **Failures: No manifest was fetched.** 

A themed splash screen ensures a high-quality experience when users launch your app from their homescreens. [Learn more.](#)

-  Does not set a theme color for the address bar. **Failures: No manifest was fetched, No ``<meta name="theme-color">`` tag found.** 

The browser address bar can be themed to match your site. [Learn more.](#)

-  Content is sized correctly for the viewport 

If the width of your app's content doesn't match the width of the viewport, your app might not be optimized for mobile screens. [Learn more.](#)

- Has a `<meta name="viewport">` tag with `width` or `initial-scale` ^

A `<meta name="viewport">` not only optimizes your app for mobile screen sizes, but also prevents [a 300 millisecond delay to user input](#). [Learn more.](#) TBT

- ▲ Does not provide a valid `apple-touch-icon` ^

For ideal appearance on iOS when users add a progressive web app to the home screen, define an `apple-touch-icon`. It must point to a non-transparent 192px (or 180px) square PNG. [Learn More.](#)

- ▲ Manifest doesn't have a maskable icon **No manifest was fetched** ^

A maskable icon ensures that the image fills the entire shape without being letterboxed when installing the app on a device. [Learn more.](#)

ADDITIONAL ITEMS TO MANUALLY CHECK (3)

Hide

- Site works cross-browser ^

To reach the most number of users, sites should work across every major browser. [Learn more.](#)

- Page transitions don't feel like they block on the network ^

Transitions should feel snappy as you tap around, even on a slow network. This experience is key to a user's perception of performance. [Learn more.](#)

- Each page has a URL ^

Ensure individual pages are deep linkable via URL and that URLs are unique for the purpose of shareability on social media. [Learn more.](#)

These checks are required by the baseline [PWA Checklist](#) but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.

🕒 Initial page load

⚙️ Custom throttling

🔍 Using Chromium 107.0.0.0 with
devtools

Generated by **Lighthouse** 9.6.6 | [File an issue](#)