****

**Reporte del proyecto desarrollado “****Lector Web de RSS Feed”**

**Report of the developed project "RSS Feed Web Reader"**

Proyecto que presenta:

**Jorge Arturo Aguilar Solís, Víctor Manuel Lavalle Cantón, Eberth Francisco Mezeta Xool**

*Universidad Autónoma de Yucatán  
Facultad de Matemáticas  
Licenciatura en Ingeniería de Software  
Periférico Norte Tablaje 13615 C.P. 97119, Mérida, Yucatán, México.*

[a17016356@alumnos.uady.mx](mailto:a17016356@alumnos.uady.mx)

[a18016318@alumnos.uady.mx](mailto:a18016318@alumnos.uady.mx)

[a18016312@alumnos.uady.mx](mailto:a18016312@alumnos.uady.mx)

Docente a cargo:

**Dr. Víctor Hugo Menéndez Domínguez**

Mérida, Yucatán, México Junio 2021

# **Resumen**

Un *feed RSS* es una tecnología que ha influenciado a muchas herramientas modernas de internet con las que estamos familiarizados. Aunque para muchas personas es un método anticuado, sigue siendo útil para estar al tanto de todo el contenido nuevo que aparece todos los días en la web. En otras palabras, es una buena herramienta para leer lo que se desee en línea.

Por sus siglas en inglés, RSS significa *Really Simple Syndication* o *Rich Site Summary*. Por lo general, estos lectores, leen los datos en formato XML y se actualizan en automático para entregar el contenido más nuevo directo al dispositivo. Básicamente, este enfoque permite a los usuarios de internet crear sus propios feeds, llenos de actualizaciones personalizadas de los sitios que les gusta visitar con regularidad.

En el presente proyecto se muestra y explica la realización de nuestro propio *lector web de RSS feed*, es decir, las tecnologías utilizadas, su arquitectura, las técnicas de optimización hechas y su evaluación de desempeño.

*-----------------------*

*Palabras claves y frases:* RSS, XML, feed, técnicas de optimización, tecnologías, lector.

*-----------------------*

# **Abstract**

An RSS feed is a technology that has influenced many modern internet tools that we are familiar with. Although for many people it is an outdated method, it is still useful to be aware of all the new content that appears every day on the web. In other words,

it is a good tool to read what you want online.RSS stands for Really Simple Syndication or Rich Site Summary. These readers typically read the data in XML format and automatically update to deliver the newest content direct to the device. Basically this approach allows internet users to create their own feeds, filled with personalized updates from the sites they like to visit regularly.

This project shows and explains the realization of our own RSS feed web reader, that is, the technologies used, its architecture, the optimization techniques used and their performance evaluation.

*-----------------------*

*Keywords and phrases:* RSS, XML, feed, optimization techniques, technologies, reader.

*-----------------------*

Índice general

[Resumen 2](#_Toc75224795)

[Abstract 2](#_Toc75224796)

[1. Introducción. 4](#_Toc75224797)

[**1.1 Objetivo General** 4](#_Toc75224798)

[**1.2 Objetivos específicos** 4](#_Toc75224799)

[**1.3 Optimización de aplicaciones web** 4](#_Toc75224800)

[**1.4 Condiciones de pruebas** 5](#_Toc75224801)

[**1.5 Funcionalidades del sistema** 5](#_Toc75224802)

[**1.5 Interfaces** 6](#_Toc75224803)

[1.5.1 Landing Page 6](#_Toc75224804)

[1.5.2 Añadir link de canal RSS 6](#_Toc75224805)

[1.5.3 Vista de noticias de manera general 7](#_Toc75224806)

[1.5.4 Vista de noticias por fecha 7](#_Toc75224807)

[1.5.5 Búsqueda de noticias (satisfactoria) 8](#_Toc75224808)

[1.5.6 Búsqueda de noticias (sin coincidencias) 8](#_Toc75224809)

[1.5.7 Ordenar noticias 9](#_Toc75224810)

[2. Arquitectura y tecnologías de la aplicación 9](#_Toc75224811)

[**2.1 Lenguajes de programación y de marcado** 9](#_Toc75224812)

[**2.2 Frameworks, bibliotecas y herramientas** 9](#_Toc75224813)

[**2.3 Diagrama de componentes** 10](#_Toc75224814)

[**2.4 Diagrama de base de datos** 10](#_Toc75224815)

[3. Evaluación diagnóstica del desempeño inicial de la aplicación 11](#_Toc75224816)

[**3.1 Página principal** 11](#_Toc75224817)

[**3.2 Consulta** 11](#_Toc75224818)

[**3.3 Ejecuciones de pruebas** 11](#_Toc75224819)

[4. Propuestas de mejora aplicadas. 12](#_Toc75224820)

[**4.1 Lado del cliente** 13](#_Toc75224821)

[**4.2 Lado del servidor** 13](#_Toc75224822)

[5. Contraste de resultados 14](#_Toc75224823)

[**5.1 Resultados Obtenidos de la optimización** 14](#_Toc75224824)

[**5.1.1 Página principal** 14](#_Toc75224825)

[**5.1.2 Consulta** 14](#_Toc75224826)

[**5.2 Comparación de resultados** 15](#_Toc75224827)

[5.2.1 Resultados de Lighthouse 15](#_Toc75224828)

[6. Conclusiones. 18](#_Toc75224829)

[Referencias 19](#_Toc75224830)

# **Introducción.**

El presente documento tiene la finalidad de evidenciar los aprendizajes adquiridos a lo largo del curso donde aplicamos una serie de técnicas adquiridas para la optimización de aplicaciones web.

El sistema en cuestión es un lector de RSS el cual obtiene las noticias de las páginas que el usuario desee agregar a su feed. El sistema indiza las noticias y las almacena en una base de datos. La página incluye un buscador para que los usuarios puedan encontrar rápidamente las noticias que deseen, de igual manera, el sistema brinda la opción de mostrar las noticias por título, descripción, categoría, url o agrupadas por fecha.

El sistema debe ejecutar varios procesos y métodos para hacer sus funcionalidades, además lleva a cabo distintas consultas a la base de datos y debe de desplegar las noticias en un formato atractivo para el usuario, por lo que los tiempos de transferencia y de carga de la página tienden a ser más elevados de lo que se esperaría que fuesen. El objetivo entonces, fue aplicar distintas técnicas de optimización a la página para reducir el tiempo de carga lo más posible y que sea una experiencia grata para el usuario.

## **1.1 Objetivo General**

El objetivo del documento es mostrar de forma gráfica la comparación entre el rendimiento de una aplicación web sin optimización y con optimización de forma que sin esfuerzo se pueda apreciar si existe o no una diferencia significativa en los tiempos de carga y peso de los recursos.

## **1.2 Objetivos específicos**

* Aplicar técnicas de optimización del lado del cliente.
* Aplicar técnicas de optimización del lado del servidor.
* Representar los resultados mediante tablas y gráficas, empleando el uso adecuado de herramientas de evaluación.
* Aplicar los conocimientos obtenidos de la materia, para realizar un análisis a profundidad sobre el proyectado desarrollado.

## **1.3 Optimización de aplicaciones web**

Una de las ventajas de las aplicaciones web es el ahorro de recursos y que su rendimiento depende del navegador que se emplee, las aplicaciones web tienen ciertos problemas de rendimiento y de consumo de recursos. Sin embargo, a lo largo de la evolución de las aplicaciones web se han desarrollado y planteado muchas técnicas para disminuir el tiempo de transferencia. Dichas técnicas de optimización pueden ser divididas en 3 secciones a optimizar:

**• Optimización del lado del cliente**

Esta optimización se realiza con el fin de mejorar el tiempo de trasferencia del lado del cliente, es decir, reducir el tiempo de carga de la página desde la parte que se ejecuta en la computadora del cliente/usuario.

**• Optimización del lado del servidor**

Esta optimización se lleva a cabo para reducir el peso y el tiempo de los archivos a transferir desde el servidor hasta el cliente.

**• Optimización de la base de datos**

Consiste en optimizar las consultas y emplear algunos respaldos en la base de datos.

## **1.4 Condiciones de pruebas**

Para el proceso de pruebas del proyecto es necesario mencionar el entorno en las que estas fueron realizadas:

a. **Equipo de Prueba:** 

Equipo Marca ASUS, Ryzen 5 2500u 2.0 GHz up to 3.6GHz, Windows 10 Home Single Language, 16 GB RAM DDR4, 1TB HDD + 128 GB SSD, 2x2 802.11ac Wireless, Firefox Developer Edition 89.0b13. 

1. **Servidor:** 127.0 0.1 mediante XAMPP v7.3.28 de 64 bits
2. **Base de Datos:** MySQL mediante phpMyAdmin   
    Dirección de acceso: *http//localhost/phpmyadmin/*

## **1.5 Funcionalidades del sistema**

El sistema contará con las siguientes funcionalidades:

* Añadir a la base de datos canales de noticias RSS
* Actualizar la base de datos con noticias nuevas
* Buscar noticias por palabras claves
* Mostrar las noticias por categoría, descripción, url, título y fecha
* Listar un resumen de las noticias cada una con su miniatura y enlace al artículo original.

## **1.5 Interfaces**

## Interfaz de usuario gráfica, Texto, Sitio web Descripción generada automáticamente1.5.1 Landing Page

### Interfaz de usuario gráfica, Texto, Sitio web Descripción generada automáticamente1.5.2 Añadir link de canal RSS

### Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico, Sitio web Descripción generada automáticamente1.5.3 Vista de noticias de manera general

### Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico Descripción generada automáticamente1.5.4 Vista de noticias por fecha

### Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico Descripción generada automáticamente1.5.5 Búsqueda de noticias (satisfactoria)

### 1.5.6 Búsqueda de noticias (sin coincidencias)

Interfaz de usuario gráfica, Texto, Sitio web

Descripción generada automáticamente

### 1.5.7 Ordenar noticias

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

# **Arquitectura y tecnologías de la aplicación**

La aplicación web se desarrolló con una arquitectura básica de Cliente-Servidor.

## **2.1 Lenguajes de programación y de marcado**

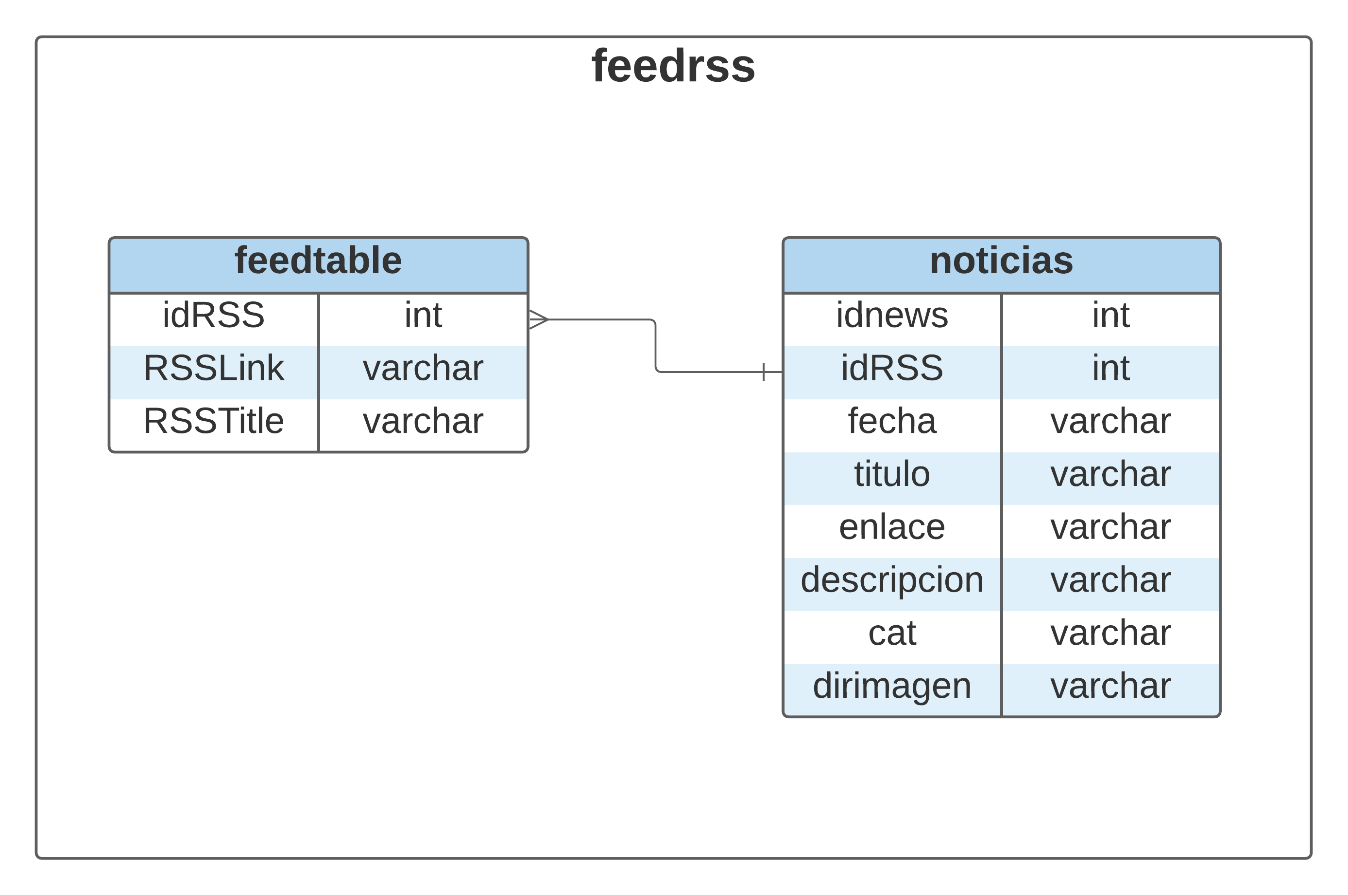
* **HTML5**: Da la estructura del sitio web.
* **CSS**: Establece los estilos de los componentes del html para una visualización agradable de este mismo.
* **Javascript**: Se utilizó para trabajar en el front-end siendo esto la actualización dinámica de los componentes del html y la conexión con el back-end con solicitudes de Ajax
* **PHP**: Se utilizó PHP 7.4.1.2 para el back-end de los microservicios.

## **2.2 Frameworks, bibliotecas y herramientas**

* **Bootstrap**: Librería de CSS que se utilizó para mejorar el aspecto de la interfaz de usuario del sitio web.
* **JQuery**: Se utilizó como complemento para Bootstrap para un mejor manejo de los elementos del HTML al igual que las peticiones AJAX al servidor.
* **Git**: Se utiliza para mantener un historial de las versiones del proyecto.
* **GitHub**: Se utiliza para mantener un repositorio remoto centralizado del proyecto, acelerando así el proceso de desarrollo por parte del equipo.

## Diagrama Descripción generada automáticamente**2.3 Diagrama de componentes**

## **2.4 Diagrama de base de datos**



# **Evaluación diagnóstica del desempeño inicial de la aplicación**

Las pruebas de rendimiento base fueron realizadas utilizando la herramienta JMeter y el análisis de desempeño de Firefox, estas herramientas nos permiten realizar la medición del volumen de datos transferidos en la página web del presente proyecto. Del mismo modo, también nos ayuda probando el rendimiento de la página la primera vez que se accede (sin caché) y las veces en las que se recargue la página (con caché).

## **3.1 Página principal**

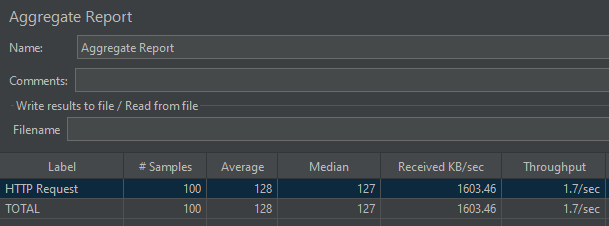
|  |  |  |
| --- | --- | --- |
| **Visita** | **Volumen de transferencia** | **Tiempo de transferencia** |
| Página Principal, primera visita (sin caché) | 13.4mB | 323ms |
| Página Principal, segunda visita (con caché) | 69.4 kB | 191ms |

## **3.2 Consulta**

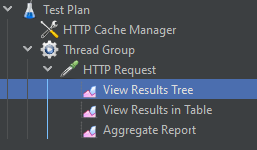
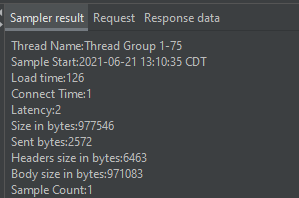
|  |  |  |
| --- | --- | --- |
| Página consulta, primera visita (sin caché) | 84.3kB | 43ms |
| Página consulta, segunda visita (con caché) | 11.3kB | 12ms |

## **3.3 Ejecuciones de pruebas**

Se muestran a continuación capturas de las herramientas con los resultados obtenidos.







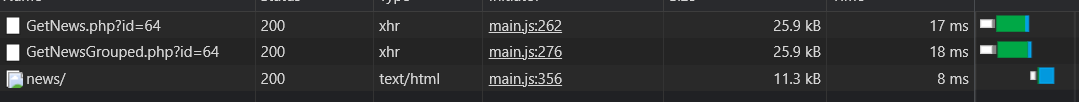
Primera página principal

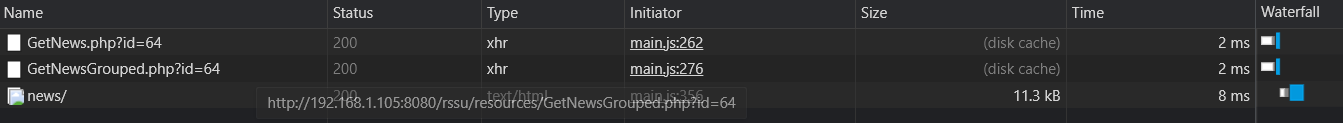


Segunda página principal



Primera consulta



Segunda consulta

# **Propuestas de mejora aplicadas.**

Es necesario explicar alguna de las técnicas antes de listar lo aplicado.

**Compresión**Ayuda a reducir el peso de los archivos, permite generar un código más limpio al eliminar las líneas de código, caracteres y espacios en blanco innecesarios en los archivos. Este proceso sólo afecta el tamaño y la optimización de la página, su implementación no afecta a la legibilidad del código.

**Minimización**Reduce el peso de los archivos de código fuente a través de la eliminación de bytes innecesarios (espacios adicionales, saltos de línea, sangrías y comentarios), también simplifica todo lo posible la escritura de las sentencias (por ejemplo, usando nombres de variables y funciones más cortos, o evitando lo más posible la repetición). Como resultado queda un código difícil de leer, pero que es más compacto y por ende se descarga más rápido desde el servidor.

**Refactorización**Es la modificación del código fuente sin cambiar su comportamiento, esto es con la intención de mejorar su consistencia interna y mejorar la facilidad de comprensión del código o cambiar su estructura y diseño y eliminar código muerto, para facilitar el mantenimiento en el futuro.

## **4.1 Lado del cliente**

Para las mejoras de esta parte del programa se realizaron las siguientes:

* HTML
  + Minimización del archivo HTML.
* JS
  + Minimización del código JS.
  + Refactorización de código.
  + Eliminación de comentarios.
  + Reducción de peticiones Ajax.
* CSS
  + Minimización del archivo CSS.
  + Eliminación de código duplicado y redundante.
* Para Bootstrap y JQuery
  + Eliminar comentarios de los archivos originales que se encontraban minificados.
* Imágenes
  + Compresión de imágenes del sistema.

## **4.2 Lado del servidor**

Para las mejoras de esta parte del programa se realizaron las siguientes:

* Archivos PHP
  + Refactors mínimos para mejorar flujo de datos.
* Mejoras con relación al servidor Apache
  + Se agregó la configuración para habilitar el *mod\_deflate,* este sirve para comprimir algunos archivos antes de enviarlos fuera del servidor.
  + Se habilitó la compresión Gzip mediante zlib en el archivo php.ini.
  + Se habilitó la configuración *mod\_expires* en la que establecimos los valores de cuánto tiempo los archivos serán almacenados por el navegador.
* Mejoras con respecto a la base de datos
  + Se rediseñaron las consultas para optimizar el tiempo de transferencia de estas.

# **Contraste de resultados**

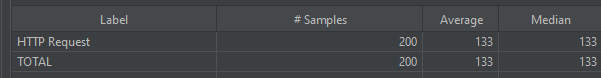
## **5.1 Resultados Obtenidos de la optimización**

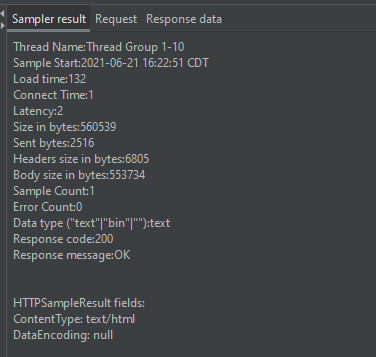
### **5.1.1 Página principal**

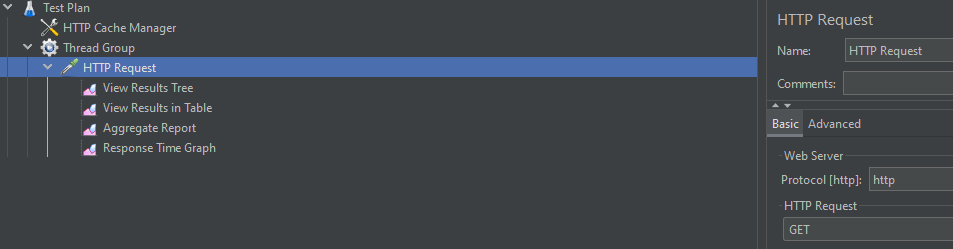
|  |  |  |
| --- | --- | --- |
| **Visita** | **Volumen de transferencia** | **Tiempo de transferencia** |
| Página Principal, primera visita (sin caché) | 568kB | 282ms |
| Página Principal, segunda visita (con caché) | 19.8kB | 178ms |

### **5.1.2 Consulta**

|  |  |  |
| --- | --- | --- |
| Página consulta, primera visita (sin caché) | 43.8kB | 32ms |
| Página consulta, segunda visita (con caché) | 11.3kB | 13ms |





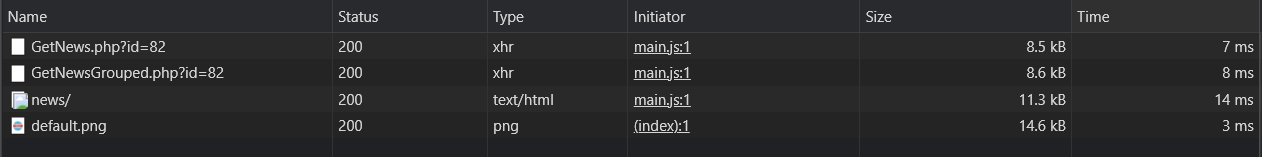


Primera visita a página principal:



Segunda visita a página principal:

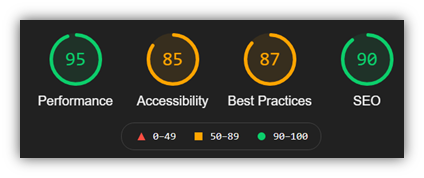


Primera consulta:

Segunda consulta:

## **5.2 Comparación de resultados**

### 5.2.1 Resultados de Lighthouse

 Antes de optimizar:

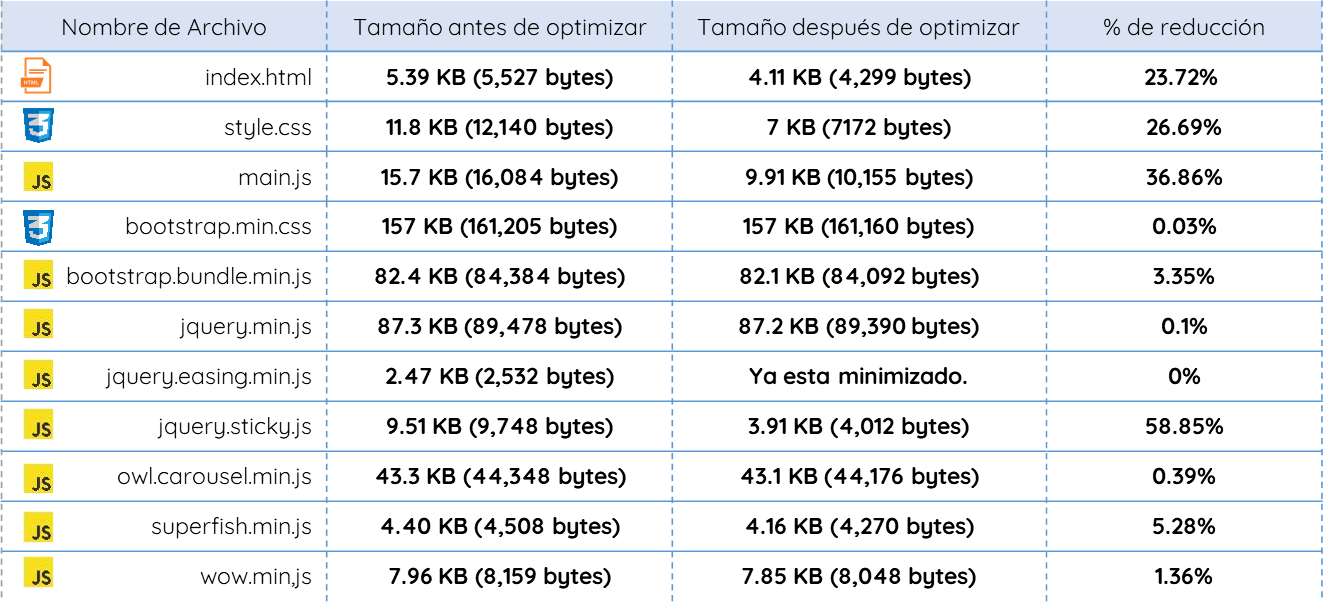
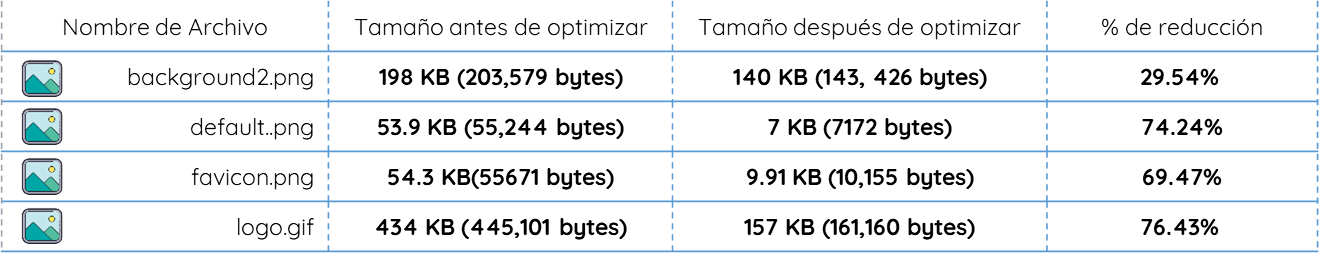
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteDespués de optimizar:

*Para más detalle se anexa el enlace a los reportes en formato HTML de la herramienta:*

[*https://github.com/The-WebOnes/OAW/tree/main/docs/lightHouse-reporrts*](https://github.com/The-WebOnes/OAW/tree/main/docs/lightHouse-reporrts)

5.2.2 Comparación de métricas

5.2.3 Comparación de tamaño de archivos

|  |  |  |
| --- | --- | --- |
| Tamaño Original de la Vista | Tamaño después de optimizar | % de reducción |
| **1.5MB (1,480,087 bytes)** | **914kB(913,770 bytes)** | **38.27%** |

5.2.4 Comparación de carga de la página

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SIN CACHÉ** | **Solicitudes** | **Volumen de Transferencia** | **Tiempo de Transferencia** | **Carga del contenido del DOM** | **Carga completa de la página** |
| Sin Optimizar | 69 | 13.4 MB | 322 ms | 160 ms | 316 ms |
| Optimizado | 24 | 568 kB | 282 ms | 171 ms | 214 ms |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CON CACHÉ** | **Solicitudes** | **Volumen de Transferencia** | **Tiempo de Transferencia** | **Carga del contenido del DOM** | **Carga completa de la página** |
| Sin Optimizar | 69 | 69.4 kB | 191 ms | 140 ms | 185 ms |
| Optimizado | 24 | 19.8 kB | 178 ms | 150 ms | 172 ms |

# **Conclusiones.**

Podemos concluir que las técnicas de optimización sean para cliente, servidor o base de datos, aportan mejoras a un proyecto. Pero hay que saber identificar que técnicas que pruebas nos dan verdaderamente un beneficio ya que para cualquiera tienen puntos tanto negativos como positivos. En la vida laboral no solo se trata de que el producto sea funcional, sino que igual este tiene que cumplir con múltiples estándares de calidad para que sea útil y capaz de resolver las tareas para las que fue diseñado.

En este caso aplicamos todo lo aprendido y visto en el curso a lo largo del semestre. Esto se puede observar en las tablas y gráficas, ya que hay una gran diferencia en cuanto a las solicitudes, al tamaño de archivos, su volumen de tráfico y que esto ayuda a disminuir los tiempos de carga, haciendo más atractivo y eficiente la página web. Nuestra mejor optimización fue la reestructuración de código de Javascript, ya que en este mejoramos las peticiones de Ajax y la carga de contenido al ingresar la página, que en conjunto con las demás optimizaciones por parte del servidor y del cliente, logramos unos resultados significativamente buenos.

Finalmente, nos quedamos muy satisfechos con lo aprendido y aplicado en este proyecto, ya que nos ayudó a comprender las distintas maneras de optimizar una aplicación web y cuál es la mejor forma de utilizar las distintas técnicas de optimización de aplicaciones web en conjunto, ya que en el mundo laboral es esencial el conocer y aplicar estar todo esto para entregar un producto final de calidad que cuente con un balance correcto de optimización en su cliente, su servidor y su base de datos.

# **Referencias**

1. Smith, P. G. (2012). Professional Website Performance: Optimizing the Front-End and Back-End. John Wiley & Sons
2. Hogan, L.C. (2015). Designing for Performance Weighing aestetics and speed. Canada: OReilly Media.
3. Peter G. Smith. (2013). Professional Website Performance: Optimizing the Front End and the Back End. Indianapolis: John Wiley & Sons.
4. Smith, P.G. (2013). Professional website performance: Optimizing the front-end and back-end. Hoboken, NJ: Wiley
5. Padilla, R. M., & Padilla, R. M. (2008, 21 enero). Optimizando tu HTML. Maestros del Web. <http://www.maestrosdelweb.com/optimizando-tu-html/?fbclid=IwAR0R3PlafpP9btqYzQwbDlmVK_xPiN98Jnct7B2FeTz6gurfz1dLEVe_514>
6. Diapositivas y aplicación del curso de “Optimización de aplicaciones web” FMAT UADY 2021.