

# **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

a18016318@alumnos.uady.mx

a18016312@alumnos.uady.mx

Docente a cargo:

**Dr. Víctor Hugo Menéndez Domínguez**

## 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 Although for many people it is an outdated method, it is still useful to be aware of all the new content that appears every day on the web.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

<b>Resumen.....</b>	<b>2</b>
<b>Abstract.....</b>	<b>2</b>
<b>1. Introducción.....</b>	<b>4</b>
<b>1.1 Objetivo General .....</b>	<b>4</b>
<b>1.2 Objetivos específicos .....</b>	<b>4</b>
<b>1.3 Optimización de aplicaciones web .....</b>	<b>4</b>
<b>1.4 Condiciones de pruebas .....</b>	<b>5</b>
<b>1.5 Funcionalidades del sistema .....</b>	<b>5</b>
<b>1.5 Interfaces.....</b>	<b>6</b>
1.5.1 Landing Page.....	6
1.5.2 Añadir link de canal RSS .....	6
1.5.3 Vista de noticias de manera general .....	7
1.5.4 Vista de noticias por fecha.....	7
1.5.5 Búsqueda de noticias (satisfactoria) .....	8
1.5.6 Búsqueda de noticias (sin coincidencias) .....	8
1.5.7 Ordenar noticias .....	9
<b>2. Arquitectura y tecnologías de la aplicación .....</b>	<b>9</b>
<b>2.1 Lenguajes de programación y de marcado .....</b>	<b>9</b>
<b>2.2 Frameworks, bibliotecas y herramientas .....</b>	<b>9</b>
<b>2.3 Diagrama de componentes.....</b>	<b>10</b>
<b>2.4 Diagrama de base de datos .....</b>	<b>10</b>
<b>3. Evaluación diagnóstica del desempeño inicial de la aplicación.....</b>	<b>11</b>
<b>3.1 Página principal .....</b>	<b>11</b>
<b>3.2 Consulta .....</b>	<b>11</b>
<b>3.3 Ejecuciones de pruebas .....</b>	<b>11</b>
<b>4. Propuestas de mejora aplicadas.....</b>	<b>12</b>
<b>4.1 Lado del cliente.....</b>	<b>13</b>
<b>4.2 Lado del servidor.....</b>	<b>13</b>
<b>5. Contraste de resultados .....</b>	<b>14</b>
<b>5.1 Resultados Obtenidos de la optimización.....</b>	<b>14</b>
5.1.1 Página principal .....	14
5.1.2 Consulta.....	14
<b>5.2 Comparación de resultados .....</b>	<b>15</b>
5.2.1 Resultados de Lighthouse .....	15
<b>6. Conclusiones. ....</b>	<b>18</b>
<b>Referencias.....</b>	<b>19</b>

## 1. 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 esperaba 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 transferencia 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.

b. **Servidor:** 127.0.0.1 mediante XAMPP v7.3.28 de 64 bits

c. **Base de Datos:** MySQL mediante phpMyAdmin

Dirección de acceso: <http://localhost/phpmyadmin/>

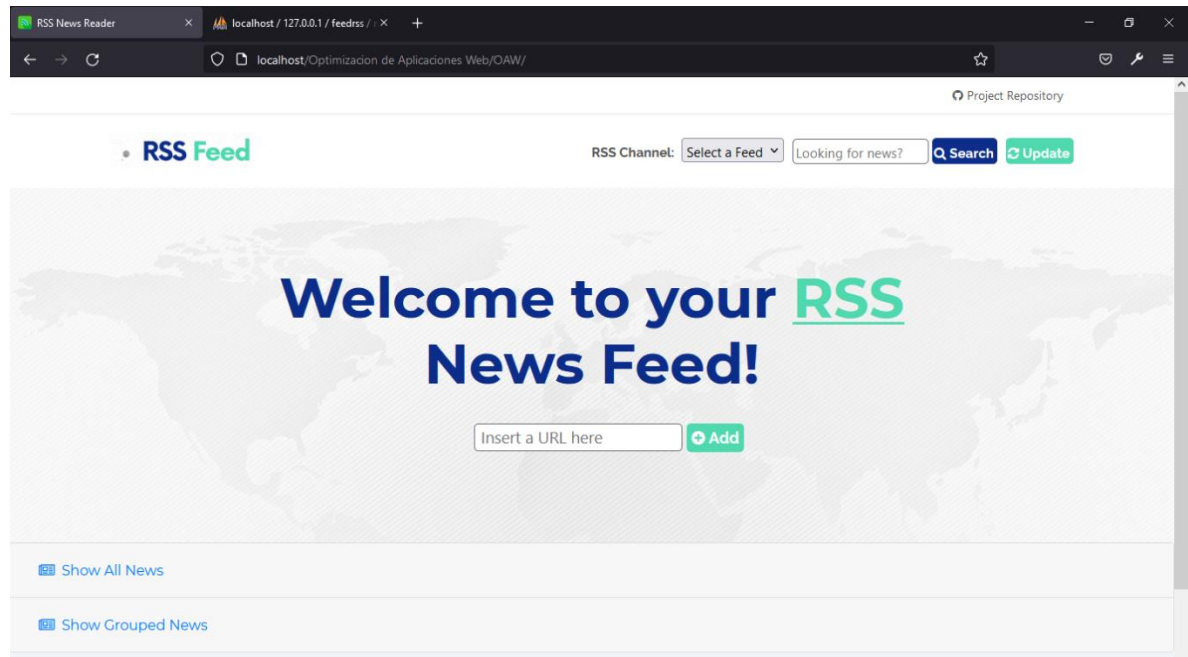
## 1.5 Funcionalidades del sistema

El sistema contará con las siguientes funcionalidades:

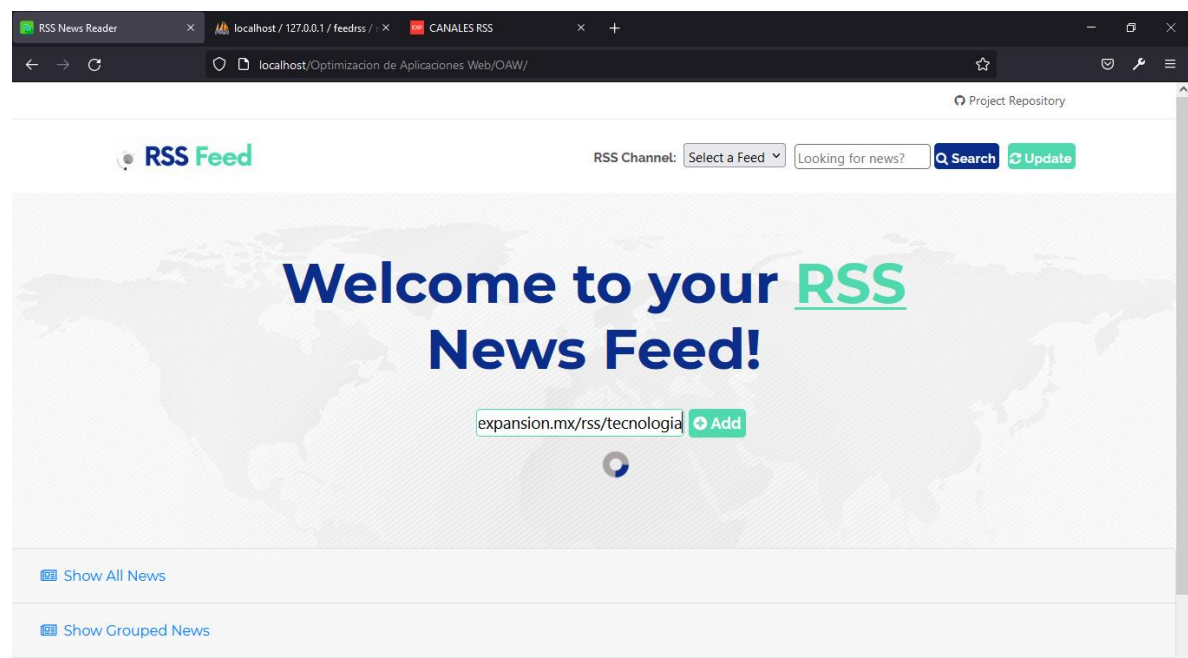
- Añadir a la base de datos canales de noticias RSS
- Actualizar la base de datos con noticias nuevas
- Buscar noticias por palabras claves
- Mostrar las noticias por categoría, descripción, url, título y fecha
- Listar un resumen de las noticias cada una con su miniatura y enlace al artículo original.

## 1.5 Interfaces

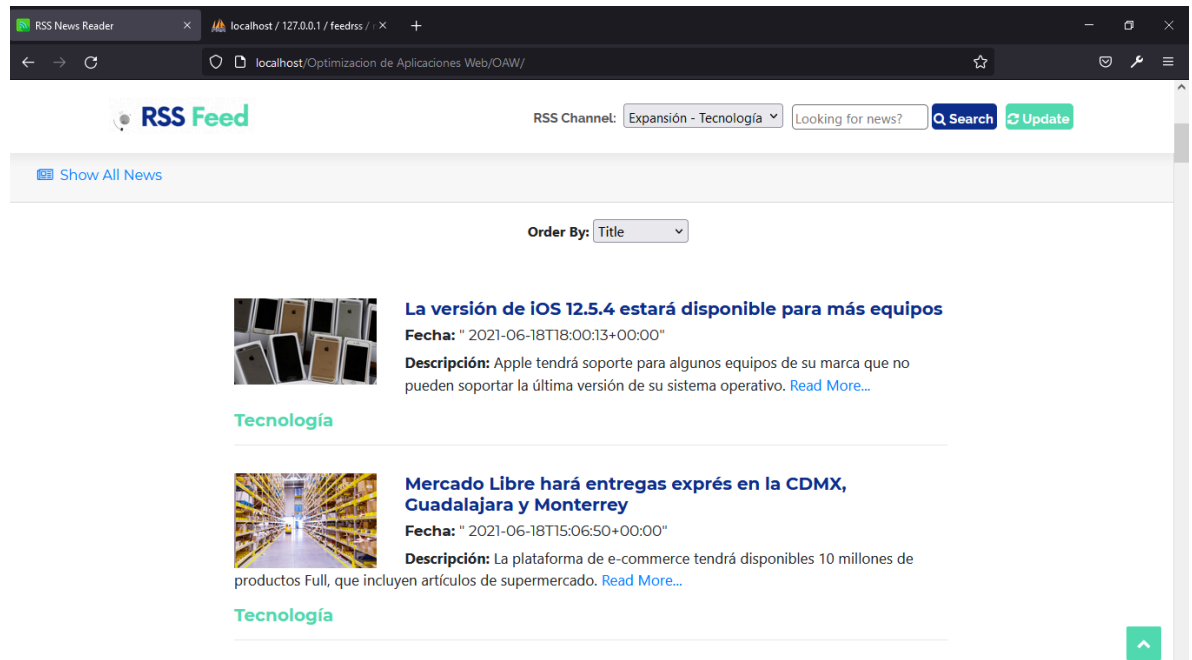
### 1.5.1 Landing Page



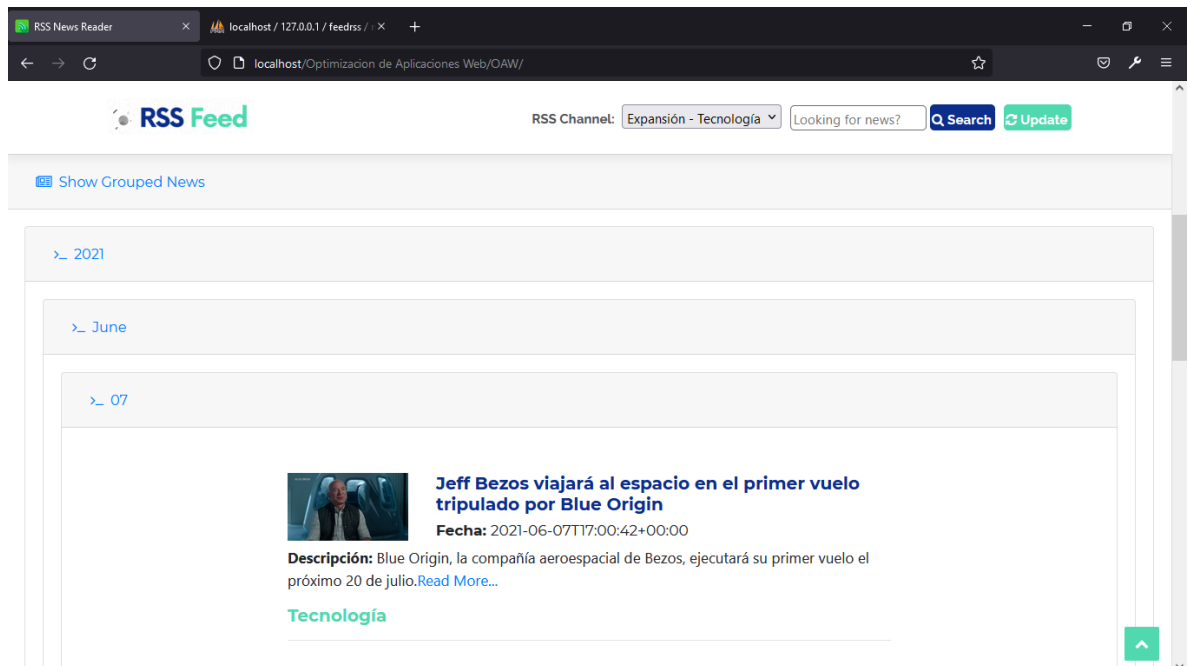
### 1.5.2 Añadir link de canal RSS



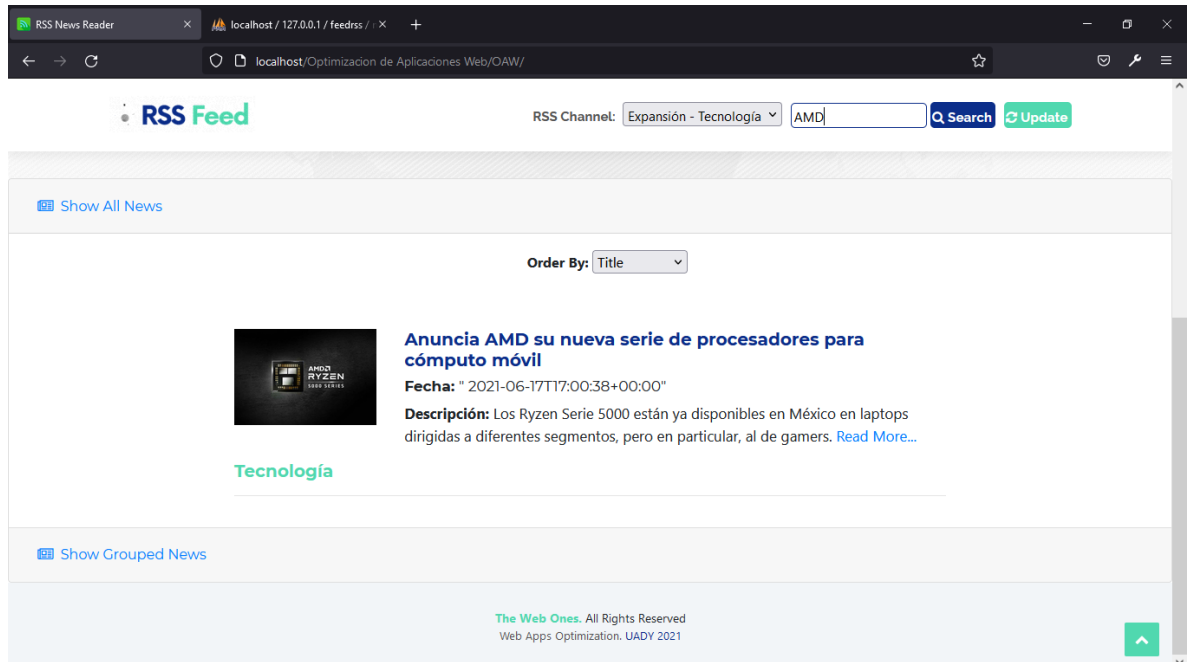
### 1.5.3 Vista de noticias de manera general



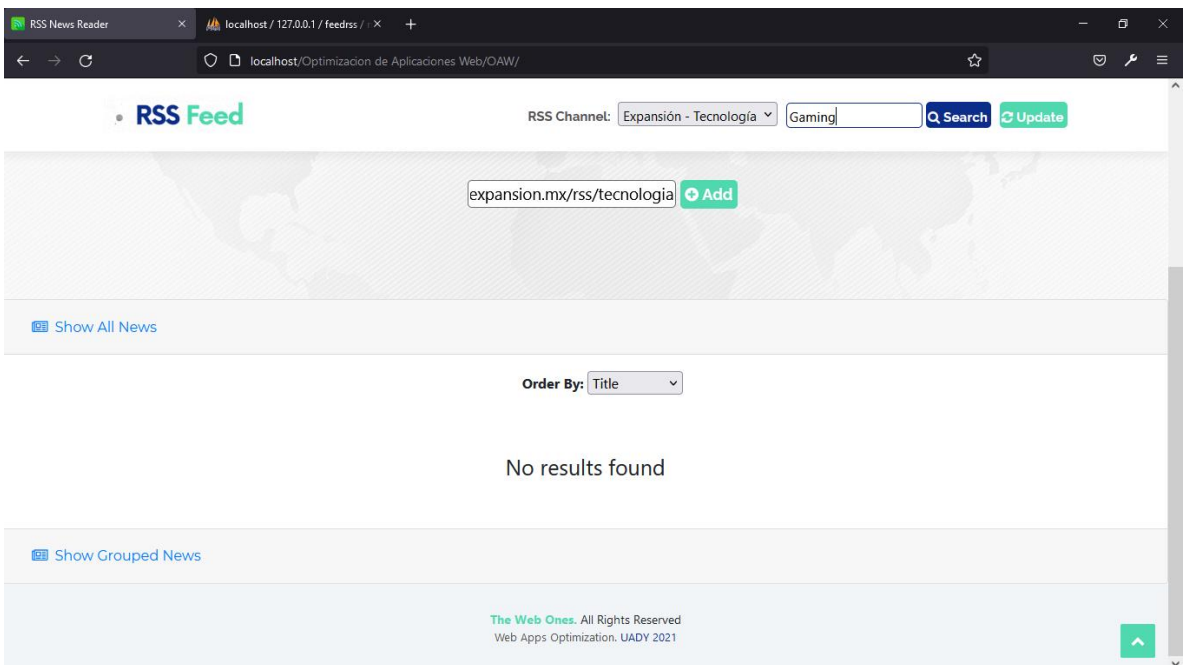
### 1.5.4 Vista de noticias por fecha



### 1.5.5 Búsqueda de noticias (satisfactoria)

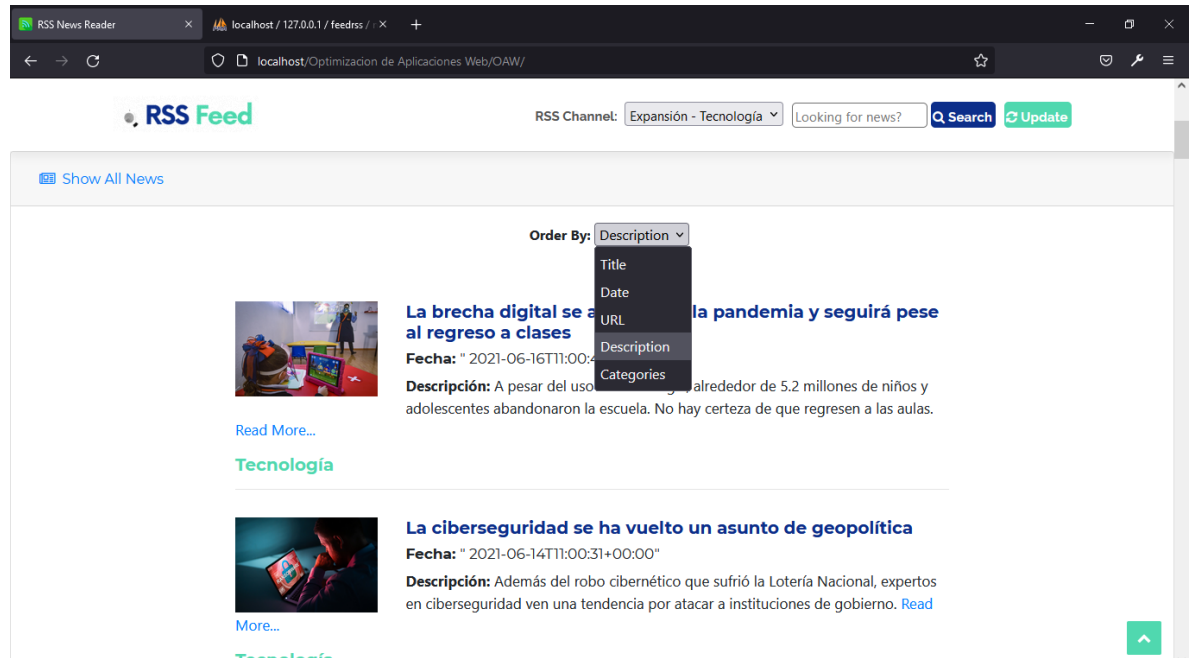


### 1.5.6 Búsqueda de noticias (sin coincidencias)





### 1.5.7 Ordenar noticias



## 2. Arquitectura y tecnologías de la aplicación

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

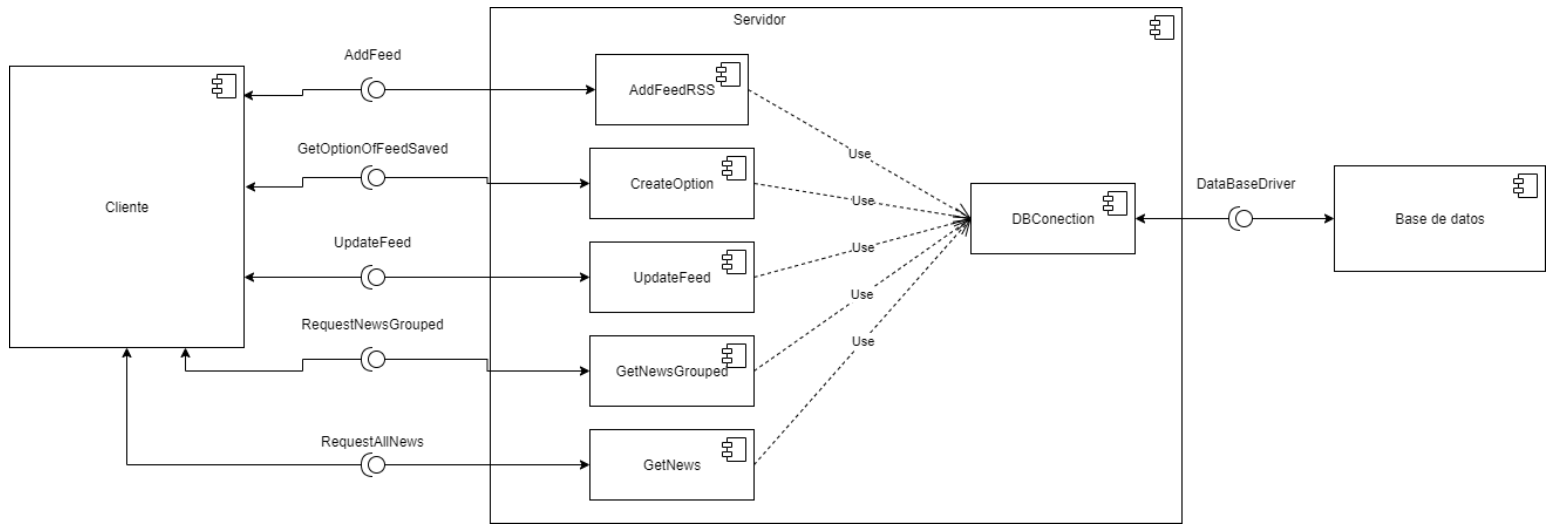
### 2.1 Lenguajes de programación y de marcado

- **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.

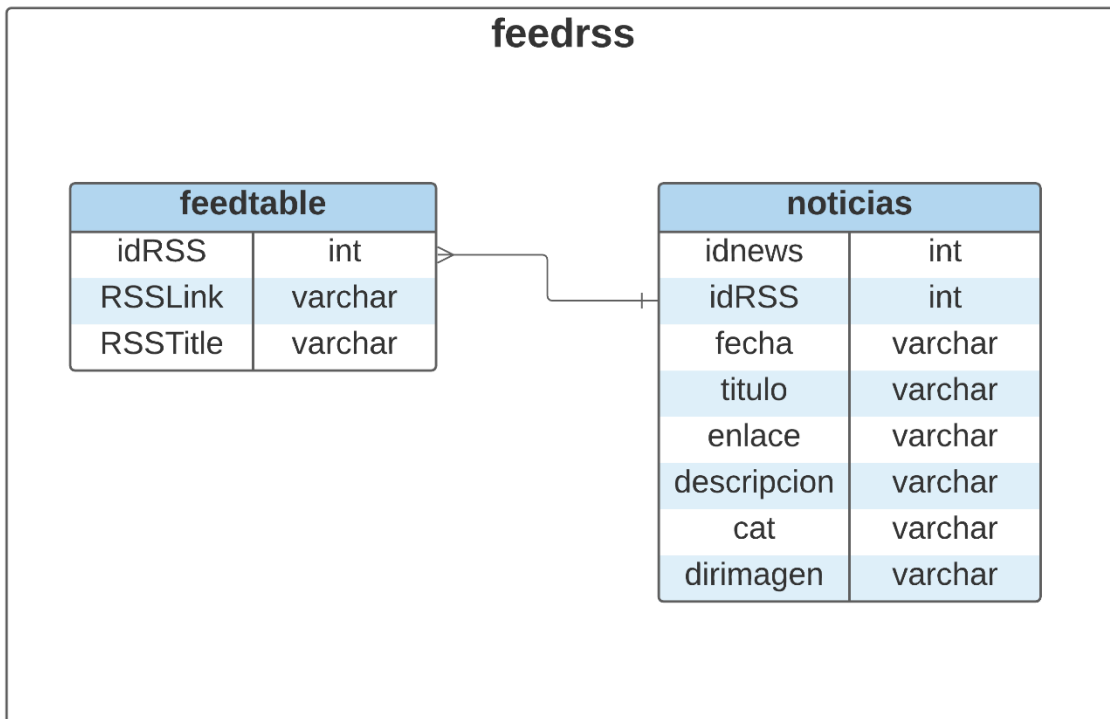
### 2.2 Frameworks, bibliotecas y herramientas

- **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.

## 2.3 Diagrama de componentes



## 2.4 Diagrama de base de datos



### 3. 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.

Aggregate Report

Name:

Aggregate Report

Comments:

Write results to file / Read from file

Filename

Label	# Samples	Average	Median	Received KB/sec	Throughput
HTTP Request	100	128	127	1603.46	1.7/sec
TOTAL	100	128	127	1603.46	1.7/sec

No of Samples

100

Latest Sample

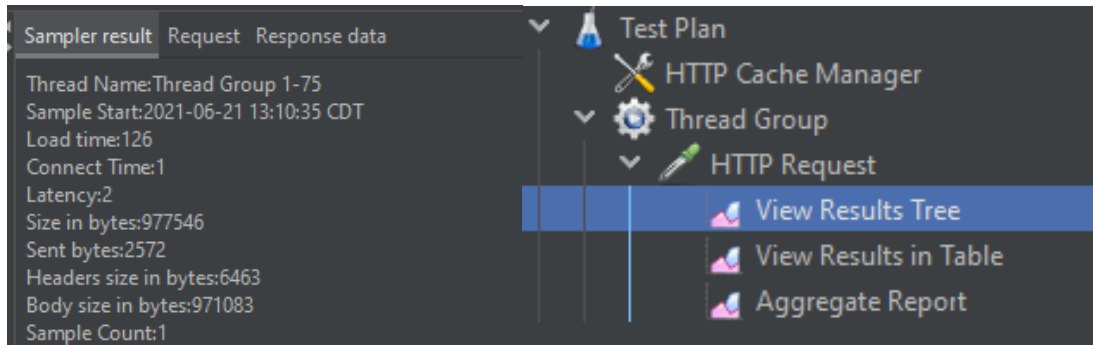
127

Average

128

Deviation

7



Primera página principal

69 requests	13.4 MB transferred	13.4 MB resources	Finish: 322 ms	DOMContentLoaded: 160 ms	Load: 316 ms
-------------	---------------------	-------------------	----------------	--------------------------	--------------

Segunda página principal

69 requests	69.4 kB transferred	13.4 MB resources	Finish: 191 ms	DOMContentLoaded: 140 ms	Load: 185 ms
-------------	---------------------	-------------------	----------------	--------------------------	--------------

Primera consulta

Name	Status	Type	Initiator	Size	Time	Waterfall
GetNews.php?id=64	200	xhr	main.js:262	25.9 kB	17 ms	
GetNewsGrouped.php?id=64	200	xhr	main.js:276	25.9 kB	18 ms	
news/	200	text/html	main.js:356	11.3 kB	8 ms	

Segunda consulta

Name	Status	Type	Initiator	Size	Time	Waterfall
GetNews.php?id=64	200	xhr	main.js:262	(disk cache)	2 ms	
GetNewsGrouped.php?id=64	200	xhr	main.js:276	(disk cache)	2 ms	
news/	200	text/html	main.js:356	11.3 kB	8 ms	

## 4. 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.

## 5. Contraste de resultados

### 5.1 Resultados Obtenidos de la optimización

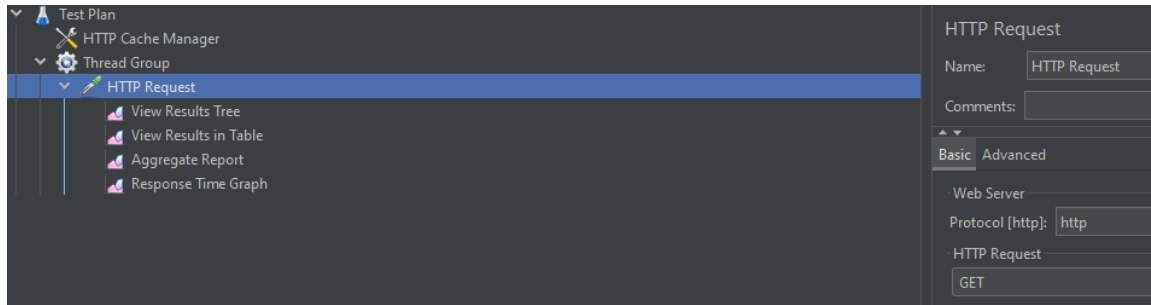
#### 5.1.1 Página principal

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

Sampler result Request Response data			
Thread Name:Thread Group 1-10			
Sample Start:2021-06-21 16:22:51 CDT			
Load time:132			
Connect Time:1			
Latency:2			
Size in bytes:560539			
Sent bytes:2516			
Headers size in bytes:6805			
Body size in bytes:553734			
Sample Count:1			
Error Count:0			
Data type ("text" "bin" ""):text			
Response code:200			
Response message:OK			
HTTPSampleResult fields:			
ContentType: text/html			
DataEncoding: null			
Label	# Samples	Average	Median
HTTP Request	200	133	133
TOTAL	200	133	133



Primera visita a página principal:

24 requests | 568 kB transferred | 931 kB resources | Finish: 282 ms | DOMContentLoaded: 171 ms | Load: 214 ms

Segunda visita a página principal:

24 requests | 19.8 kB transferred | 931 kB resources | Finish: 178 ms | DOMContentLoaded: 150 ms | Load: 172 ms

Primera consulta:

Name	Status	Type	Initiator	Size	Time
GetNews.php?id=82	200	xhr	main.js:1	8.5 kB	7 ms
GetNewsGrouped.php?id=82	200	xhr	main.js:1	8.6 kB	8 ms
news/	200	text/html	main.js:1	11.3 kB	14 ms
default.png	200	png	(index):1	14.6 kB	3 ms

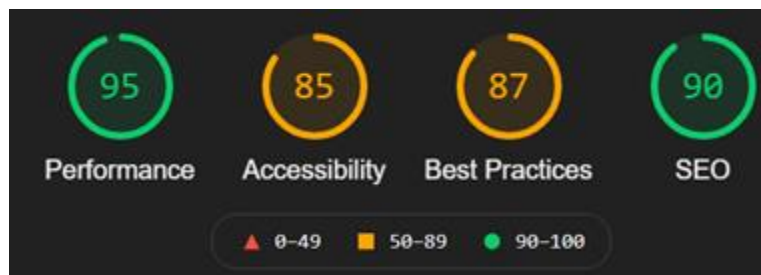
Segunda consulta:

Name	Status	Type	Initiator	Size	Time
GetNews.php?id=82	200	xhr	main.js:1	(disk cache)	2 ms
GetNewsGrouped.php?id=82	200	xhr	main.js:1	(disk cache)	2 ms
news/	200	text/html	main.js:1	11.3 kB	8 ms
default.png	200	png	(index):1	(disk cache)	1 ms

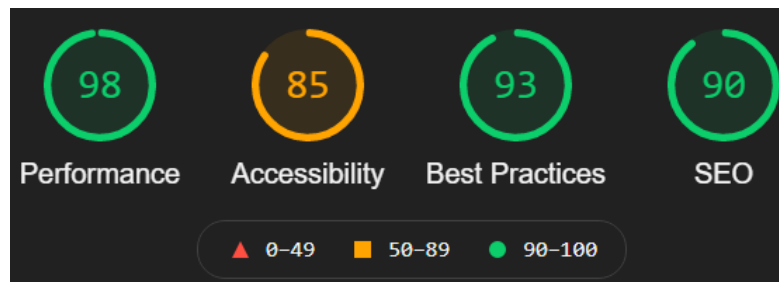
## 5.2 Comparación de resultados

### 5.2.1 Resultados de Lighthouse

Antes de optimizar:



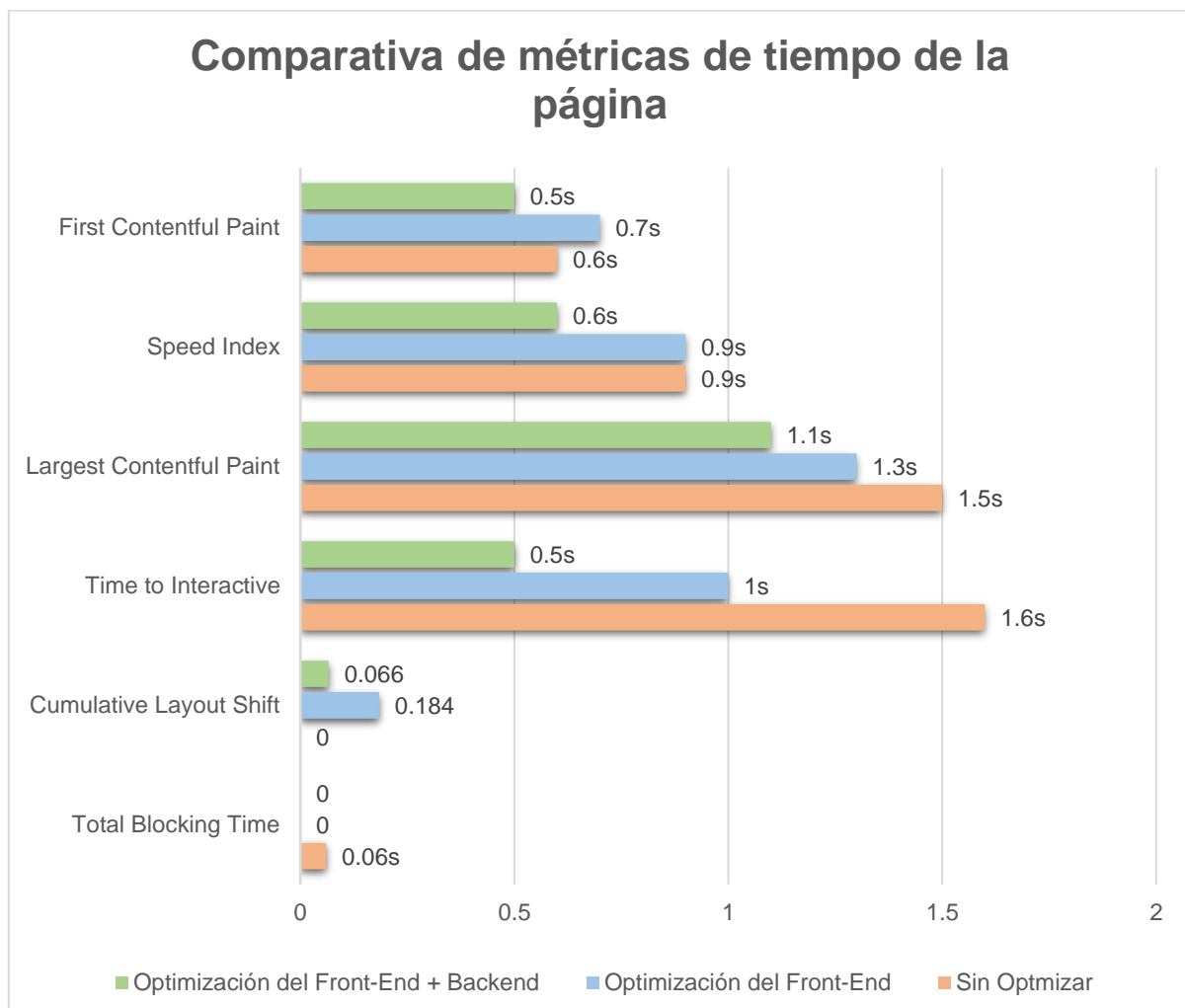
Después de optimizar:



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>

### 5.2.2 Comparación de métricas





### 5.2.3 Comparación de tamaño de archivos

Nombre de Archivo	Tamaño antes de optimizar	Tamaño después de optimizar	% de reducción
 index.html	5.39 KB (5,527 bytes)	4.11 KB (4,299 bytes)	23.72%
 style.css	11.8 KB (12,140 bytes)	7 KB (7172 bytes)	26.69%
 main.js	15.7 KB (16,084 bytes)	9.91 KB (10,155 bytes)	36.86%
 bootstrap.min.css	157 KB (161,205 bytes)	157 KB (161,160 bytes)	0.03%
 bootstrap.bundle.min.js	82.4 KB (84,384 bytes)	82.1 KB (84,092 bytes)	3.35%
 jquery.min.js	87.3 KB (89,478 bytes)	87.2 KB (89,390 bytes)	0.1%
 jquery.easing.min.js	2.47 KB (2,532 bytes)	Ya esta minimizado.	0%
 jquery.sticky.js	9.51 KB (9,748 bytes)	3.91 KB (4,012 bytes)	58.85%
 owl.carousel.min.js	43.3 KB (44,348 bytes)	43.1 KB (44,176 bytes)	0.39%
 superfish.min.js	4.40 KB (4,508 bytes)	4.16 KB (4,270 bytes)	5.28%
 wow.min.js	7.96 KB (8,159 bytes)	7.85 KB (8,048 bytes)	1.36%
 background2.png	198 KB (203,579 bytes)	140 KB (143,426 bytes)	29.54%
 default.png	53.9 KB (55,244 bytes)	7 KB (7172 bytes)	74.24%
 favicon.png	54.3 KB (55,671 bytes)	9.91 KB (10,155 bytes)	69.47%
 logo.gif	434 KB (445,101 bytes)	157 KB (161,160 bytes)	76.43%

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 CACHE	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 CACHE	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

## 6. 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 aesthetics 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](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.