



UADY
UNIVERSIDAD
AUTÓNOMA
DE YUCATÁN
"Luz, Ciencia y Verdad"

Reporte de actividades

Fecha de entrega:

25 de mayo del 2022

Maestro: Víctor Hugo Menéndez Domínguez

Optimización de aplicaciones web

Miembros del equipo:

Ana Pérez

Carlos Morales

Lenin Rosas

Pablo Rosas



- **Introducción**

- **Objetivo**

Describir la aplicación Lector web de noticias RSS, detallado sus principales funcionalidades y componentes del sistema. Se hace un énfasis en la optimización de la aplicación, para lo cual, este documento incluye información de su desempeño, así como las propuestas aplicadas para mejorar su rendimiento, tanto del lado del cliente como del servidor, posteriormente se realizarán pruebas en donde se comparen los resultados de las acciones de mejora.

- **Descripción de la aplicación**

El lector RSS es una página web, creada con HTML, CSS, JavaScript y PHP principalmente. La finalidad de esta aplicación es poder agregar los enlaces de los feeds de sitios de noticias que el usuario desee ver; estas noticias se muestran en miniatura en la parte de inicio de la página, pero es posible verlas completas al seleccionar alguna de las miniaturas.

En la sección 1.3 se incluyen descripciones más detalladas de cada una de las funcionalidades de la aplicación, así como sus respectivas capturas de cada una de ellas.

- **Importancia de la optimización**

En general, es recomendable que cualquier sistema pase por un proceso de optimización. Para el lector de feeds la importancia de este proceso es debido a que se desea que el usuario tenga la mejor experiencia al utilizar la aplicación y la prefiera por sobre otra similares; al recibir demasiada información de todas las noticias (y conforme se vayan agregando más fuentes de información no hará más que incrementar el tamaño de datos recibidos), el sitio debe ser capaz de poder aceptarla toda y seguir teniendo un buen rendimiento, sin que se hagan muy largos los tiempos de espera para cargar las noticias en el inicio o al actualizarlas.



De igual manera, para la parte de los desarrolladores, con un código bien organizado y optimizado se facilita la tarea del mantenimiento y corrección de errores.

- **Funcionalidades**

Página de Recientes



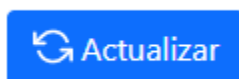
En esta sección se listan las ocho noticias más recientes, además, conforme se va deslizando hacia abajo en la página, las noticias también se muestran separadas por categoría. En la parte final, está la opción para cambiar de página y continuar con la navegación de las noticias:

Página 1 de 8

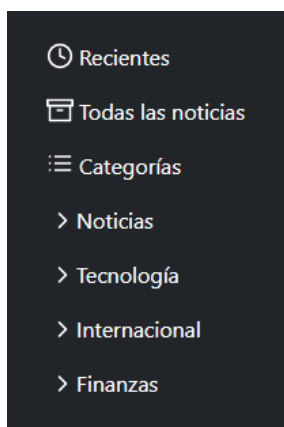
Anterior

Siguiente

Esta pantalla también cuenta con otras funcionalidades como *Actualizar*, ubicada en la parte superior derecha:



Esta opción, como el botón lo indica, refresca la página para que aparezcan noticias nuevas, es decir, más recientes que las que estaban listadas con anterioridad.



En la sección de la izquierda se encuentra un menú, con varias funcionalidades de esta aplicación web. La primera es *Recientes*, que te dirige a la pantalla mostrada anteriormente con las noticias del día en el que se realiza la consulta. Después se tiene *Todas las noticias*:

Todas las noticias

Todas las noticias existentes de tus feeds

Actualizar

Ordenar por ▼



Clima en Jalisco, hoy 18 de mayo del 2022: cielo nublado y presencia de lluvias habrá este día

Debate hace 15 minutos



Fiscal Anticorrupción ratifica el desafuero de sobrina de AMLO

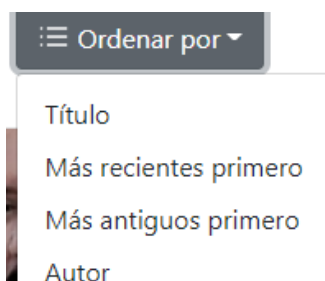
Debate hace 18 minutos



Caso Estrada Ferreiro: ¿AMLO tiene la última palabra?

Debate hace 19 minutos

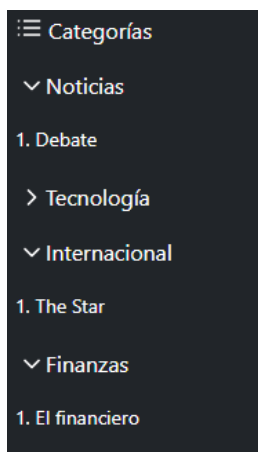
En esta parte se cargan todas las noticias, sin importar la categoría a la que pertenezcan. Como con la de Recientes, también se tiene disponible el botón de *Actualizar*, pero adicional a este, se cuenta con el *Ordenar por*:



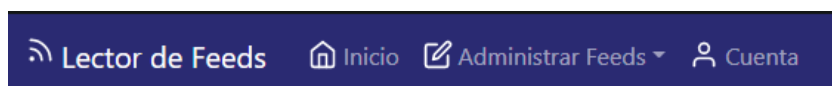


Esta opción hace posible que las noticias cambien de posición dependiendo de la elección del usuario de entre las 4 opciones que se pueden observar en la imagen anterior.

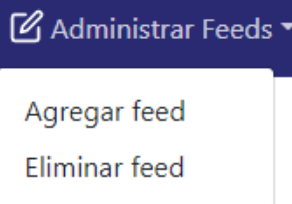
Regresando al menú lateral, también incluye el apartado de *Categorías*, que lista todos los tipos existentes hasta el momento. Al seleccionar alguna, se abre una subsección en donde lista el nombre de los lugares donde se extraen las noticias, como se ve a continuación:



Para esta demostración, se seleccionaron las categorías *Noticias*, *Internacional* y *Finanzas*. Para poder agregar más opciones o nuevas páginas de donde sacar noticias, es posible realizarlo con una de las funcionalidades que se encuentran en la barra de navegación, que está en la parte superior de la página:



En este menú, tanto el botón de *Inicio* y el logo de *Lector de Feeds*, pueden ser seleccionados y ambos redireccionarán a la página principal de la página. Después está la opción de *Administrar Feeds*, que, al darle clic, lista dos funcionalidades más:



Agregar feed es en donde se añaden nuevas fuentes de información para extraer las noticias y *Eliminar feed* es para borrar cualquiera de las que se han agregado, al elegir alguna de estas dos opciones, el sistema carga la pantalla con dos secciones para hacer lo anterior:

Añadir feed(s) ^

Agrega tus feeds

y no te pierdas ninguna noticia de tus fuentes favoritas.

Nombre

Categoría

Link

Agregar feed

Modificar feed(s) v

ID	Fuente	Opciones
82	Debate	<div>Eliminar</div> <div>Editar</div>
84	SDP Noticias	<div>Eliminar</div> <div>Editar</div>
86	The Star	<div>Eliminar</div> <div>Editar</div>
89	El financiero	<div>Eliminar</div> <div>Editar</div>

La primera es para agregar, la cual contiene campos necesarios para tener un mejor orden de las noticias. La parte de la derecha es donde se eliminan los feeds que se tienen hasta ahora; se listan todas las fuentes y además se cuenta con dos botones, uno para borrar definitivamente y el otro que es simplemente para editar alguno de los campos que se llenaron cuando se agregó.

- **Arquitectura y tecnologías de la aplicación**

Arquitectura.

Three-Layer.

La arquitectura de tres capas presenta una capa intermedia en donde se realiza toda la lógica a partir de las entradas recibidas en la capa de presentación, estos datos permiten a la capa lógica la interacción con la capa de datos. Esto nos sirvió



para tener mejor control sobre cada capa, aplicando el principio de la Responsabilidad Única.

Así mismo, presenta varias ventajas como:

- Mayor escalabilidad de la aplicación.
- Mejoría en el rendimiento.
- Sencilla de implementar.
- Reducción de la latencia y tiempos de comunicación.

A continuación, se describe cada capa y la funcionalidad que tiene en la página web.

Capa de presentación.

Se encarga de generar y presentar la interfaz al usuario. Los datos ingresados por usuario en esta capa son procesados por la capa intermedia (capa lógica), cuando están listos los resultados son procesados para generar la presentación con dichos datos, finalmente son mostrados al usuario.

Los navegadores web son los encargados de hacerle llegar la presentación al usuario para que se pueda llevar a cabo la interacción de la presentación con el usuario.

Capa lógica.

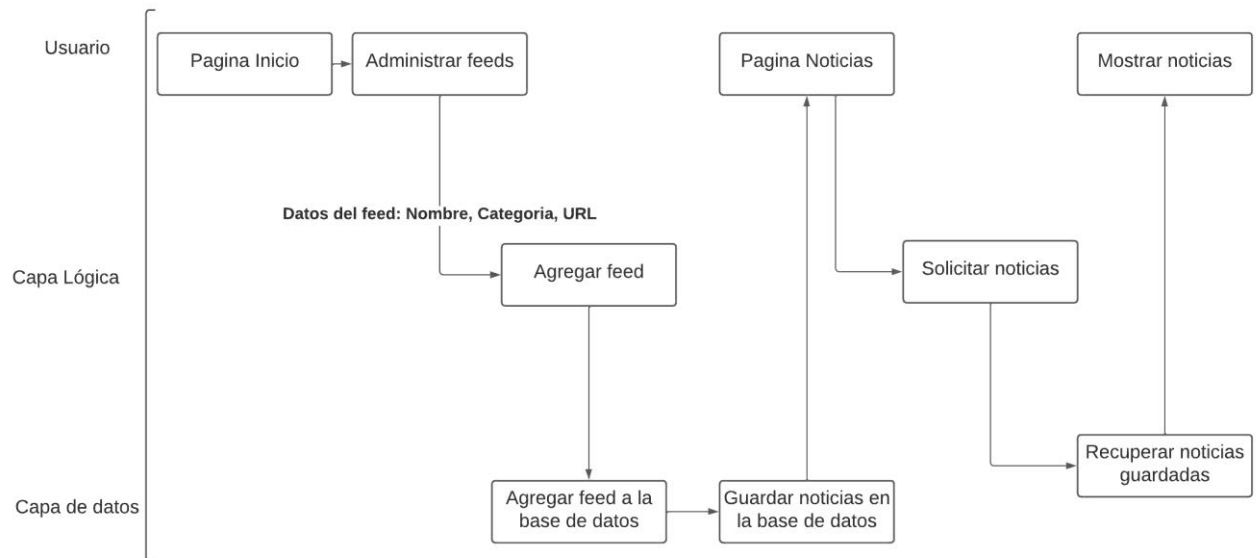
La capa lógica o capa intermedia se hace cargo de conectar la capa de presentación con la capa de datos mediante las entradas y salidas de estas capas. Esta capa toma la información que el usuario introduce en la capa de presentación para realizar las operaciones necesarias.

Cuando se han hecho las operaciones necesarias entre la capa de presentación y la de datos, la capa intermedia mediante su lógica programada compone el HTML necesario para la capa de presentación.

Esta capa es realizada por los archivos PHP y Javascript de la página web, los cuales realizan las operaciones requeridas.

Capa de datos.

Consiste en un software de gestión de base de datos, en esta capa se gestionan las peticiones de información por parte de la capa intermedia. En esta capa se guarda la información sobre los feeds registrados. El software utilizado para esta capa es MariaDB, el cual es integrado por Lampp.



Tecnologías.

Bootstrap. Es un framework creado para el desarrollo web. Consiste en una serie de códigos que se relacionan entre sí para brindar una funcionalidad específica. Es utilizado principalmente por programadores front-end, pues ofrece recursos que simplifican y aceleran la creación de sitios web.

JavaScript. Es un lenguaje de programación o de secuencias de comandos que permite implementar funciones complejas en páginas web. Permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes, entre otras cosas.

PHP: Es un lenguaje de programación para desarrollar aplicaciones y crear sitios web, favoreciendo la conexión entre los servidores y la interfaz de usuario. Es de código abierto, lo que significa que cualquier puede hacer cambios en su estructura.



De igual forma, está en constante perfeccionamiento. Generalmente es definido como un lenguaje del lado del servidor.

XAMPP/LAMPP: Es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl.

SimplePie: Es un script cuyo funcionamiento consiste en interpretar los contenidos de cualquier canal RSS, convirtiéndolo en una página web.

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

Página principal (Todas las noticias)

	Volumen de transferencia	Tiempo de transferencia.
Primera visita (Sin caché).	22.7 MB	4.61 s
HTML / PHP	341 kB transferidos	
JS	81.4 kB transferidos	
IMG	22.2 MB transferidos	
CSS	245 kB transferidos	
Siguiente visita (Con caché).	1.5 MB	938 ms
HTML / PHP	341 kB transferidos	
JS	0 b transferidos	
IMG	1.4 MB transferidos	



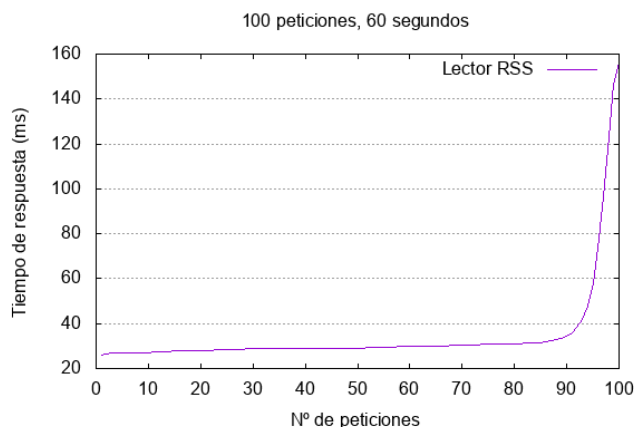
CSS	0 b transferidos
-----	------------------

Página resultados de búsqueda (Noticias de un Feed en específico: *El debate*)

	Volumen de transferencia	Tiempo de transferencia.
Primera visita (Sin caché).	8.2 MB	1.98 s
HTML / PHP	146 kB transferidos	
JS	81.4 kB transferidos	
IMG	7.9 MB transferidos	
CSS	245 kB transferidos	
Siguiente visita (Con caché).	146 kB	190 ms
HTML / PHP	146 kB transferidos	
JS	0 b transferidos	
IMG	0 b transferidos	
CSS	0 b transferidos	

Prueba de carga utilizando Apache Benchmark previo a la optimización del sitio.

Prueba 1. Página principal



Resultados Prueba 1. Página principal:

- Tardó 3.397 segundos en completar el envío de 100 solicitudes.
- El servidor pudo responder a todas las solicitudes sin tener alguna falla.
- El servidor pudo manejar en promedio 29.44 solicitudes por segundo.
- La velocidad de transferencia de datos fue de 9778.79 Kbytes/segundo

```
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done

Server Software:      Apache/2.4.52
Server Hostname:      localhost
Server Port:          80

Document Path:        /sitio-no-optimizado/todasNoticias.php
Document Length:      339934 bytes

Concurrency Level:    1
Time taken for tests:  3.397 seconds
Complete requests:    100
Failed requests:       0
Total transferred:    34015800 bytes
HTML transferred:     33993400 bytes
Requests per second:  29.44 [#/sec] (mean)
Time per request:     33.970 [ms] (mean)
Time per request:     33.970 [ms] (mean, across all concurrent requests)
Transfer rate:        9778.79 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0    0   0.0      0    0
Processing: 26   34  21.7     29   156
Waiting:    2    2   1.0      2    8
Total:      26   34  21.7     29   156

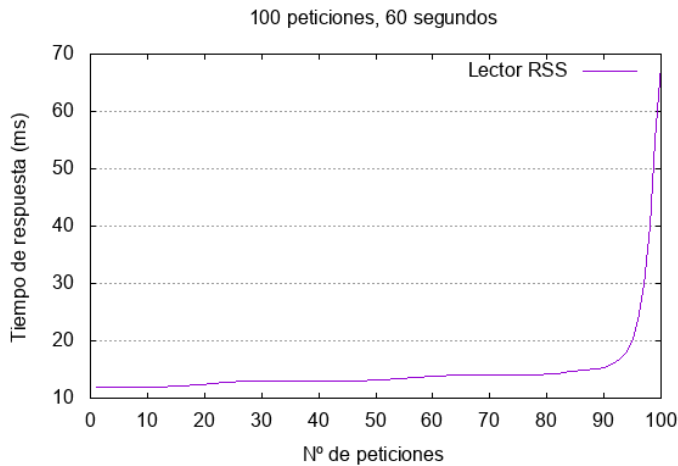
Percentage of the requests served within a certain time (ms)
 50%    29
 66%    30
 75%    31
 80%    31
 90%    33
 95%    43
 98%   152
 99%   156
100%   156 (longest request)
```

Tiempos de respuesta para las diferentes etapas de las solicitudes HTTP:

- Dado que es el proyecto se encuentra alojado localmente, no tomó tiempo en establecerse la conexión con el servidor.
- El tiempo de procesamiento de las solicitudes fue en promedio de 34ms.
- El tiempo promedio que se tuvo que esperar entre el envío de la solicitud y la respuesta fue de 2 ms.

- Finalmente, tenemos que el tiempo promedio total transcurrido desde que se inició la conexión, se recibió la respuesta y su cierre fue de 34 ms.

Prueba 2. Resultados de búsqueda (feed El debate)



Resultados Prueba 2. Resultados de búsqueda (feed El debate)

- Tardó 1.473 segundos en completar el envío de 100 solicitudes.
- El servidor pudo responder a todas las solicitudes sin tener alguna falla.
- El servidor pudo manejar en promedio 67.88 solicitudes por segundo.
- La velocidad de transferencia de datos fue de 9664.72 Kbytes/segundo

```
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done

Server Software:      Apache/2.4.52
Server Hostname:      localhost
Server Port:          80

Document Path:        /sitio-no-optimizado/noticiasFeed.php?feed=El%20debate
Document Length:      145567 bytes

Concurrency Level:    1
Time taken for tests:  1.473 seconds
Complete requests:    100
Failed requests:       0
Total transferred:    14579100 bytes
HTML transferred:     14556700 bytes
Requests per second:  67.88 [#/sec] (mean)
Time per request:     14.731 [ms] (mean)
Time per request:     14.731 [ms] (mean, across all concurrent requests)
Transfer rate:        9664.72 [Kbytes/sec] received

Connection Times (ms)
      min      mean[+/-sd] median    max
Connect:    0       0   0.0      0
Processing: 12      15   7.5     13
Waiting:    2       2   0.7      2
Total:      12      15   7.5     13

Percentage of the requests served within a certain time (ms)
 50%    13
 66%    14
 75%    14
 80%    14
 90%    15
 95%    18
 98%    63
 99%    68
100%    68 (longest request)
```

Tiempos de respuesta para las diferentes etapas de las solicitudes HTTP:

- Dado que es el proyecto se encuentra alojado localmente, no tomó tiempo en establecerse la conexión con el servidor.
- El tiempo de procesamiento de las solicitudes fue en promedio de 15ms.
- El tiempo promedio que se tuvo que esperar entre el envío de la solicitud y la respuesta fue de 2 ms.



- Finalmente, tenemos que el tiempo promedio total transcurrido desde que se inició la conexión, se recibió la respuesta y su cierre fue de 15 ms.

- **Propuestas de mejora aplicadas**

Minimización / Ofuscamiento

Este proceso ayuda mucho para poder hacer el código lo más compacto posible y que este pese muy poco. Este proceso le quita todos los espacios innecesarios, cambios de línea, tabuladores, comentarios, etcétera. Con lo anterior, queda un código muy complicado para la lectura, pero que ocupa mucho menos espacio y se puede descargar más rápido desde el servidor.

Las herramientas utilizadas son las siguientes:

1. PHP Minify
2. CSSMin
3. YUI Compressor

Templates con PHP

Las plantillas realizadas en PHP ayudan a ahorrarse muchas líneas de código que se repetirían si no se utilizaran estas plantillas, además, se ahorra mucho tiempo y esfuerzo a la hora del mantenimiento pues no se tiene que realizar la modificación en diferentes archivos.

Para este proyecto se identificaron 3 secciones en común de las páginas:

- Header
- Modal
- Sidebar

Se crearon plantillas para evitar su repetición, reduciendo así también el tamaño total de los archivos que la incluían.

Sprites CSS



Se utilizaron ya que con esto tenemos una única imagen, por lo que el servidor web solo deberá realizar una petición en vez de la cantidad de peticiones que se realizarían por cada uno de los elementos del sprite. De igual forma, seleccionar las imágenes que se quieren utilizar en cada parte de la web mediante CSS es más rápido que realizar una petición al servidor para que este nos sirva esa información.

Se agruparon 10 iconos correspondientes a la barra de navegación, lateral y los incluidos en algunos botones de la página en una sola imagen.

Optimización de las imágenes

La gran parte del tamaño de las páginas web corresponden a imágenes, siendo algunas de ellas muy grandes y ocasionando la ralentización de la página web y, por lo tanto, una mala experiencia por parte del usuario.

Como solución podemos reducir el tamaño y calidad de las imágenes y aún mantenerlas lo suficientemente atractivas para mostrarse en la página.

Se utilizó la herramienta Squoosh para reducir el tamaño de las imágenes estáticas de la página hasta un 75% de su tamaño original y con una disminución de la calidad tanto como se pudo sin llegar a una diferencia notable.

Implementación de carga diferida (Lazy Loading)

Se hizo uso de esta técnica ya que, al momento de cargar tantas noticias el sitio tomaba un tiempo mayor de descarga, maquetado y carga inicial, por lo que, optamos por el Lazy Loading para que el usuario pueda ver el contenido que está en la ubicación que se encuentra y le permita al sitio cargar los demás recursos sin tanta sobrecarga.

Esta técnica fue aplicada para las páginas "todas las noticias" y las correspondientes a los feeds en específico, cargando únicamente las noticias que son visibles y retrasando la carga del resto hasta que se necesiten, es decir, que el usuario deslice hacia abajo y descubra más noticias.

Paginación de las noticias



Otra técnica aplicada para solucionar el problema de tener que descargar una cantidad muy elevada de noticias fue la paginación. Esta consiste en dividir los contenidos de una sección en grupos de resultados y desplegar las opciones para avanzar y retroceder al final de la página.

Aplicamos esta técnica para la página de noticias recientes, limitando la cantidad de noticias con un máximo de ocho por cada categoría.

Para descubrir el resto de las noticias de hoy, se debe ir pasando de página con los controles situados al final, reduciendo así el número de solicitudes que se realizan al servidor.

Filtrado de noticias

Para que el usuario tenga una experiencia con mayor personalización se creó un apartado de Filtrado de noticias, estas funcionalidades también permiten controlar la mayor parte de la carga al reducir el número de recursos generados y peticiones hechas a la base de datos.

Compresión gzip.

La compresión gzip reduce los tiempos de respuesta reduciendo el tamaño de la respuesta HTTP.

Para ello, se comenzó por habilitar la extensión zlib de PHP, la cual nos ofrece la opción de comprimir de forma transparente las páginas sobre la marcha, si el navegador solicitante lo admite.

De igual manera se habilitaron los módulos mod_deflate y mod_filter para apache y se continuó con la especificación del directorio que debe servirse usando la compresión gzip, así como también los tipos de archivos a ser comprimidos.

Uso de la caché

Con el almacenamiento en caché, una copia de los datos se almacena en el lado del cliente o servidor, de modo que el contenido no tiene que ser recuperado desde



el servidor cada vez que se realiza una petición. Esto ahorra ancho de banda, disminuye la carga en el servidor y reduce el tiempo de respuesta.

Para este proyecto se realizaron configuraciones para el control de la caché a través de las cabeceras HTTP. También se habilitó el op caché y la caché en el lado del servidor, con el que el contenido al que más se accede es almacenado en la memoria del servidor de manera que pueda ser más rápido.

Caché del navegador

Para habilitar la caché en el navegador trabajamos con las cabeceras Expires y Caché-Control con el objetivo de configurar los archivos como cacheables y así evitar peticiones HTTP innecesarias en numerosas visitas de la página.

Apache habilita estas cabeceras gracias a los módulos `mod_expires` y `mod_headers`.

Por lo tanto, tras activar los módulos, se añadieron las expiraciones de los diferentes archivos especificando el tiempo en segundos en el que será válido después del acceso. En el caso de `mod_headers` no fue necesario asignar la directiva `max-age` debido a que ya es configurado por el módulo `mod_expires`.

Caché del servidor

En el caso del servidor, para este momento ya habíamos conseguido que comprima los contenidos antes de mandarlos, sin embargo, tenía que estar comprimiendo una y otra vez los archivos, sobrecarga que puede ser evitada gracias al módulo `mod_cache`.

Se habilitó el módulo `mod_disk_cache` que implementa la memoria caché basada en el almacenamiento de archivos en disco duro. Se activo la caché de disco y se especificó el directorio donde queríamos que se almacenen los archivos de caché.

Así, podemos conseguir que los documentos servidos se almacenen ya comprimidos en la caché ahorrándonos un elevado e inútil uso de CPU causado por la compresión repetidas veces de lo mismo.



- **Contraste de resultados**

Tras aplicarse las propuestas de mejoras señaladas en el punto 4, se obtuvo una reducción significativa en el tamaño de los archivos. Esto y las configuraciones del servidor para la compresión y almacenamiento de los archivos en caché ocasionaron una reducción sustancial en el tiempo de descarga las páginas.

La comparativa entre los resultados obtenidos en las pruebas de desempeño iniciales y las realizadas posteriores a la aplicación de las mejoras se presentan a continuación:

Página principal (Todas las noticias)

Visita a la página sin caché:

	Volumen de transferencia	Tiempo de transferencia.
Primera visita No optimizado	22.7 MB	4.61 s
HTML / PHP	341 kB transferidos	
JS	81.4 kB transferidos	
IMG	22.2 MB transferidos	
CSS	245 kB transferidos	
Primera visita Optimizado	1.2 MB	230 ms
HTML / PHP	20.2 kB transferidos	
JS	26.5 kB transferidos	
IMG	1.2 MB transferidos	



CSS	26.5 kB transferidos
-----	----------------------

Análisis de los resultados:

Reducción en la vista completa	
Atributo	Porcentaje de reducción
Volumen de transferencia	94.71%
Tiempo de transferencia	95.01%

Reducción en los componentes de la página	
Tipo de documento	Porcentaje de reducción
HTML / PHP	94.07%
JS	67.44%
IMG	94.59%
CSS	89.18%

Visita a la página con caché:

Siguiente visita (Con caché). No optimizado	1.5 MB	938 ms
HTML / PHP	341 kB transferidos	
JS	0 b transferidos	
IMG	1.4 MB transferidos	
CSS	0 b transferidos	
Siguiente visita (Con caché). Optimizado	20.2 kB	135 ms
HTML / PHP	20.2 kB transferidos	



JS	0 b transferidos
IMG	0 b transferidos
CSS	0 b transferidos

Análisis de los resultados:

Reducción en la vista completa	
Atributo	Porcentaje de reducción
Volumen de transferencia	98.65%
Tiempo de transferencia	85.60%

Página resultados de búsqueda (Noticias de Feed en específico: *El debate*)

Visita a la página sin caché:

	Volumen de transferencia	Tiempo de transferencia.
Primera visita No optimizado	8.2 MB	1.98 s
HTML / PHP	146 kB transferidos	
JS	81.4 kB transferidos	
IMG	7.9 MB transferidos	
CSS	245 kB transferidos	
Primera visita Optimizado	1.2 MB	192 ms
HTML / PHP	10.8 kB transferidos	
JS	26.5 kB transferidos	



IMG	1.1 MB transferidos
CSS	37.7 kB transferidos

Análisis de los resultados:

Reducción en la vista completa	
Atributo	Porcentaje de reducción
Volumen de transferencia	85.36%
Tiempo de transferencia	90.30%

Reducción en los componentes de la página	
Tipo de documento	Porcentaje de reducción
HTML / PHP	92.60%
JS	67.44%
IMG	86.07%
CSS	84.61%

Visita a la página con caché:

Siguiente visita No optimizado	146 kB	190 ms
HTML / PHP	146 kB transferidos	
JS	0 b transferidos	
IMG	0 b transferidos	
CSS	0 b transferidos	
Siguiente visita Optimizado	10.8 kB	61 ms



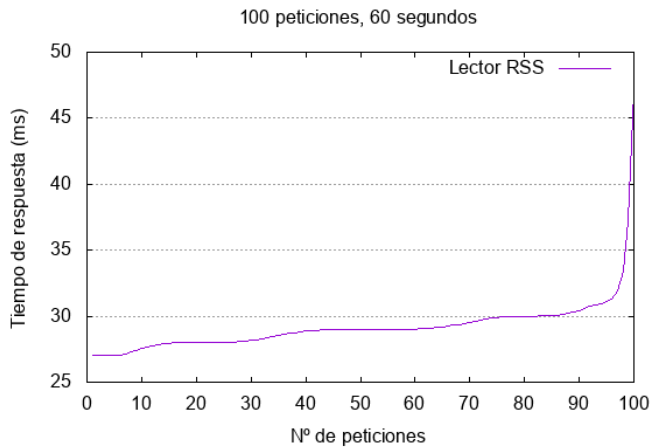
HTML / PHP	10.8 kB transferidos
JS	0 b transferidos
IMG	0 b transferidos
CSS	0 b transferidos

Análisis de los resultados:

Reducción en la vista completa	
Atributo	Porcentaje de reducción
Volumen de transferencia	92.60%
Tiempo de transferencia	67.89%

Pruebas Apache Benchmark posterior a la optimización del sitio.

Prueba 1. Página principal



Resultados *Prueba 1. Página principal:*

- Tardó 2.917 segundos en completar el envío de 100 solicitudes.
- El servidor pudo responder a todas las solicitudes sin tener alguna falla.
- El servidor pudo manejar en promedio 34.28 solicitudes por segundo.
- La velocidad de transferencia de datos fue de 5703.95 Kbytes/segundo

```
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done

Server Software:      Apache/2.4.52
Server Hostname:      localhost
Server Port:          80

Document Path:        /sitio-optimizado/todasNoticias.php
Document Length:      170058 bytes

Concurrency Level:    1
Time taken for tests:  2.917 seconds
Complete requests:    100
Failed requests:       0
Total transferred:    17039800 bytes
HTML transferred:     17005800 bytes
Requests per second:  34.28 [#/sec] (mean)
Time per request:     29.174 [ms] (mean)
Time per request:     29.174 [ms] (mean, across all concurrent requests)
Transfer rate:        5703.95 [Kbytes/sec] received

Connection Times (ms)
      min      mean[+/-sd] median    max
Connect:    0       0   0.0      0      0
Processing: 27      29   2.1     29     47
Waiting:    3       3   0.7      3     10
Total:      27      29   2.1     29     47

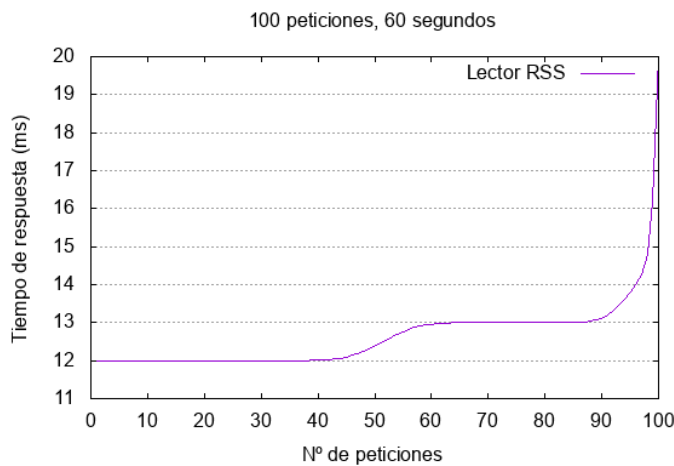
Percentage of the requests served within a certain time (ms)
 50%      29
 66%      29
 75%      30
 80%      30
 90%      31
 95%      31
 98%      32
 99%      47
100%      47 (longest request)
```

Tiempos de respuesta para las diferentes etapas de las solicitudes HTTP:

- Dado que es el proyecto se encuentra alojado localmente, no tomó tiempo en establecerse la conexión con el servidor.
- El tiempo de procesamiento de las solicitudes fue en promedio de 29ms.
- El tiempo promedio que se tuvo que esperar entre el envío de la solicitud y la respuesta fue de 3 ms.
- Finalmente, tenemos que el tiempo promedio total transcurrido desde que se inició la conexión, se recibió la respuesta y su cierre fue de 29 ms.



Prueba 2. Resultados de búsqueda (feed El debate)



Resultados Prueba 2. Resultados de búsqueda (feed El debate)

- Tardó 1.271 segundos en completar el envío de 100 solicitudes.
- El servidor pudo responder a todas las solicitudes sin tener alguna falla.
- El servidor pudo manejar en promedio 78.67 solicitudes por segundo.
- La velocidad de transferencia de datos fue de 5777.14 Kbytes/segundo

```
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient).....done

Server Software:      Apache/2.4.52
Server Hostname:      localhost
Server Port:          80

Document Path:        /sitio-optimizado/noticiasFeed.php?feed=El%20debate
Document Length:      74857 bytes

Concurrency Level:    1
Time taken for tests:  1.271 seconds
Complete requests:    100
Failed requests:       0
Total transferred:    7519700 bytes
HTML transferred:     7485700 bytes
Requests per second:  78.67 [#/sec] (mean)
Time per request:     12.711 [ms] (mean)
Time per request:     12.711 [ms] (mean, across all concurrent requests)
Transfer rate:        5777.14 [Kbytes/sec] received

Connection Times (ms)
  min  mean[+/-sd] median   max
Connect:    0    0  0.0      0    0
Processing: 12   13  0.9     12   20
Waiting:    2    2  0.6      2    8
Total:      12   13  0.9     12   20
WARNING: The median and mean for the processing time are not within a normal deviation
These results are probably not that reliable.
WARNING: The median and mean for the total time are not within a normal deviation
These results are probably not that reliable.

Percentage of the requests served within a certain time (ms)
 50%    12
 66%    13
 75%    13
 80%    13
 90%    13
 95%    14
 98%    14
 99%    20
100%    20 (longest request)
```

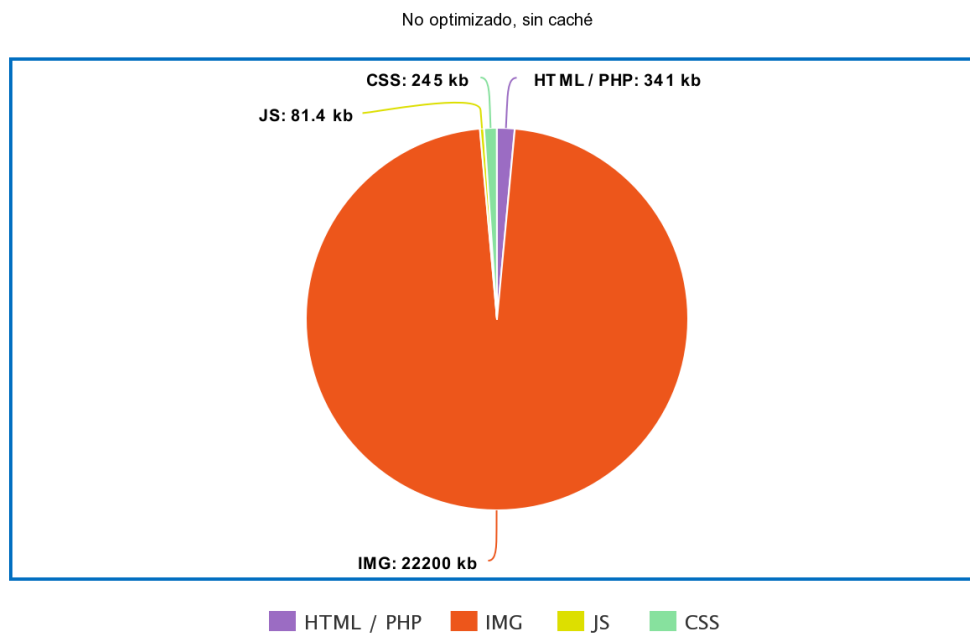
Tiempos de respuesta para las diferentes etapas de las solicitudes HTTP:

- Dado que es el proyecto se encuentra alojado localmente, no tomó tiempo en establecerse la conexión con el servidor.
- El tiempo de procesamiento de las solicitudes fue en promedio de 13ms.
- El tiempo promedio que se tuvo que esperar entre el envío de la solicitud y la respuesta fue de 2 ms.
- Finalmente, tenemos que el tiempo promedio total transcurrido desde que se inició la conexión, se recibió la respuesta y su cierre fue de 13 ms.

Tamaño de los componentes (sin caché)

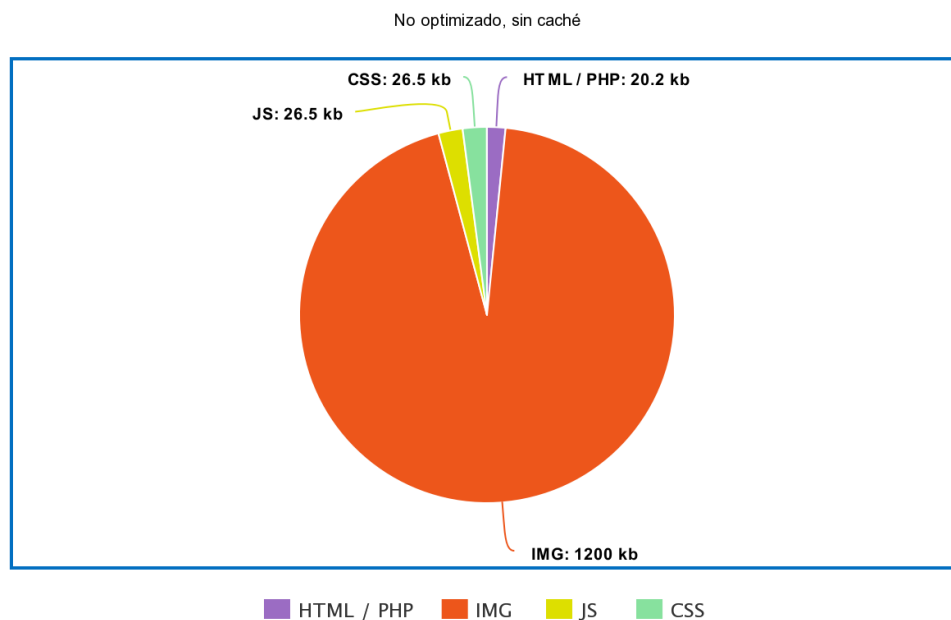
No optimizado:

Tamaño total 22.8 MB



meta-chart.com

Optimizado:



meta-chart.com



Conclusiones

Al finalizar el proyecto y después de realizar las pruebas con y sin la optimización es posible observar los cambios importantes que se tuvieron como consecuencia: el tiempo y volumen de transferencia en cada una de las páginas del Lector RSS. Para la optimización del sitio se emplearon técnicas como el ofuscamiento y minimización, la optimización de las propias imágenes, además de implementar el Sprite CSS, paginación y filtrado de las noticias, así como como el lazy loading, los cuales fueron un factor clave para lograr que en la página principal para mostrar las noticias se pudiera tener una reducción en hasta un 94.71% y 95.0% en el volumen y tiempo de transferencia respectivamente. Uno de los componentes de la página que tuvo una disminución más significativa con un 94.59% fueron las imágenes, lo cual es muy beneficioso al ser un tipo de documento muy pesado y causante de una velocidad de carga mayor.

Por otra parte, también se implemento la compresión gzip y el uso del caché, tanto para el navegador como para el servidor, causantes de que posterior a nuevas pruebas se pudiera obtener una reducción de un 98.65% en el tiempo de transferencia y un 85.60% en el volumen.

Sin duda, la implementación de técnicas de optimización son un factor muy importante para la mejora de tiempos y tamaños en estos sitios, trayendo muchos beneficios como una pagina más fluida y rápida, además de amigable para los usuarios. Principalmente para este tipo de sitios web de noticias, es crucial que los usuarios no tengan que esperar eternamente para que las noticias carguen una por una y se lleven una mala experiencia.

- **Bibliografía**

1. N/A, (2021), "¿Qué es Bootstrap y cuáles son sus ventajas?", en: https://www.hostgator.mx/blog/que-es-bootstrap/?gclid=Cj0KCQjwvqeUBhCBARIsAOdt45brNXIHIW5zjoDYq1d30rnQOLdWGKVaveD5WRfWV_iUG9HqNgDLQQoaAhfhEALw_wcB Recuperado el 18/05/2022.



UADY
UNIVERSIDAD
AUTÓNOMA
DE YUCATÁN
"Luz, Ciencia y Verdad"

2. N/A, (N/A), "¿Qué es JavaScript?", en:
https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript Recuperado el 18/05/2022.
3. <https://ourcodeworld.co/articulos/leer/503/como-habilitar-la-compresion-gzip-en-xampp-server>