



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

GRADO EN INGENIERÍA TELEMÁTICA

# OPTIMIZACIÓN DE LA VELOCIDAD DE CARGA DE UNA WEB DE NOTICIAS

Autor: Guillermo Fernández Martín

Director: Marcos Romero Rodríguez

**Madrid**

Junio 2017



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título  
Optimización de la velocidad de carga de una web de noticias  
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el  
curso académico 2016/2017 es de mi autoría, original e inédito y  
no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido  
tomada de otros documentos está debidamente referenciada.

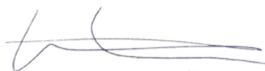


Fdo.: Guillermo Fernández Martín

Fecha: 04 / 07 / 2017

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Marcos Romero Rodríguez

Fecha: 04 / 07 / 2017

Vº Bº del Coordinador de Proyectos

Fdo.: David Contreras Bárcena

Fecha: ...../ ...../ .....



## **AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO**

### **1º. Declaración de la autoría y acreditación de la misma.**

El autor **D. Guillermo Fernández Martín** DECLARA ser el titular de los derechos de propiedad intelectual de la obra: **Optimización de la velocidad de carga de una web de noticias**, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

### **2º. Objeto y fines de la cesión.**

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

### **3º. Condiciones de la cesión y acceso**

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducir la en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

### **4º. Derechos del autor.**

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

### **5º. Deberes del autor.**

- El autor se compromete a:
  - a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
  - b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
  - c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
  - d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción

de derechos derivada de las obras objeto de la cesión.

**6º. Fines y funcionamiento del Repositorio Institucional.**

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 04 de Julio de 2017

**ACEPTA**



Fdo. Guillermo Fernández Martín



Fdo. Marcos Romero Rodríguez  
DIRECTOR DEL PROYECTO

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:



# Agradecimientos

Quisiera agradecer en primer lugar a mis padres, sin los cuales, esta realidad no sería posible. Ellos siempre han estado cuando los he necesitado y han sido un apoyo constante en estos 5 años de carrera. Aunque no empezó como teníamos previsto, no perdieron nunca la fe en mis capacidades y fueron una de las razones por las que pude continuar luchando y conseguir mis objetivos académicos.

A mi hermano, que siempre ha sido una persona en la que he podido confiar y que ha supuesto un apoyo diario.

A mis compañeros de promoción, con los cuales ha sido un placer pasar estos cuatro años de grado y espero pasar dos años más disfrutando en el master.

A mis compañeros de Europa Press, que siempre se han mostrado disponibles para ayudarme en caso de cualquier duda y que desde el primer momento me han acogido como a uno más.

A todas las personas, principalmente a mis amigos y familia, que en algún momento se han interesado por este proyecto y me han transmitido todo su apoyo.

Finalmente, me gustaría dedicárselo a una persona muy especial que ya no está entre nosotros. A la cual, durante estos dos últimos años, he echado mucho de menos cada día que conseguía un aprobado y me encontraba más cerca de graduarme.



# OPTIMIZACIÓN DE LA VELOCIDAD DE CARGA DE UNA WEB DE NOTICIAS

**Autor: Fernández Martín, Guillermo**

Director: Romero Rodríguez, Marcos

Entidad Colaboradora: Europa Press Noticias, S.A.

## RESUMEN DEL PROYECTO

Este proyecto se centra en el estudio del entorno de la página web de noticias Europa Press, en la cual se busca la optimización del sistema para conseguir una mejora del rendimiento de su velocidad de carga. Se estudiarán las diversas tecnologías disponibles y técnicas recomendadas a la vez que se llevará a cabo un estudio de rendimiento del sistema inicial. A partir de esto, se llevará a cabo el diseño de un modelo que tenga en cuenta los cambios que vamos a tener que realizar sobre el sistema, para finalmente llevar a cabo su implementación.

**Palabras clave:** HTTP, AMP, Web, Rendimiento, noticias

### 1. Introducción

En el mercado de las páginas web de noticias, existe la necesidad de desmarcarse de los competidores, otorgando a los usuarios la mejor experiencia de navegación posible. El rendimiento de carga de las webs siempre ha sido considerado como uno de los factores clave, en algunos casos como el más importante de acuerdo con las consideraciones de Nielsen [1], para el éxito de una página. La presencia de retardos del orden de los milisegundos puede suponer grandes diferencias sobre los posibles ingresos económicos [2]. Los usuarios no están dispuestos a gastar su tiempo esperando por contenidos que pueden encontrar en distintas fuentes de información, son impacientes por defecto. Por ello, se busca que la carga no entre en el rango de percepción humana. Si esto se alcanza, se incrementarán tanto el número de visitas como la retención y el *engagement* de los usuarios, lo que resulta clave en un modelo de negocio que tiene como fuente principal de ingresos las conversiones mediante anuncios. Es entonces crítico optimizar la velocidad de carga de la web.

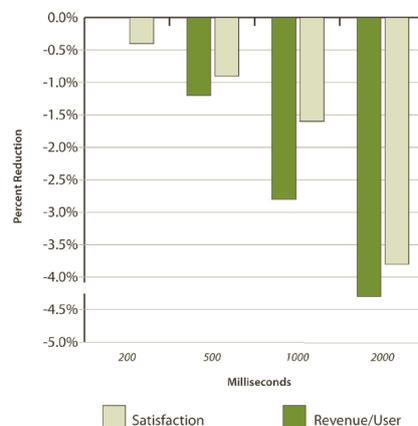


Ilustración 1. Efecto del retraso de carga en la satisfacción y los ingresos por cliente [3]

## **2. Definición del proyecto**

Los objetivos dentro de este TFG son:

- Diseño de una solución a la velocidad de la web

Este diseño vendrá precedido de un informe que haga un diagnóstico sobre qué es lo que entra en juego a la hora de conectarse a la web de Europa Press. Una vez hecho esto, se podrán considerar cuales son los principales aspectos mejorables en el diseño de esta web y proponer unas posibles medidas a implementar.

- Implementación cambios que supongan una notable mejora de la velocidad

A partir de la propuesta de mejoras de la página web, habrá que desarrollar todo lo que sea posible modificar dentro de la actual estructura de la web, para conseguir mejoras que sean lo suficiente buenas para apreciarse a nivel del ojo humano.

- Estudio de resultados

A partir de los cambios que se hayan podido efectuar, habrá que realizar de nuevo un diagnóstico para ver cómo han afectado estas mejoras a la web y decidir si mantener los cambios realizados. Aquí entrará en juego ver en qué medida se han podido satisfacer los resultados que se esperaban obtener. Como ya se ha explicado anteriormente, uno de los motivos por los que tan importante este estudio de optimización dentro de la página es que la mejora que se haya podido obtener repercuta en la experiencia de usuario y pueda suponer beneficios en algún otro aspecto de web, como por ejemplo en términos de SEO.

- Realizar un plan de mejoras futuras

Al estar trabajando dentro de un plazo de tiempo tan pequeño, van a ser difíciles realizar los cambios más ambiciosos dentro de la web, es por eso que después de realizar todas estas modificaciones será necesario volver a hacer un diagnóstico de la web. Será acertado desarrollar una nueva propuesta que sugiera unas mejoras futuras a partir de la valoración del proyecto sobre los cambios pendientes, las cosas que no han funcionado dentro de las mejoras implementadas y recogiendo nuevas mejoras posibles que se puedan haberse considerado al final del proyecto.

## **3. Descripción del sistema**

El sistema de mejoras que se ha desarrollado puede dividirse en dos partes. Por un lado, tenemos los cambios que van a repercutir en la mejora de la velocidad que experimente la gente que consulte la web desde un dispositivo de sobremesa y del otro lado los visitantes que usen un móvil para navegar.

En el sistema desktop hemos desarrollado lo siguiente:

- Se ha realizado una renovación de los formatos de las distintas portadas de la web. A la hora de hacer el análisis se podía fácilmente apreciar que no se estaba obteniendo en ellas

un formato que facilitase a una experiencia de usuario rápida. Es por esto que se ha decidido cambiar y unificar formatos, lo que también mejorará la facilidad de *troubleshooting*, que existía debido a la gran diversidad de formatos anteriores, para los que se va a tener en cuenta un diseño enfocado a simplicidad y cumplir los cánones recomendados para una buena velocidad de carga.

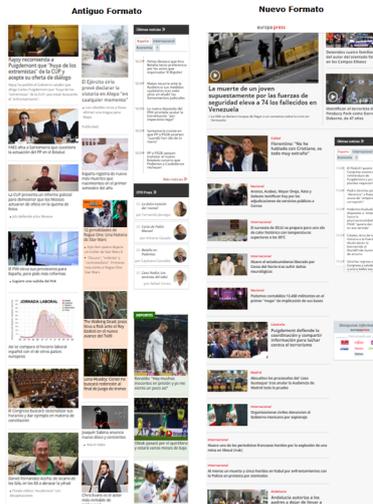


Ilustración 2. Comparativa entre modelos de la portada

- Vamos a hacer uso de la tecnología JavaScript para reducir gran parte de la carga de datos introducida por la necesidad de cargar imágenes, problemática muy común en las distintas webs de noticias, que siempre llevan una gran carga de imágenes. Mediante el uso de la técnica llamada Lazy Load, conseguiremos evitar que las imágenes se carguen hasta que se sepa que el usuario tiene una alta probabilidad de hacer *scroll* hasta la posición de cada una de ellas.
- Uno de los puntos más importantes en nuestro estudio teórico ha consistido en el análisis de las limitaciones que conlleva el uso de HTTP/1 y como ya existe el protocolo HTTP/2 que resuelve los defectos y provoca la mejora del rendimiento de descarga de una web [4]. Es por esto que vamos a realizar la migración de los orígenes de descarga de distintos recursos. Esto lo vamos a realizar mediante el uso de un CDN para la distribución de los recursos de la web. Ha sido necesario hacer un estudio de mercado de las distintas plataformas de CDN disponibles en el mercado para elegir la opción más competente. Este cambio nos va a permitir transmitir nuestros recursos mediante sockets seguros con HTTPS y que empleen el protocolo HTTP/2. Este cambio también repercutirá para la versión móvil puesto que es sobre los recursos, como pueden ser las imágenes y las hojas de estilo.

Para la versión móvil hemos desarrollado lo siguiente:

- Existía un problema con el uso de CSS, por el cual se cargaban muchos datos inútiles nada más conectarse a la web, de media un 75% del contenido de los CSS en móvil no llegaban a ser usados, lo que demoraba la carga del primer punto de vista. Por esta razón, se ha desarrollado un sistema de CSS inline que de manera dinámica cargue los estilos del primer punto de vista del navegador al principio del HTML, para que su carga se

haga más rápidamente y el usuario no pueda observar que no se han cargado el resto de los estilos, los cuales se cargan al terminar la carga de la web.

- AMP es un *framework* por el que Google está apostando de manera fuerte, como se puede ver por la primera edición de la conferencia Google AMP que se ha producido este año. Por ello vamos a implementar un formato de noticias que corra por este *framework*, que promete una gran velocidad de carga mediante su uso.



*Ilustración 3. Ejemplo de una noticia de Europa Press en el formato AMP*

- Uno de los problemas del uso de las imágenes en móvil es la diversidad de dispositivos que existen. Por ello vamos a implementar un sistema de imágenes que sean realmente responsivas. Queremos evitar el uso de una imagen genérica para todos los casos que sea de gran tamaño y tenga que ser reescalada hacia abajo o, en el peor de los casos, que está imagen sea menor del tamaño necesario y sea reescalada hacia arriba, lo que resulte en mala calidad de imagen. Esto lo vamos a generar mediante el *tag picture* en el cual vamos a introducir muchas fuentes distintas para que se escoja la mejor, según el tamaño y la nitidez de la pantalla del dispositivo. En este sistema también vamos a implementar imágenes con extensión WebP, para que en los casos que el navegador sea compatible, se use esta extensión de mayor calidad y menos tamaño.

#### **4. Resultados**

A partir de la implementación de este sistema obtenemos diversas mejoras en las métricas de carga de las distintas versiones:

	Versión móvil	Versión móvil AMP (lo comparamos con la versión móvil estándar antigua)	Versión Desktop
Tiempo carga global versión antigua	10,368 s	10,368 s	14,152 s
Tiempo carga global versión versión mejorada	7,651 s	5,545 s	10,368
Mejora	25%	46%	27%
<b>MEJORA MEDIA EN LA WEB</b>		<b>33%</b>	

Tabla 1. Resumen global de resultados de carga total

- Mejoría en un 30% del tiempo de carga de la portada home de Europa Press en desktop
- Hemos mejorado un 25% la carga de la web en la versión móvil
- La versión AMP supone una mejoría del 45% con respecto a la versión móvil antigua y un 26% mejor que la nueva
- También nos encontramos con mejoras fuera de la velocidad de carga, como una mejoría en SEO gracias al nuevo formato de portada y a la adopción de AMP, a la vez que un sistema más seguro por la migración a HTTPS de ciertos recursos

## 5. Conclusiones

La realización de este sistema de mejoras ha supuesto cumplir los objetivos que estaban fijados. La web de Europa Press ha mejorado, por lo que podemos esperar que así lo hagan los resultados de visitas y de ingresos por publicidad.

Aunque se hayan realizado estos cambios, todavía existe margen de mejora, ya que la duración del proyecto no ha permitido optimizar completamente el sistema. Se propone entonces una serie de trabajos futuros sobre lo realizado, que supongan que este sistema migre completamente a HTTPS y al uso de HTTP/2, que se haga uso del *framework* LESS para la gestión de hojas de estilo y que se cree una versión AMP de la totalidad de los formatos de la web y no solamente de las noticias.

## 6. Referencias

- [1] Fiona Fui-Hoon Nah, "A study on tolerable waiting time: how long are Web users willing to wait?" College of Business Administration-University of Nebraska - Lincoln (2004)
- [2] DoubleClick by Google, "The Need For Mobile Speed" (2016)
- [3] Gomez, "Why Web Performance Matters: Is Your Site Driving Customers Away?" (2010)
- [4] Ilya Grigorik, "High Performance Browser Networking" - O'Reilly (2013)



# NEWS WEBSITE LOADING SPEED OPTIMIZATION

**Author: Fernández Martín, Guillermo**

Supervisor: Romero Rodríguez, Marcos

Collaborating Entity: Europa Press Noticias, S.A.

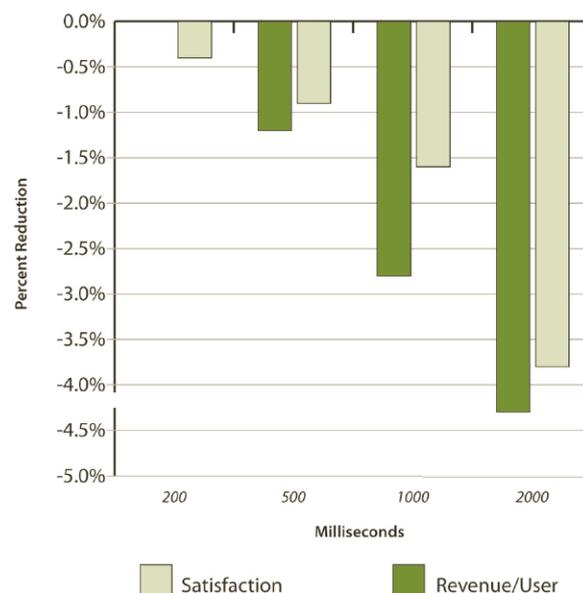
## ABSTRACT

This project aims to study Europa Press website's environment to optimize its system, in order to get the highest download speed possible. The latest available technologies and recommended techniques are going to be studied at par with an analysis on where the web performance stands at the start of this project. From this, a model on which are the possible modifications to be carried out will be designed, to at last, bring its development into reality.

**Keywords:** HTTP, AMP, Web, Performance, News.

## 1. Introduction

News Websites market is loaded with a wide number of media that provide readers a huge range of choice. As we are on a very competitive environment, the need to gain notoriety by bringing a cutting-edge user experience is vital. Website performance as always been considered as one of the key factors, if not the single most important according to Nielsen [1], to have a successful web-page. Having delays of only a couple of milliseconds translates in economic income differences [2] as users are not willing to wait in order to gain access to content that is available all around the web, they are impatient by nature. Therefore, the aim is to avoid having a loading time slow enough to enter human perception time range. Staying under this threshold would translate into an increase in the number of visits, stickiness and user engagement, which is key on a business model consisting on advertisement revenue as its income source. Thus, optimizing our web-page's loading speed is critical in this context.



*Illustration 1. Effects on user satisfaction and income in function of time delay [3]*

## **2. Project definition**

This project has as objectives:

- Web speed improvement model

This design will be preceded by a diagnosis taking into accounts which are the main factors that have an influence when some device tries to connect to our webpage. As soon as this will be made, we can identify which aspects of our web have an improvement margin and will make a proposal on how to improve them with a changes model proposal.

- Model implementation

From the changes suggested on the proposal, every one that fits into our system will have to be developed, in order to improve the system. The improvements made by this development are expected to be big enough to be noticeable by only making a visual comparison between the early stage and the final status.

- Analysis of results

Having made a few changes in the system, an in-depth analysis of every one of those will be needed in order to quantify how much we are benefiting from them, at the same time that we have to decide if they are not needed and revert those who doesn't. We will then see if the expectations are being met, alongside with some improvements on customer experience and other possible positive side effects on a diversity of webpage's stats such as SEO.

- Future work proposal

This project takes a short time span, which is a drawback as we are working on developing an optimization project. All the possible modifications from which this system could have benefited from are not going to be developed. So, having made an analysis of results on how the system is at the end of this project, we can still detect some flaws which set us apart from having a totally optimized system. It will then be necessary to leave a proposal on which can be the future works carried out in this website, as a follow-up to all the work developed during these months.

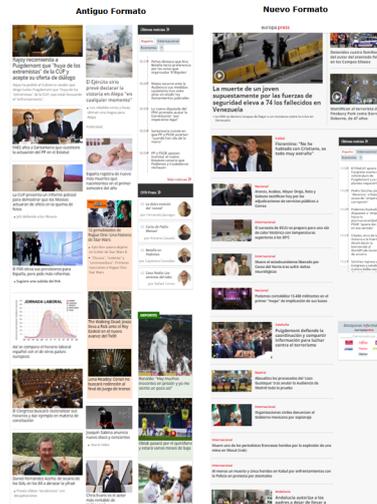
## **3. System Description**

The improved system can be divided into two, on the one hand, we have the modifications affecting the desktop version from the web, and in the other, the version that is displayed when someone access the website from a handheld device such as a smartphone.

For the desktop version, we have the following changes:

- All the home pages from the different channels on the website have met a renewal, that unifies them into a new format, which improves the flaws those previous models had. Those formats where not aiming for a good download speed as they were designed long time ago. For that reason, we have chosen to unify all these formats

on a single one, which will help easing any need of troubleshooting, as there will no longer be a large number of possible sources of errors. This new format will follow the studied guidelines for having a good download speed and aiming for a minimalist design



*Illustration 2. Home Page format comparison*

- We are going to make use of Javascript technology to reduce the amount of data required to be downloaded, by the need for image loading, a common problem among news websites, which usually have a huge amount of image content. By using a function called Lazy Load, the images will not be loaded until there exists a huge probability for the user to scroll to the position of every single one of those.
- On our theoretical study, one of the main points treated consists on how HTTP/1 is an outdated protocol that has a great number of limitations that can be solved by the use of its latest version, HTTP/2, at the same time downloading performance is increased [4]. Therefore, we need to make a migration of the sources where our content is downloaded from. We are going to make use of a CDN, which will be chosen with a market study. By this migration, our resources will be transmitted through encrypted sockets, using HTTPS, and will benefit from the use of HTTP/2 protocol. This migration will also affect the mobile version.

For the mobile version, we have the following developments:

- There existed a problem with how CSS was used that caused a huge load of data to be downloaded by the start of the HTML document. Most of its data was not needed, as on average 75% of the classes were not used, causing a delay on the load of the first viewport. We have then developed a system which includes CSS inline at the

head of the HTML, replacing the use of CSS files at the header. The styles included on this CSS inline are only the ones required on the first viewport for the user to not appreciate that not all the classes have been loaded, and to having this viewport loaded sooner. The rest of the styles will be loaded in the footer of the HTML.

- AMP is a framework being developed by Google, which is being hugely promoted as can be perceived with the first Google AMP conference having been held this year. Therefore, we are going to develop an AMP version to display our news on smartphones, which will also benefit from a good performance, as Google is preaching.



*Illustration 3. New on AMP format*

- One problem we have with the use of images on handheld devices is the great range of devices that are available on the market. We are then developing a true responsive image delivery system. We want to avoid the need of downgrading or upgrading an image, to fit the dimensions of each device, which creates a need to download more data than needed on the first case and a bad image quality on the second. We are doing this by using the tag picture alongside a list of different sources which will provide the convenient image every time, depending on the size of the device and the sharpness of its display. This will be combined with the use of WebP image format, which if the browser can run it, provides better quality images at a lower data size.

#### **4. Results**

By the end of this implementation, we meet some performance improvements on the diverse versions of the website, which are:

	Mobile version	Mobile AMP version (compared with the former mobile version)	Desktop version
Former total download time	10,368 s	10,368 s	14,152 s
Total download time for new version	7,651 s	5,545 s	10,368
Improvement rate	25%	46%	27%
<b>AVERAGE IMPROVEMENT RATE</b>		<b>33%</b>	

*Table 1. Total download time results breakdown*

- Europa Press Homepage takes 30% less time to be downloaded
- Mobile version of the site has improved its downloading speed by 20%
- AMP performs 45% faster than the old mobile version and 26% better than the improved one
- We also have improvements outside downloading performance, with an improved SEO, thanks to the AMP system and the new formats developed, and a more secure website having most of its resources being delivered by HTTPS

## 5. Conclusions

By developing this improved system, we have met the different objectives set by the start of this project. Europa Press' website has improved enough to have this difference be noticed by a simple visual comparison, so we are expecting for the number of visitors and the income generated to increase.

However, having an improved environment doesn't mean that we have met optimal conditions, as time as not been enough for this. Thus, we propose several future works as a follow-up from this project, which are, a total migration of the system into the use of HTTPS and HTTP/2, the development of a dynamic management system of style sheets with the LESS framework and the creation of an AMP version of the whole page, involving all the formats existing on desktop and not only the news.

## 6. References

- [1] Fiona Fui-Hoon Nah, "A study on tolerable waiting time: how long are Web users willing to wait?" College of Business Administration-University of Nebraska - Lincoln (2004)
- [2] DoubleClick by Google, "The Need For Mobile Speed" (2016)
- [3] Gomez, "Why Web Performance Matters: Is Your Site Driving Customers Away?" (2010)
- [4] Ilya Grigorik, "High Performance Browser Networking" - O'Reilly (2013)



## *Índice de la memoria*

<b>Capítulo 1. Introducción.....</b>	<b>9</b>
1.1 Motivación del proyecto .....	9
1.2 EUROPA PRESS NOTICIAS S.A. ....	10
1.3 Historia de internet y de los medios de comunicación.....	12
<b>Capítulo 2. Descripción de las Tecnologías.....</b>	<b>15</b>
2.1 Herramientas de desarrollo .....	15
2.1.1 Visual Studio .....	15
2.1.2 GIT .....	15
2.1.3 Amazon Web Services .....	16
2.2 Webep .....	16
2.3 Servidores.....	17
2.4 Herramientas de análisis .....	25
2.4.1 WebPageTest.....	25
2.4.2 Google Analytics .....	26
2.4.3 Chrome DevTools.....	26
2.4.4 BrowserStack.....	27
<b>Capítulo 3. Estado de la Cuestión .....</b>	<b>28</b>
3.1 Limitaciones HTTP/1.....	28
3.2 HTTP/2 .....	31
3.3 AMP .....	34
<b>Capítulo 4. Definición del Trabajo.....</b>	<b>38</b>
4.1 Justificación .....	38
4.2 Objetivos .....	39
4.3 Metodología .....	40
4.4 Planificación y Estimación Económica.....	42
4.4.1 Planificación Inicial.....	42
4.4.2 Estimación Económica.....	44

<b>Capítulo 5. Diagnóstico web</b> .....	<b>46</b>
5.1 Estudio tráfico.....	46
5.2 Estudio rendimiento.....	49
5.2.1 Versión Desktop.....	50
5.2.2 Versión Móvil.....	54
5.3 Competencia.....	56
<b>Capítulo 6. Sistema Desarrollado</b> .....	<b>60</b>
6.1 Análisis del Sistema.....	60
6.2 Diseño.....	64
6.3 Implementación.....	66
6.3.1 Formato nuevo.....	66
6.3.2 Lazy load.....	71
6.3.3 HTTPS.....	73
6.3.4 CSS inline.....	77
6.3.5 AMP.....	81
6.3.6 Gestión de imágenes.....	85
<b>Capítulo 7. Análisis de Resultados</b> .....	<b>91</b>
7.1 Resultados versión Desktop.....	91
7.2 Resultados versión Móvil.....	96
7.3 Recopilación de totalidad de resultados.....	105
<b>Capítulo 8. Conclusiones y Trabajos Futuros</b> .....	<b>106</b>
8.1 Conclusiones.....	106
8.2 Trabajos futuros.....	107
<b>Capítulo 9. Bibliografía</b> .....	<b>110</b>
 <b>ANEXO A</b>	<b>115</b>
 <b>ANEXO B</b>	<b>121</b>
 <b>ANEXO C</b>	<b>126</b>



## *Índice de figuras*

Figura 1. Efecto del retraso de carga en la satisfacción y los ingresos por cliente [1] .....	9
Figura 2. Logotipo de Europa Press S.A.....	11
Figura 3. Canal de Fundación Unicaja en europapress.es .....	12
Figura 4. Evolución del número de usuarios de internet en el mundo según datos de la ITU [4].....	13
Figura 5. Métricas de tamaño del código.....	17
Figura 6. Esquema de arquitectura de servidores .....	19
Figura 7. Diagrama de flujo de tratamiento de requests .....	23
Figura 8. Plataforma de gestión de cluster de servidores.....	24
Figura 9. Tiempo de carga en relación con la latencia y el ancho de banda [10] .....	28
Figura 10. Ejemplo de paralelismo en HTTP/1 .....	29
Figura 11. Ejemplo de HOL Blocking por retardo de descarga de script [10] .....	30
Figura 12. Estructuras de datos de Http/1 y Http/2[10] .....	32
Figura 13. Ejemplo de comunicación HTTP/2[10].....	33
Figura 14. Compresión de cabeceras mediante HPACK en HTTP/2 [10] .....	33
Figura 15. Planificación inicial del proyecto .....	43
Figura 16. Distribución del número de visitas en función del dispositivo en la web de Europa Press (15/01/2017 hasta 15/02/2017).....	47
Figura 17. Distribución del tiempo global de duración de sesión en función del dispositivo en la web de Europa Press .....	48

---

Figura 18. Crecimiento del tráfico de datos en función del dispositivo [61].....	49
Figura 19. Captura del momento de carga del DOM y estilos.....	51
Figura 20. Desglose del contenido de la página .....	51
Figura 21. Detalle de los 25 primeros elementos.....	52
Figura 22. Detalles de una request correspondiente a una imagen .....	53
Figura 23. Resultado PageSpeed Insight .....	54
Figura 24. Fotograma estudio de rendimiento noticia móvil antigua .....	55
Figura 25. Móvil PageSpeed insight.....	56
Figura 26. HTTPS del Washington Post.....	58
Figura 27. Request seguras en Washington Post .....	58
Figura 28. Request de una imagen del Washington Post.....	59
Figura 29. Estructura .master-.aspx[13].....	61
Figura 30. Formatos portada Europa Press .....	67
Figura 31. Estructura de Portales y Canales del grupo Europa Press .....	68
Figura 32. Cambio formato portada del canal Turismo .....	70
Figura 33. Estructura abstracta de un CDN .....	74
Figura 34. Comparativa CDNs .....	76
Figura 35. Favicon de Europa Press .....	77
Figura 36. Google Audit sobre la portada móvil .....	78
Figura 37. Archivos implementación versión AMP de la web.....	83
Figura 38. Ejemplo de noticia en formato AMP.....	84
Figura 39. <amp-ad> dentro del contenido de una noticia .....	84
Figura 40. Imagen para móvil con WebP .....	86

---

Figura 41. Imagen JPG .....	87
Figura 42. Compatibilidad del formato WebP con los diversos navegadores [18] .....	88
Figura 43. Fotograma estudio de rendimiento de la nueva portada de Desktop.....	92
Figura 44. Análisis de contenido en la carga de la nueva portada de Desktop.....	93
Figura 45. Conexiones en la nueva portada desktop.....	94
Figura 46. Request HTTP/2 .....	95
Figura 47. Análisis WebPage Tools Insights sobre la nueva portada desktop .....	95
Figura 48. Fotograma estudio de rendimiento noticia móvil nueva .....	97
Figura 49. Análisis WebPage Tools Insights sobre la nueva portada desktop .....	98
Figura 50. Resultado de la demo comparativa del tiempo de carga de las dos versiones de las noticias para móvil (AMP contra m.).....	99
Figura 51. Ejemplo de un anuncio amp mal empleado.....	101
Figura 52. Noticias AMP de europapress indexadas en Google.....	102
Figura 53. Análisis de las fuentes de tráfico .....	103
Figura 54. Galería de noticias AMP cacheadas por Google .....	104

## *Índice de tablas*

Tabla 1. Leyenda de la figura 4 .....	21
Tabla 2. Costes de personal .....	44
Tabla 3. Costes materiales .....	45
Tabla 4. Categorización de las visitas (15/01/2017 hasta 15/02/2017).....	46
Tabla 5. Categorización de la duración de sesiones .....	48
Tabla 6. Estudio rendimiento portada desktop inicial .....	50
Tabla 7. Datos rendimiento de carga noticia móvil antigua .....	55
Tabla 8. Análisis de la competencia. ....	57
Tabla 9. Comparativa de distintos CDNs .....	75
Tabla 10. Estadísticas de uso del contenido CSS descargado .....	79
Tabla 11. Resultados test nueva portada desktop .....	91
Tabla 12. Datos rendimiento de carga noticia móvil antigua .....	96
Tabla 13. Comparativa de tiempos de carga entre AMP y móvil.....	100
Tabla 14. Resumen global de resultados de carga total.....	105



## Capítulo 1. INTRODUCCIÓN

### 1.1 MOTIVACIÓN DEL PROYECTO

Desde los inicios del desarrollo web, la velocidad de descarga de las páginas web ha sido uno de los mayores criterios para evaluar la calidad de estas. Siempre han existido criterios que creaban un canon de calidad dentro de Internet, en el apartado del rendimiento web la regla de los ocho segundos era la más extendida. Esta regla especificaba que cualquier página web que sobrepasaba ese tiempo de carga no iba a ser aceptable, en cuanto que suponía una mala experiencia para los usuarios. En la actualidad esa regla no es un canon global. Es más interesante seguir distintos estudios sobre el comportamiento humano, como el que introduce la teoría de como las personas suelen perder la atención en periodos de carga desde los 2 segundos, según la teoría que introduce Miller en el año 1968, basándose en la capacidad humana de procesar la información.

Esto supondría el límite de lo que es apreciable en cuanto a la percepción humana, por eso tenemos que tener en cuenta que incluso en los márgenes más cortos, del orden de los milisegundos, se pueden multiplicar los beneficios se encuentran en variaciones del orden del milisegundo. Podemos ver cómo influyen estas variaciones en la Figura 1:

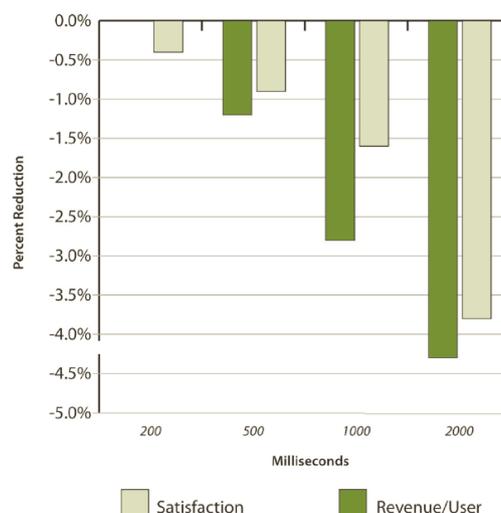


Figura 1. Efecto del retraso de carga en la satisfacción y los ingresos por cliente [1]

Es por esto que todas las grandes empresas buscan que sus webs sean lo más rápidas posibles, dejándose influir por diversos estudios, como pueden ser los realizados por parte de Google [2], donde se llega a la conclusión de que, trabajando sobre una misma web, un mayor tiempo de carga, por muy pequeño que sea, supone una pérdida tanto en el tráfico de dicha web como en la *stickiness* de los navegantes. Este último es de una gran importancia dentro del contexto en el que este TFG se desarrolla, puesto que estamos en una web de noticias. El concepto de *stickiness* se refiere al tiempo que invierte un usuario dentro de una página web, lo cual se traduce en una mayor conversión de anuncios y visitas a través de los diferentes *sites* de la página. Esto provoca que existan grandes márgenes de ingresos conseguidos a partir de la optimización de la velocidad de carga, con mejoras que suponen reducciones en los tiempos de cargas del orden de pequeños márgenes de tiempo.

Dentro de la web de EP, existe una gran serie de formatos que llevan en producción mucho tiempo, los cuales se consideran algo obsoletos y se prevé que sufran una renovación. Todo cambio supone muchas dificultades, por lo que se necesita asegurar que estos cambios no introducen fallos de rendimiento y están optimizados. Al trabajar con una estructura tan grande, con un equipo de muchas personas suelen aparecer diversas problemáticas, las cuales no suelen ser supervisadas con la atención que requieren, por lo cual es necesario realizar un estudio que nos permita detectar qué elementos pueden ser modificados para optimizar la velocidad de la web.

## ***1.2 EUROPA PRESS NOTICIAS S.A.***

Europa Press Noticias S.A. es la segunda agencia de noticias más grande de España, por detrás de la Agencia EFE, la agencia oficial del estado cuya titularidad es de orden público, lo que posiciona a nuestra empresa como la primera entre las agencias privadas del país. Su modelo de negocio se basa en la venta de noticias a los diferentes medios de comunicación, ya sea desde la venta de artículos redactados hasta imágenes obtenida por los fotógrafos de la agencia que puedan ser interesantes para aportar contenido visual para una publicación de algún medio. Es un negocio de venta de todo tipo de información para las distintas

publicaciones del país. Pero, además de facilitar noticias al resto de medios, también ejerce de medio de comunicación de forma propia mediante la página web oficial de la empresa, donde se publican las noticias que redactan el cuerpo de periodistas.



*Figura 2. Logotipo de Europa Press S.A.*

Esta web se financia mediante la obtención de acuerdos publicitarios y patrocinadores, a la vez que ayuda a la exposición de la calidad de los artículos para las publicaciones interesadas en convertirse en abonadas del servicio de venta de información. Los primeros pagan por introducir anuncios dentro del contenido de la web mientras que los patrocinadores buscan que el equipo de desarrollo establezca plataformas específicas para la difusión de contenidos relacionados con sus actividades privadas dentro de la propia web, por ejemplo el Banco Unicaja tiene un acuerdo con Europa Press por el cual existe un canal de noticias completamente dedicado a todas las campañas sociales que desarrollan en su fundación para facilitar su difusión, el servicio por el cual paga esta empresa es por crear contenido relacionado con su entidad en la misma estructura de la página web. Es aquí donde entra en juego este TFG ya que cuanto mayor sea el éxito de este trabajo, mayor serán los beneficios para el modelo de negocio en el que se basa la página web de la empresa.



Figura 3. Canal de Fundación Unicaja en europapress.es

En cuanto a los números que se manejan en esta web, según el ranking de tráfico web de Alexa Internet Inc., la web de Europa Press se encuentra en el puesto número 192 en España. A partir de los datos recolectados con la herramienta Google Analytics, podemos saber que la página de Europapress recibe de manera mensual 14M de sesiones que generan 27M visitas a páginas (un promedio de 1,89 páginas por sesión) con una duración media de 6 minutos por cada sesión que se crea.

### 1.3 HISTORIA DE INTERNET Y DE LOS MEDIOS DE COMUNICACIÓN

Internet es una realidad muy presente dentro de la vida de la mayoría de personas, en los países desarrollados la ITU considera que el 80% de la población usa Internet [3], tenemos millones de dispositivos conectados a esta red global de comunicaciones y gran número de las actividades que ejercen las personas dentro del mundo actual se basan en la necesidad de este soporte para realizarlas. Su creación se remonta a los años 1980s, procede del desarrollo de un proyecto de conectividad entre distintas redes independientes, por el cual el Departamento de Defensa de los Estados Unidos buscaba conectar diferentes organismos tanto de orden académico como de orden militar. Esta red se llamaría ARPANET, debido a ser desarrollada en la agencia de investigación de proyectos avanzados o ARPA. En el 29 de

octubre de 1969, se establece la primera comunicación dentro de esta red, mediante una conexión entre la universidad de UCLA y el instituto de investigaciones de Stanford, que evolucionaría hasta llegar a contar con 500 computadoras conectadas en 1983. En ese mismo año se adoptó dentro de esta red el protocolo TCP/IP, protocolo en el cual se basa las comunicaciones dentro de Internet en la actualidad. No será hasta el año 1990 que entren en juego las páginas web.

Gracias a la invención de la *World Wide Web*, red informática mundial o (WWW), se creó la oportunidad de visualizar páginas web gracias al uso de navegadores web. La popularidad global de estos navegadores llegará en torno a finales de los 90, donde entran en juego los web browsers de difusión general masiva, los cuales tendrán una gran competencia entre ellos, como demuestra la guerra de navegadores que existió en los inicios de Internet Explorer, el cual fue creado por Microsoft como herramienta para desbancar al navegador *Netscape Navigator*. Estos permitían tener alcance a contenido web sobre Internet al público general, a partir de este modelo el número de usuarios comenzó a incrementarse de manera drástica hasta llegar a los niveles actuales de tráfico web que se estiman en aproximadamente 3500 millones de usuarios en 2016 [4].

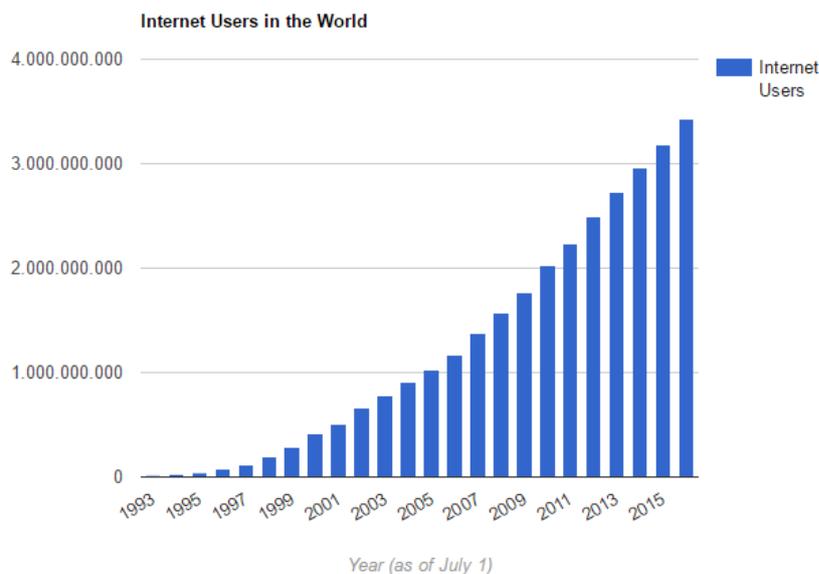


Figura 4. Evolución del número de usuarios de internet en el mundo según datos de la ITU [4]

Esta realidad de Internet ha tenido un impacto en la forma de trabajar de los medios de comunicación y sobre todo de la prensa escrita. Los periódicos siempre han estado relacionados con un formato a papel, el cual se compraba por la mañana y se tenía la información del día anterior condensada en un periódico de papel. Esta realidad ha dado un vuelco de 360 grados ya que en la actualidad la gente consume las noticias en el momento, ya no se quiere encontrar en los medios escritos las noticias de las últimas 24 horas, sino que los usuarios están cada vez más acostumbrados a consumir el contenido en tiempo real, con una experiencia casi a tiempo real. La última hora ha pasado de contener elementos de un periodo de 24 horas, a un punto en el que se conjuga con el tiempo presente. Internet tiene gran importancia en esto ya que pone a disposición del público la posibilidad de estar conectado en cualquier circunstancia físico-temporal con las entidades informativas. Ahora los diarios impresos no llegan a ser tan importantes como las páginas webs de esos mismos medios. Estas webs proveen una plataforma gratuita en la cual se tiene un acceso rápido y ágil a las noticias que más interesan al público general, en oposición al formato escrito por el cual se debe pagar por toda la información aun no estando el consumidor interesado en toda esta. Ha sido en estos últimos años que el mercado de los anuncios online ha sufrido una evolución considerable, como muestra el un estudio del sector provisto por Deloitte [5], que señala que entre el año 2008 y 2013, el mercado de los anuncios online en los medios se incrementó en un 47%. Por su parte, un estudio de PwC [6], estima que los ingresos por medios digitales llegarán a representar el 30% de todos los ingresos en los medios en 2020, por el contra del 16% que suponían en el año 2015. Se puede entonces deducir como los medios de noticias deben de tener en cuenta, en sus modelos de negocio, la necesidad de tener una plataforma digital que contenga sus noticias, buscando un retorno de inversión a partir de la inclusión de contenido patrocinado.

## Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

### 2.1 HERRAMIENTAS DE DESARROLLO

#### 2.1.1 VISUAL STUDIO



La página web de Europa Press se basa en el lenguaje VisualBasic, perteneciente al *framework* .NET desarrollado por Microsoft, alternativa principal en el entorno de desarrollo web al lenguaje Java. Por ello, en el equipo de desarrollo web se emplea el IDE Visual Studio, también desarrollado por Microsoft, usando su versión profesional.

#### 2.1.2 GIT



Para la gestión de versiones, entre los diversos miembros del equipo, empleamos el sistema *open source* Git. Este sistema permite trabajar de manera independiente a los distintos miembros del equipo de desarrolladores con versiones distintas del código y unir, a la hora de realizar cambios, las modificaciones deseadas en una versión común, llamada producción, disponible en los distintos servidores de difusión de la página.

### 2.1.3 AMAZON WEB SERVICES



Amazon Web Services es un servicio de localización de recursos en la nube. Se usa para que haga de host de distintos recursos, necesarios a la hora de conectarse a la web, para liberar carga pesada sobre los servidores que contienen el código fuente de la web. Al estar en una página web de noticias, hay una gran carga de imágenes por lo que se usa este servicio fundamentalmente para emplazar allí todas las imágenes de las diversas noticias de la web, a parte de otros recursos estáticos.

## 2.2 WEBEP

La web de Europa Press se basa en el *framework* .NET desarrollado por Microsoft, utilizando el lenguaje de programación Visual Basic. La arquitectura de este proyecto web sigue el patrón MVC. Este se caracteriza por una estructura distribuida en 4 capas:

- Datos: En esta capa se define toda la lógica que está relacionada con el acceso y la gestión de la base de datos.
- Modelos: Esta capa da la lógica necesaria para gestionar todos los datos de la web, instanciando diversas clases y funciones esenciales en la lógica de la web
- Negocio: En esta capa se gestiona la comunicación entre la capa de la vista de la web y a la capa de datos

- Vista Web: En esta capa se instancian todos los elementos que caracterizan el modo en el que se pinta la página web además de contener diversa lógica.

Estamos trabajando con un proyecto de gran tamaño como podemos ver en la Figura 5, con un total de más de un millón de líneas de códigos.

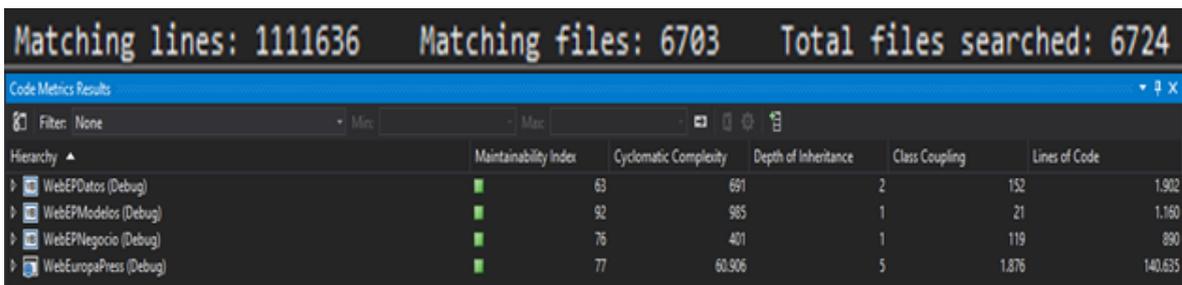


Figura 5. Métricas de tamaño del código

En este proyecto, nuestro trabajo se va a centrar en trabajar con el proyecto llamado WebEuropapress, que en el patrón MVC se corresponde con la capa de la vista.

## 2.3 SERVIDORES

Además de interesarnos al código fuente de la web, nos interesamos a conocer la arquitectura servidores web, lo que permite comprender mejor el funcionamiento de cómo se procesan las peticiones cuando alguien se quiere conectar a nuestra página web, lo que está altamente relacionado con su rendimiento.

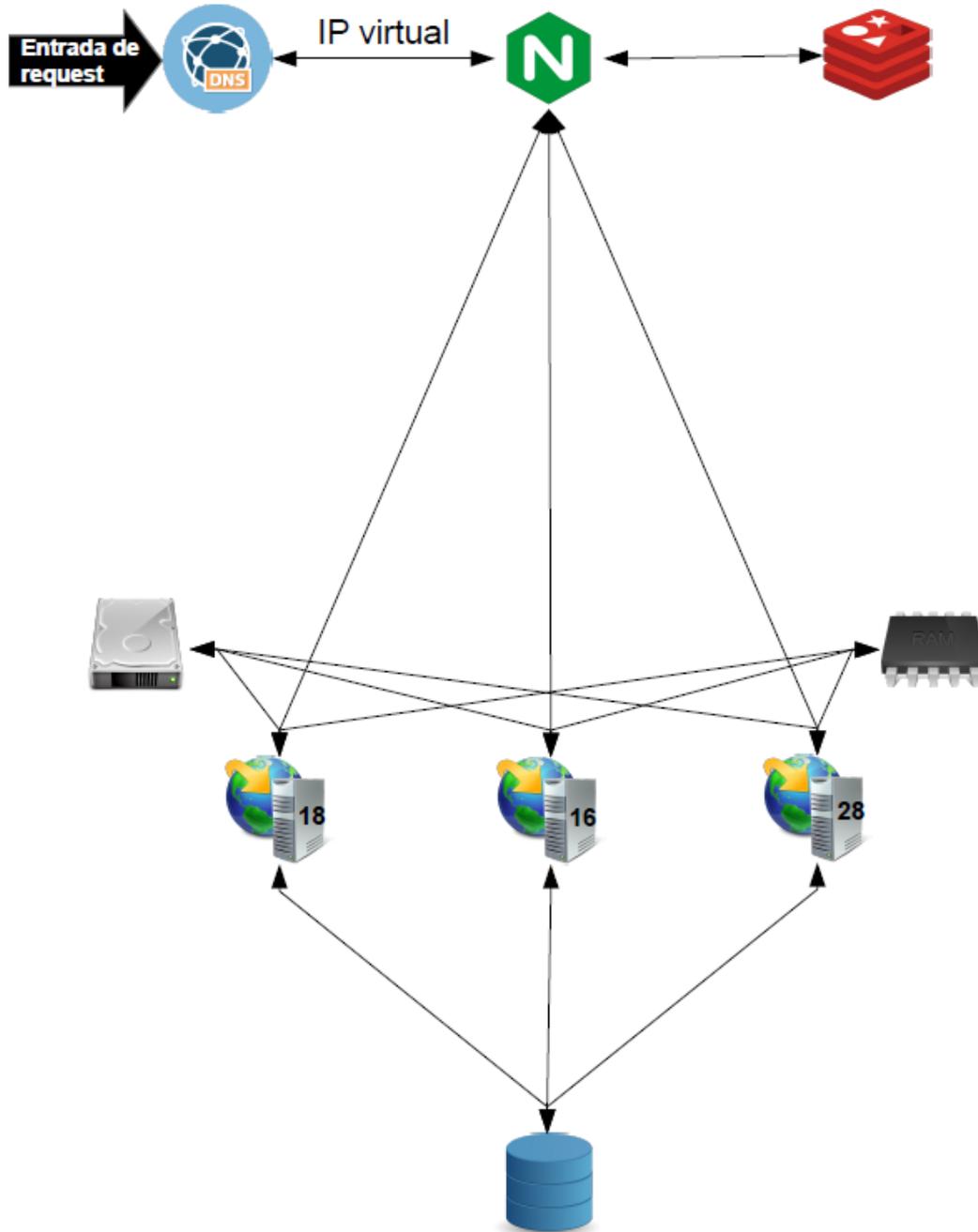
A la hora de conseguir una buena experiencia de usuario, es necesario tener implementada la mejor arquitectura de servidores posible. En Europa Press, existe una arquitectura de servidores que busca tanto la mayor disponibilidad como la mayor velocidad. Para ello se busca evitar que a la hora de conectarse a la web se necesite llegar hasta la base de datos, donde se localizan los recursos para conseguir el contenido, mediante pasos intermedios donde diversos mecanismos van a trabajar como cachés. Además, la alta disponibilidad se consigue mediante la implementación de un “*high availability cluster*” compuesto de tres servidores cuya labor principal es la de recibir distinto tipo de tráfico específico, pero

*DESCRIPCIÓN DE LAS TECNOLOGÍAS*

---

también actuar de respaldo a la hora de las posibles caídas de los demás. Estas caídas son bastante frecuentes ya que a la hora de actualizar los contenidos del código fuente de la web, se necesita desconectar y vaciar de tráfico dentro de cada servidor para actualizar la versión con la que está trabajando cada uno. Esta práctica se realiza de manera diaria, por lo que es crítico que el volcado de tráfico se realice de manera fluida sin que esto pueda ser observado por cualquier usuario que esté navegando por la página web a la vez que se está publicando la nueva web.

La arquitectura que sostiene este sistema se corresponde con el diagrama que se presenta en la Figura 6:



*Figura 6. Esquema de arquitectura de servidores*

Para entender este diagrama es necesario hacer una leyenda que explique cada elemento y cuál es su función dentro de este sistema, esta se incluye a continuación en la Tabla 1; **Error! No se encuentra el origen de la referencia.** :

*Leyenda de elementos de la arquitectura de servidores de Europa Press*

<i>Icono</i>	<i>Descripción</i>
	<b><u>Servidor DNS</u></b> Servidor que resuelve las direcciones pertenecientes a Europa Press por una IP virtual
	<b><u>Cluster de Nginx</u></b> <i>Cluster</i> de dos proxys inversos de Nginx. Uno de ellos de respaldo, no escucha peticiones, sino que está escuchando el estado del otro proxy
	<b><u>Redis</u></b> Mecanismo de caché rápida, el cual guarda las noticias según las visitas que se realicen a estas. Se busca que las noticias más actuales y con más popularidad estén disponibles en esta caché, al ser el primer elemento que se encuentra en la arquitectura del sistema. En el caso de reflote de una noticia antigua, se intenta que esta se quede en caché un menor tiempo que una noticia de igual relevancia, pero con un fechor más actual, esto se consigue asignando un TTL mayor para aquellas que tienen un fechor más actual

	<p style="text-align: center;"><b><u>Caché en disco</u></b></p> <p>Caché en disco estándar que guarda las noticias en producción más recientes para actuar de caché</p>
	<p style="text-align: center;"><b><u>Caché en RAM</u></b></p> <p>RAM que contiene la última hora de noticias para devolverlas rápidamente</p>
	<p style="text-align: center;"><b><u>Base de datos</u></b></p> <p>Base de datos donde se encuentra la totalidad de los contenidos de la Web. Se encuentra al final de la estructura de acceso de peticiones porque se quiere limitar su uso.</p>
	<p style="text-align: center;"><b><u>IIS</u></b></p> <p>El sistema se compone de tres servidores. El número 18 que procesa las noticias actuales, el 16 que trabaja sobre las noticias antiguas y el 28 sobre las sesiones de abonados. Entre ellos forman un <i>cluster</i> para que por ejemplo en caso de caída del número 18, se ejecute un balanceo de la carga de peticiones al número 16, su servidor de respaldo.</p>

Tabla 1. Leyenda de la figura 4

*DESCRIPCIÓN DE LAS TECNOLOGÍAS*

---

Una vez conocida la arquitectura de los servidores y teniendo en cuenta todos los elementos del sistema y cuáles son sus roles, es importante tener en cuenta cómo se desarrolla el comportamiento de este sistema a la hora de recibir las peticiones generadas por los visitantes de la página web. Esto se detalla en el siguiente diagrama de flujos, Figura 7:

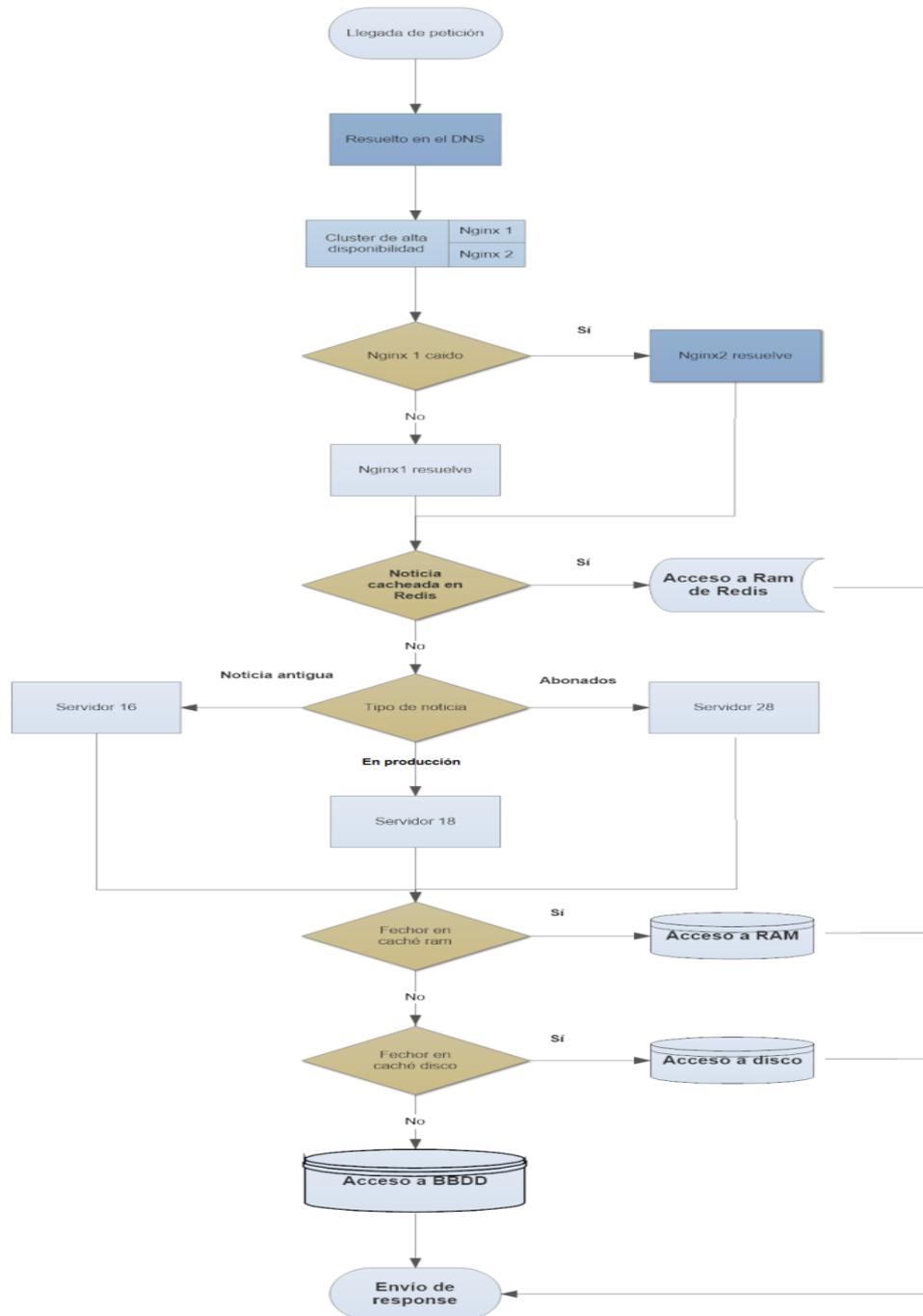
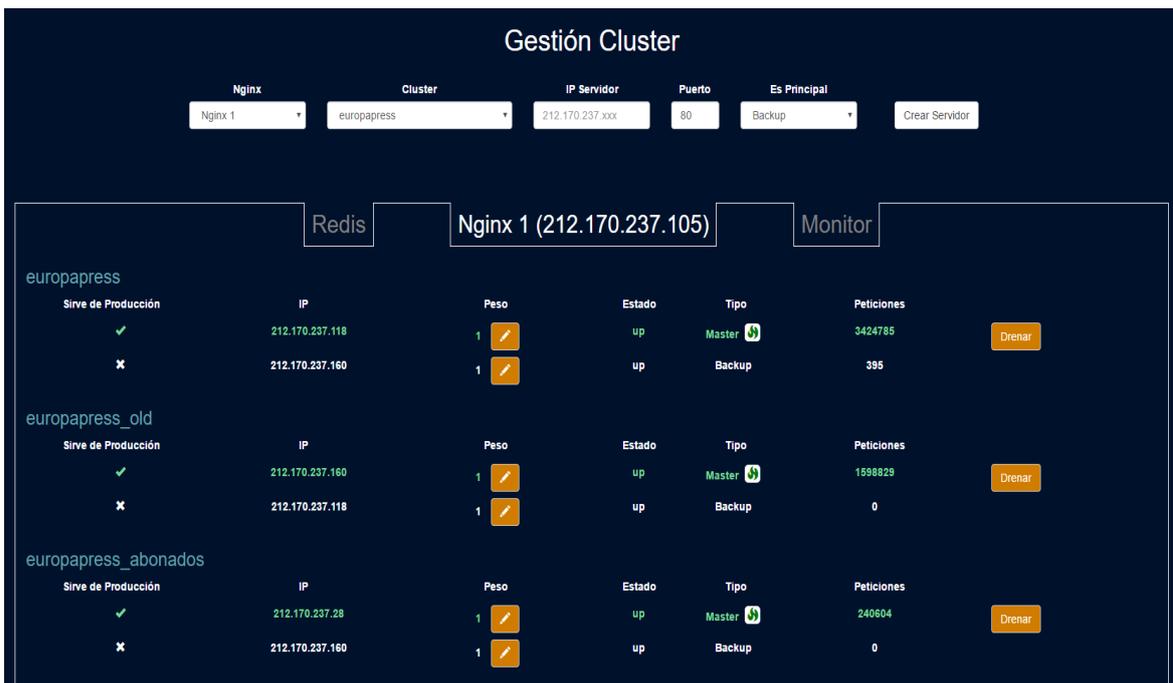


Figura 7. Diagrama de flujo de tratamiento de requests

Gracias a este diagrama de flujo, se confirma como este sistema está optimizado de manera que el acceso a la base de datos de todas las noticias sea el último recurso necesario a la hora de devolver la información a nuestros visitantes. Se implementan dos capas de cacheo por

RAM y otro de acceso a disco, de manera que los tiempos de carga de la información relativa a las noticias sea el menor posible y a la vez limitar lo máximo el acceso a base de datos.

Finalmente, cabe reseñar como este sistema de alta disponibilidad esta automatizado con un sistema de gestión remota del *cluster*, ver Figura 8.



*Figura 8. Plataforma de gestión de cluster de servidores*

En esta figura, podemos observar como para cada tipo de noticias, se tiene un *cluster* compuesto de un servidor principal y otro de respaldo como se explicó en la Tabla 1.

Esta plataforma es crítica para resolver dos problemáticas que se presentan en este entorno de trabajo. De manera general, este sistema de gestión nos permite saber en todo momento la salud de los servidores, por lo que, en el caso de detectar anomalías en el funcionamiento de la web, como por ejemplo un rendimiento de carga del web muy lento, problema crítico para este TFG, saber si esto se debe los servidores, como por ejemplo una caída de uno de ellos que repercute en la saturación del servidor que actúa de respaldo de este. Aprovechando la modalidad de respaldos entre servidores, podemos hacer una segura actualización del código de la Web. Es dentro de estos servidores que se encuentra la versión en producción

de la Web, por lo que, a la hora de hacer una fusión entre la versión en desarrollo, la versión que comparten de manera local los desarrolladores de la web, y la versión de producción, es necesario realizar un drenado de cada servidor en el momento de realizar esta actualización, lo que se gestiona de forma automática con el uso de este sistema. Con esto nos ahorramos fallos inesperados y tener que realizar cambios sobre una plataforma que está repartiendo contenido a nuestros visitantes, lo que podría llevar a fallos en los servidores en cuanto al uso de recursos, por ejemplo.

## ***2.4 HERRAMIENTAS DE ANÁLISIS***

Al estar haciendo un proyecto enfocado a la velocidad de rendimiento es necesario el uso de distintas herramientas que permitan hacer un estudio de esta métrica a lo largo de todo este desarrollo. De esta manera nos podemos guiar de manera objetiva a medida que se realizan las diferentes modificaciones.

Las herramientas principales que sirven de guía de este proyecto son:

### **2.4.1 WEBPAGETEST**



Analizador de velocidad de carga de páginas web [7] que nos permite un análisis detallado de los fenómenos que ocurren a la hora de conectarse a una web.

Nos permite conocer los detalles del tiempo de carga de cada elemento, saber cuánto y qué tipo de contenido se usa en cada web y analizar cada request que entre en juego en la carga. También aporta gran versatilidad, pudiendo elegir entre distintas localizaciones desde donde

se va a llevar cada test, lo que repercute sobre el RTT estimable, y nos da a elegir entre distintos dispositivos y navegadores.

### **2.4.2 GOOGLE ANALYTICS**



Google Analytics

Herramienta de Google que permite una monitorización del tráfico que reciben las páginas web que deseamos. Esta herramienta nos permite por medio de introducir scripts dentro del código fuente de la web, conocer al detalle los parámetros de visitas por páginas, tiempo de sesión, estos dos siendo importantes para nuestro TFG, junto con una gran variedad de parámetros que pueden proveer gracias a la potencia de la plataforma Google.

### **2.4.3 CHROME DEVTOOLS**



Herramienta incluida en el navegador de Chrome que nos otorga una gama de funcionalidades para analizar páginas a las que nos conectamos. Estas herramientas nos permiten analizar los componentes del DOM y los estilos, poder depurar y ejecutar código JavaScript y hacer distintos análisis de carga de las páginas webs.

#### **2.4.4 BROWSERSTACK**



Herramienta por la cual podemos ejecutar cualquier tipo de dispositivo móvil físico de forma remota, lo que sirve para hacer pruebas en tiempo real desde terminales reales externos. Podemos entonces obtener resultados totalmente independientes de nuestro entorno de desarrollo, en vez de simular o ejecutar los cambios desde equipos locales, como los que se encontraría cualquier persona que se conecte desde su propio dispositivo personal, de esta manera podemos ver el mapa de los resultados según los dispositivos disponibles en el mercado

## Capítulo 3. ESTADO DE LA CUESTIÓN

### 3.1 LIMITACIONES HTTP/1

A día de hoy, gracias a la liberalidad del mercado, tenemos gran cantidad de ofertas de distintos proveedores de servicios de Internet. Las velocidades ofertadas son cada vez más grandes y siguen subiendo a un buen ritmo [8]. Pero a la hora de carga de una página web, no solo nos tenemos que interesar a tener un buen ancho de banda que nos permita cargar una gran cantidad de datos, puesto que existe otro parámetro muy importante en cuanto a rendimiento de carga de página, este es la latencia. Si vemos los estudios realizados por Mike Belshe [9], podemos observar como el impacto de la latencia sobre la velocidad de carga es casi lineal, a cuanto menor latencia menos se tarda en descargar la página, mientras que por mucho que aumentemos el ancho de banda, parece que por encima de niveles de 5 Mbps de ancho de banda no se obtiene una mejora significativa sobre cuánto tarda un web en cargar.

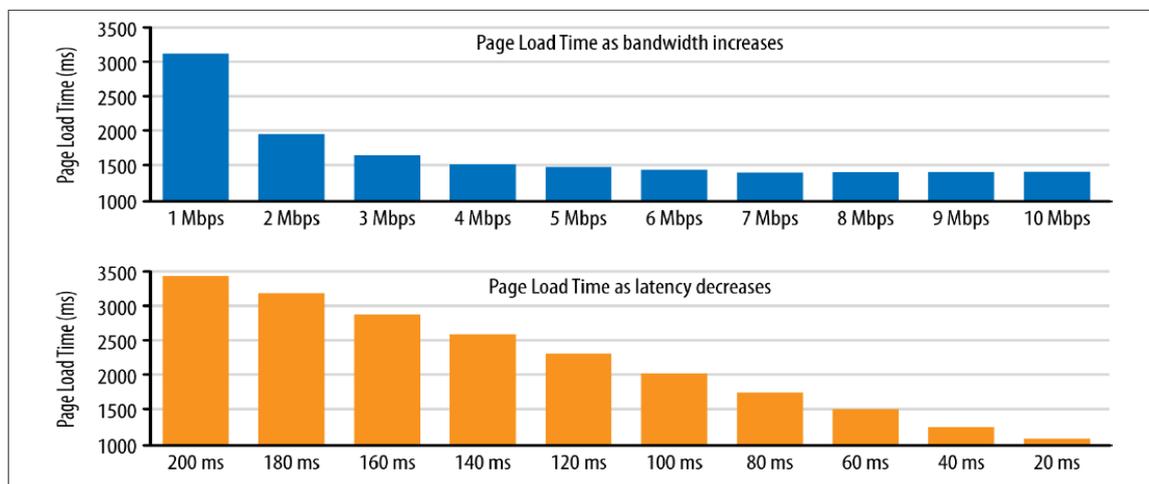


Figura 9. Tiempo de carga en relación con la latencia y el ancho de banda [10]

Es por esto que se debe dar importancia al factor de RTT (Tiempo de ida y vuelta) a la hora de conseguir buenos tiempos de carga a la hora de navegar por Internet.

HTTP/1 es un protocolo dirigido a conexión. A la hora de conectarse a una página web, HTTP/1 incurre en producir múltiples conexiones al origen para cargar de manera paralela diversos recursos de la página que se quiere cargar. Esto podría parecer muy beneficioso, pero este tipo de paralelismo está limitado. Usualmente se realizan 6 conexiones en paralelo por origen, como se puede ver en la figura siguiente:

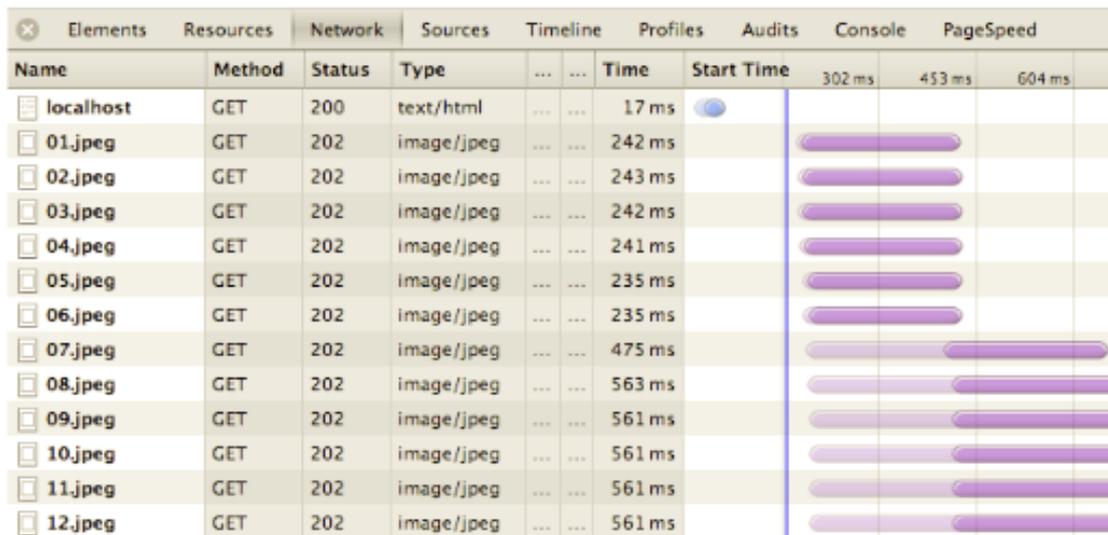
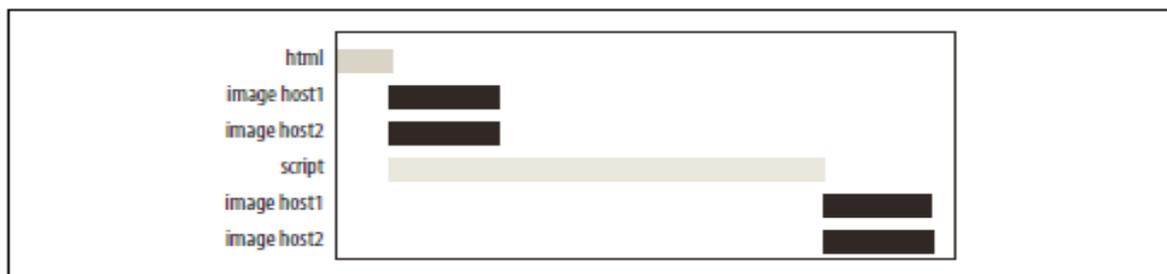


Figura 10. Ejemplo de paralelismo en HTTP/1

Parece todo muy positivo en cuanto a lo que podríamos tardar en cargar cada recurso haciendo miles de conexiones paralelas, pero esta práctica no sería útil y no se realiza por distintos motivos. A la hora de realizar este paralelismo, cada una de estas conexiones incurre en un *handshake* de TCP completo. HTTP/1 es un protocolo basado en TCP, por lo que cuando establece una conexión necesita ejecutar el mecanismo de establecimiento de conexión entre cliente y servidor. Esto significa tres etapas, el conocido *three way handshake* de TCP, en las que añadimos tiempos de RTT, primero el cliente debe enviar un SYN al servidor, el cual se debe confirmar mediante un ACK y finalmente el cliente debe confirmar la recepción de este ACK enviándole uno propio, al final de estos pasos la conexión ya estaría establecida. Una vez hecho esto ya se puede empezar a realizar intercambios de datos de aplicación. Entonces, aunque estemos realizando conexiones en paralelo, estas están aumentando la latencia de la conexión. Si estamos trabajando con una conexión segura, habrá que añadir la etapa de negociación de conexión de TLS. Además, al estar realizando

tareas en paralelo estamos gastando más recursos en los proxys y servidores, lo que no es ideal. No solamente estamos saturando estos elementos, sino que estamos complicando el control de congestión ya que cada una de estas conexiones está compitiendo con las demás que se estén estableciendo de manera simultánea. Una técnica que se utiliza para mejorar esto es el *domain sharding*, que consiste en dividir en varios hosts los dominios donde se alojan los datos de la web, pero no es más que un truco que complica la estructura de la web y en el caso de llegar a ser abusado puede terminar no siendo tan eficaz.

HTTP/1 no tiene únicamente problemas en la optimización en cuanto a minimizar la latencia que introduce al realizar las conexiones, sino que además tiene algunos otros problemas que empeoran la velocidad de carga de las webs. Uno de ellos es el *head-of-line blocking* (HOL blocking). A la hora de la carga de la página, los recursos se van cargando de forma secuencial, y un problema que puede aparecer es un bloqueo indefinido, al estar el cliente esperando un recurso preciso:



*Figura 11. Ejemplo de HOL Blocking por retardo de descarga de script [10]*

En la Figura 11 vemos como por un problema en la entrega de un script se genera una espera muy larga que ralentiza de manera muy significativa la carga de la web. No podemos entregar el resto de datos ya que la manera de trabajar de este protocolo requiere proveer los datos solicitados por el cliente siguiendo el orden de elementos, no podemos cargar el resto de elementos hasta que se haya terminado la carga de lo solicitado, por lo que estamos expuestos a posibles errores de transmisión que pueden provocar bloqueos indefinidos.

No solamente estamos desprotegidos ante los bloqueos indefinidos, sino que la estructura de los mensajes intercambiados no está preparada para ser comprimida de manera óptima.

Aunque en la actualidad se use el algoritmo GZIP sobre HTTP/1, este algoritmo solo se aplica sobre los bloques de datos de los mensajes. Pero hay una parte del mensaje que no estamos reduciendo, y este es la cabecera. La estructura de los mensajes de HTTP/1 no nos permiten comprimir cabeceras por lo que estamos enviando muchos mensajes que comparten cabeceras y no somos capaces de reducir estos datos redundantes. Se pueden dar incluso casos en los que los datos enviados son tan pequeños que tienen un tamaño parecido a las cabeceras del mensaje que necesitan crear para ser enviados. Podemos entonces afirmar según lo que acabamos de describir como HTTP/1 no está optimizado para conseguir la mejor latencia posible.

### **3.2 HTTP/2**

HTTP/1 tiene diversas versiones que se han desarrollado a lo largo de su vida útil, pero no ha sido hasta hace poco que se ha decidido cambiar este denominativo a HTTP/2. Esto se debe a un cambio drástico en la manera de trabajar de este protocolo. Como hemos dicho anteriormente, el estándar anterior trabajaba sobre varias conexiones TCP, pero con la llegada de esta nueva versión se ha decidido que solamente se realice una conexión en toda la carga de la página. Para ilustrar la relevancia de este cambio vamos a citar a Patrick McManus, *Principal Enginner* en Mozilla, que dijo: *“One great metric around that which I enjoy is the fraction of connections created that carry just a single HTTP transaction (and thus make that transaction bear all the overhead). For HTTP/1 74% of our active connections carry just a single transaction - persistent connections just aren't as helpful as we all want. But in HTTP/2 that number plummets to 25%. That's a huge win for overhead reduction”*. Esto sirve de referencia para observar como con este cambio en la manera de trabajar se está reduciendo lo que explicamos anteriormente en el apartado sobre HTTP/1 y ya no estamos utilizando tantas cabeceras de forma ineficaz. Pero para que pueda funcionar bajo esta única conexión se ha cambiado la estructura de los mensajes que se intercambian. En HTTP/1, tanto la parte de datos como la cabecera estaban unidas en una misma estructura que se pasaba a través del socket de la conexión, mientras que, bajo el nuevo formato, se

separa esta unidad en dos *frames* más reducidos, que se distinguen entre *frames* de datos y *frames* de cabeceras.

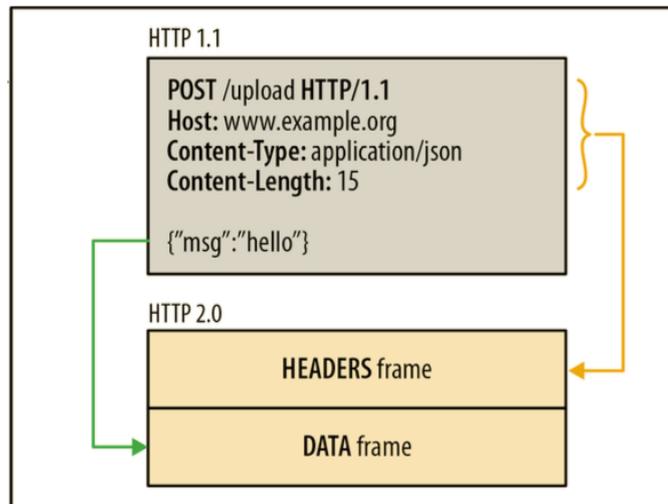


Figura 12. Estructuras de datos de Http/1 y Http/2[10]

Estos se envían de forma separada y mediante un identificador pueden ser reagrupados cuando lleguen a su destino. Si lo ponemos en relación con lo explicado en el apartado anterior, parece que esto puede tener un efecto positivo sobre la velocidad de carga puesto estamos mejorando nuestra latencia. No será necesario hacer muchos establecimientos de conexiones paralelas, los cuales introducían un tiempo mayor de latencia por añadir pasos de RTT en las conexiones. Pero no solo estamos ahorrando latencia innecesaria, sino que depurando errores anteriores.

Anteriormente, vimos como el HOL *blocking* podía producir bloqueos indeterminados que suponían una ralentización de la carga de las páginas de manera muy significativa, con este nuevo sistema de intercambio de mensajes este problema está resuelto. Antes los intercambios de los mensajes seguían un orden y hasta que el recurso que pidiera el cliente no llegase, se bloqueaba la carga del resto de recursos. Con esta nueva versión, lo que se está haciendo es transmitir sobre una única conexión distintos recursos a la vez. Los *frames* de datos y de cabeceras se envían de manera no secuencial, por lo que puedo enviar a la vez un

*frame* de datos de un archivo CSS seguido de otro de un script, no hace falta tener uno para empezar a transmitir el otro.

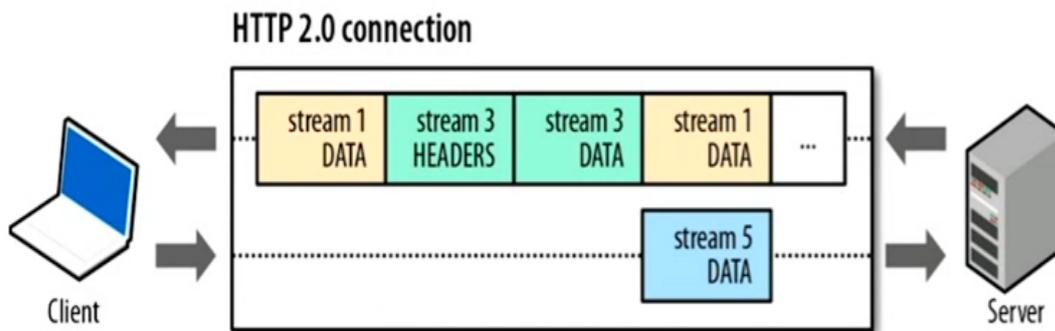


Figura 13. Ejemplo de comunicación HTTP/2[10]

Ahora podemos realizar de manera no secuencial las diversas *request* y *responses*, se pierde esa secuencialidad por una estructura en la cual los datos vienen intercalados y luego se recomponen, sin tener que incurrir en bloqueos. En cuanto a las cabeceras, al ir ahora separadas de los datos de mensaje, HTTP/2 introduce una manera de aplicar compresión sobre estas cabeceras. En vez de enviar el mismo dato en cada *request* o *response*, HTTP/2 usa una tabla de cabeceras tanto en el cliente como en el servidor para guardar y controlar los pares de valor de claves. Estas tablas persisten a lo largo de la vida de esta única conexión y se van incrementando a lo largo del periodo de intercambio de mensajes. Con esta tabla se podrá comprimir las cabeceras usando estos valores, mediante la comparación con la cabecera que se esté compartiendo.

### HPACK header compression

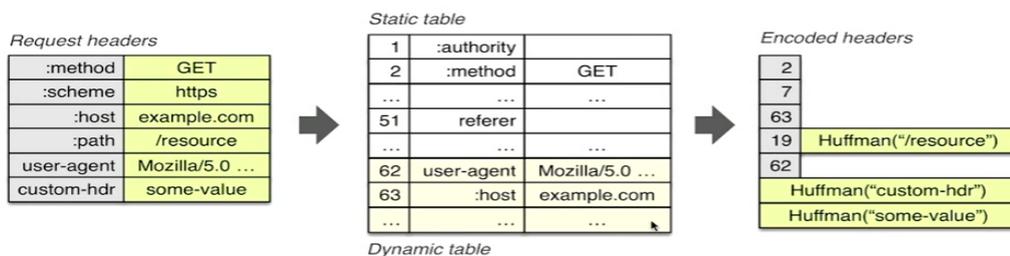


Figura 14. Compresión de cabeceras mediante HPACK en HTTP/2 [10]

A parte de solucionar los problemas de HTTP/1, se introducen nuevas mejoras que resultan de interés para el rendimiento web. Al estar trabajando con una única conexión de TCP, se está simplificando el control de flujo, se puede además implementar un sistema donde se introduce la prioridad de los envíos, para que los *frames* que determinamos que son críticos, sean cargados antes que el resto. A nivel de servidor, HTTP/2 permite realizar el llamado *server push*, mediante este mecanismo el servidor es capaz de proveer archivos al cliente de manera autónoma. No necesita que el cliente le haga la petición, lo que puede reducir el número de etapas de una comunicación. Esto puede ser útil cuando hay dos elementos muy relacionados entre sí y la correlación entre que el cliente pida uno seguido luego del otro es muy elevada, por ejemplo, si el cliente pide un archivo JavaScript de una web, el servidor mediante *push* le devuelve el JS acompañado del código CSS de esa web, dando por hecho que existe una certeza de que cuando se pida ese JS se haga la petición de ese CSS a continuación.

Aún con todos los beneficios de esta nueva versión del protocolo de intercambio de hipertexto, es necesario que todas las conexiones se realicen sobre sockets seguros. Por lo que, si la web no es compatible con HTTPS, no es posible implementar HTTP/2. Esto puede no ser necesario para un cierto tipo de web, en cuanto no se considere necesario crear canales seguros para ese tipo de contenido. Una migración desde un sistema con sockets estándar a una que corre con todas sus conexiones encriptadas para tener HTTPS supone un proceso muy costoso, que puede llegar a no valer la pena si la web no determina crítico una velocidad de carga optimizada por HTTP/2 o tener sockets encriptados. Para ilustrar la dificultad de esta transición, podemos citar al NY Times, el cual tardó dos años en hacer la transición a HTTPS para a posteriori implementar el protocolo HTTP/2 en su web.

### **3.3 AMP**

La cuota de mercado de los dispositivos móviles en el sector de Internet se está convirtiendo de manera progresiva más importante. Progresivamente la gente consume más datos de Internet cuando y donde quieren gracias a la realidad, ya más que suficientemente extendida, de los smartphones. Cisco prevé que la tasa anual compuesta de crecimiento de datos móviles

entre 2016 y 2021 será del 47%, suponiendo la cantidad de los datos provenientes de smartphones y dispositivos similares el 66% de la cuota de este mercado [11].

El problema viene entonces en que en muchos casos estos usuarios tienen acceso a redes inferiores a los 4G, que no aseguran la carga efectiva de las páginas, pero el usuario sigue esperando una experiencia parecida a la que tendría en un ordenador de sobremesa. Igualmente, la tecnología 4G no garantiza tener una experiencia de carga de página rápida, ya que no todas las páginas tienen una versión optimizada para las conexiones móviles. Es por esto que Google ha creado un *framework* sobre HTTP llamado *Accelerated Mobile Pages* (AMP).

Google propone un nuevo estándar que busca la simplificación de todo lo que existe en HTTP, con el objetivo de poner el rendimiento como prioridad en el diseño web. Una de las características más importante para ellos es la de que la página no debe, bajo ningún concepto, parar de cargar en ningún momento. Para ello, AMP rechaza la carga síncrona de códigos Javascript, por lo que si se quiere usar algún tipo de tecnología Javascript se deberá hacer mediante carga asíncrona o implementarlo en un *iframe* compatible con AMP. También presenta un enfoque distinto al ahora conocido con las hojas de estilo de CSS, propone que todo ese código esté implementado en el propio archivo HTML, al inicio de este. Esta práctica es conocida como CSS *inlining*, pero para esto pone un límite, todo ese código no podrá exceder los 500kB de tamaño, ya que de superar este tamaño se pierden los beneficios de velocidad y se considera un tamaño suficiente para los estilos de una página, pero no para una mezcla de diversos formatos. Combinando estas dos prácticas estamos evitando que se produzcan peticiones HTTP sin que no estén cargadas las fuentes de texto, lo que de no ser así crearía el conocido como *Flash of unstyled text*, lo que viene a ser que salgan las letras de un formato y luego cambian dando un efecto de flash que da una experiencia de usuario muy negativa. Además, se quiere evitar que, durante la carga, la página experimente cambios en los tamaños de los elementos, por lo que se pide indicar el tamaño exacto de cada elemento para que a la hora de renderizar, antes de cargar el contenido, se deje un *placeholder* de tamaño equivalente, y no se muevan de repente los elementos, mientras el usuario está consultando la página, una vez estén cargados.

Si lo descrito anteriormente ayuda a que las páginas se carguen de manera más fluida, existen tres características proporcionadas por Google que son determinantes para convencer a los posibles usuarios que creen sus propias páginas AMP. Estos son el *prerendering*, el Google AMP caché [12] y el carrusel de noticias. El *prerendering* consiste en la capacidad de las páginas de AMP de cargar en segundo plano contenido al que el usuario va a acceder potencialmente a continuación de entrar en una página. Lo que hace este método es, al terminar de cargar una página, cargar en segundo plano otra muy susceptible de ser visitada, descargando solo los elementos del primer punto de vista de ese contenido y evitando la carga de JS de terceras partes que se puedan contener en algún *iframe*. Esto da una sensación de velocidad al usuario ya que, si accede a este contenido, este ya estará cargado y no percibirá la carga del resto de la página que no está cargada todavía. El AMP caché se combina con el carrusel de noticias, este servicio consiste en que son los propios servidores de Google los que cachean las webs AMP que consideren de destacada importancia y las muestran en un carrusel en su propia página, por ejemplo si buscamos la palabra deportes en el motor de búsqueda de Google desde un smartphone, se nos sugieren unas noticias relacionadas con esa búsqueda en una sección con el nombre de noticias destacadas, que en el caso de elegir una de ella, abre una galería de noticias con formato AMP, que se encuentran hospedadas por la misma Google.

Aquí no solamente estamos hablando de obtener un buen rendimiento gracias al cacheo que ejerce Google sobre esas páginas AMP, sino que al aparecer en ese carrusel se otorga una incomparable exposición a esas páginas, muy valorada en términos de SEO, lo que en caso de no ser aprovechado por parte de las páginas, se estarían perdiendo una gran oportunidad de negocio.



## Capítulo 4. DEFINICIÓN DEL TRABAJO

### 4.1 JUSTIFICACIÓN

En vista de los contenidos presentados en el Capítulo 3. , podemos afirmar como en el sector de las páginas web, el estándar HTTP/1 está entrando en fase de declive. La importancia de dar la mejor experiencia de usuario posible, en cuanto a velocidad de rendimiento se refiere, es crucial para conseguir buenos indicadores de vistas e implicación por parte de los visitantes, ya que se traduce en un mayor índice de conversión por anuncios. Los defectos de este protocolo ya conocen una solución, que se ha implementado en la nueva versión HTTP/2. Esta versión presenta cambios de funcionalidad, que otorgan resultados superiores sobre la anterior y debe ser tenida en cuenta por los desarrolladores. Pero no solo debemos centrarnos en este estándar ya que, el mercado de los smartphones es cada día más y más significativo. Para este sector existe un estándar introducido por Google, llamado AMP. Este *framework* optimiza el rendimiento de navegación web desde estos dispositivos y provee oportunidades de negocio que viniendo desde un gigante de internet no deben de ser pasados por alto. Estos estándares deben ser muy tenidos en cuenta por aquellas empresas que quieren conseguir una ventaja estratégica sobre las demás, dando a sus clientes experiencias de velocidad rápidas.

Todo esto en relación a lo presentado sobre el caso empresarial en el que se engloba este TFG, explicado en el Capítulo 1. , sirve de justificación a lo que se va a desarrollar en este proyecto.

## **4.2 OBJETIVOS**

Los objetivos dentro de este TFG son:

- Diseño de una solución a la velocidad de la web

Este diseño vendrá precedido de un informe que haga un diagnóstico sobre qué es lo que entra en juego a la hora de conectarse a la web de Europa Press. Una vez hecho esto, se podrán considerar cuales son los principales aspectos mejorables en el diseño de esta web y proponer unas posibles medidas a implementar.

- Implementación cambios que supongan una notable mejora de la velocidad

A partir de la propuesta de mejoras de la página web, habrá que desarrollar todo lo que sea posible modificar dentro de la actual estructura de la web, para conseguir mejoras que sean lo suficiente buenas para apreciarse a nivel del ojo humano.

Estudio de resultados

A partir de los cambios que se hayan podido efectuar, habrá que realizar de nuevo un diagnóstico para ver cómo han afectado estas mejoras a la web y decidir si mantener los cambios realizados. Aquí entrará en juego ver en qué medida se han podido satisfacer los resultados que se esperaban obtener. Como ya se ha explicado anteriormente, uno de los motivos por los que tan importante este estudio de optimización dentro de la página es que la mejoría que se haya podido obtener repercuta en la experiencia de usuario y pueda suponer beneficios en algún otro aspecto de web, como por ejemplo en términos de SEO.

- Realizar un plan de mejoras futuras

Al estar trabajando dentro de un plazo de tiempo tan pequeño, van a ser difíciles realizar los cambios más ambiciosos dentro de la web, es por eso que después de realizar todas estas modificaciones será necesario volver a hacer un diagnóstico de la web. Será acertado desarrollar una nueva propuesta que sugiera unas mejoras futuras a partir de la valoración

del proyecto sobre los cambios pendientes, las cosas que no han funcionado dentro de las mejoras implementadas y recogiendo nuevas mejoras posibles que se puedan haberse considerado al final del proyecto.

### **4.3 METODOLOGÍA**

La realización de este proyecto se puede dividir en 6 partes principales que son las siguientes, en orden cronológico:

- Estudio teórico:

Este estudio va a suponer el establecimiento de las bases principales del proyecto. Es la tarea inicial ya que es durante esta que se puede empezar a esbozar todos los componentes que van a entrar en juego a la hora de realizar la optimización de la web. Se va a hacer un estudio del mercado por el cual, queremos saber qué es lo que se exige en la actualidad, en el ámbito de las páginas web de noticias. Además, me voy a interesar en estudiar los comportamientos de los usuarios para poder definir unos objetivos que se centren en conseguir que todo lo que se busca optimizar tenga un efecto apreciable a nivel de la experiencia de usuario, ya que únicamente sabiendo que es lo que realmente busca el usuario podemos deducir que se debe mejorar.

El estudio del estado del arte forma parte imprescindible de esta etapa, puesto que se necesita saber qué es lo que hay disponible actualmente, para que, a la hora de analizar el estado de la web, se puedan detectar fácilmente cuales son las carencias principales de la página. Por lo que el siguiente paso debe ser el de conocer a fondo como funciona y de que se compone la estructura web de Europa Press.

- Análisis de rendimiento

Nuestro objetivo principal en este proyecto es el de que los esfuerzos se traduzcan en mejoras de rendimiento. Por ello, es esencial monitorizar cual es el rendimiento anterior a que se ejecuten los cambios que se propongan durante este diseño. Mediante este estudio, no

solamente estamos dejando preparado el terreno para realizar el estudio de resultados final, sino que durante este análisis se podrá observar problemas existentes en el rendimiento. Para realizar esto de manera más eficaz, se va a recurrir al análisis de webs de la competencia para aprender de sus errores y detectar las buenas prácticas propias. Esto nos ayuda a poner en contexto el análisis de nuestra web y detectar que se está haciendo bien y en que se está fallando.

- Diseño de plan de optimización

A partir de todo lo recopilado en los pasos previos, se debe realizar una propuesta de cambios justificados, para buscar la mejora de la web. Toda alternativa debe ser considerada y deliberar un juicio crítico que lleve a la decisión de llevar a cabo, o no, las distintas opciones que puedan entrar en juego.

- Desarrollo de cambios

En esta fase se busca implantar en la estructura de la web de Europa Press todos los cambios que hayan sido considerados como viables y validados por la empresa.

- Análisis de cambios

Esta etapa sirve de evaluación sobre los cambios implementados y de reevaluación del plan de optimización. Se busca analizar qué es lo que ha aportado realmente cada cambio y ver si su implementación estaba realmente justificada. Además, se quiere comparar el estado de la página con respecto a la situación de partida y así dilucidar si ha habido otro tipo de impacto, como pueda ser a nivel SEO, por ejemplo.

- Desarrollo de un plan de mejoras futuras

Una vez todo el estudio ejecutado, será necesario hacer un análisis crítico sobre todo lo realizado. Esto ayudará a elaborar un plan, que aprovechando de los conocimientos adquiridos y las conclusiones a las que se haya llegado, proponga una continuación a este

proyecto, describiendo cuales son las posibles mejoras futuras que puedan suponer una continuación al trabajo realizado durante estos meses de proyecto.

## **4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA**

### **4.4.1 PLANIFICACIÓN INICIAL**

Este proyecto se ha realizado a lo largo de un periodo de 6 meses de trabajo en la empresa Europa Press, precedidos de 4 meses de búsqueda individual por parte del alumno para investigar sobre las posibles opciones y las distintas tecnologías que pudieran ser interesantes en la consecución de este trabajo.

A principios del segundo cuatrimestre del año lectivo 2016-2017 se han facilitados los plazos estimados de entrega de TFG para este curso que correspondían con:

- Fechas de la primera convocatoria:
  - Entrega de la memoria : 14 de junio 2017
  - Defensa : 17 de junio 2017
- Fechas de la segunda convocatoria:
  - Entrega de la memoria : 18 de julio 2017
  - Defensa : 20 de julio 2017

Teniendo en cuenta los plazos proporcionados por la escuela, se ha determinado un plan de distribución del trabajo a lo largo de la duración que se estima para un proyecto de estas características. Este esquema de reparto de tareas representa lo estipulado en las fases previas del proyecto, definiendo así la línea base de este y pudiendo no haber sido respetado literalmente a lo largo de su desarrollo

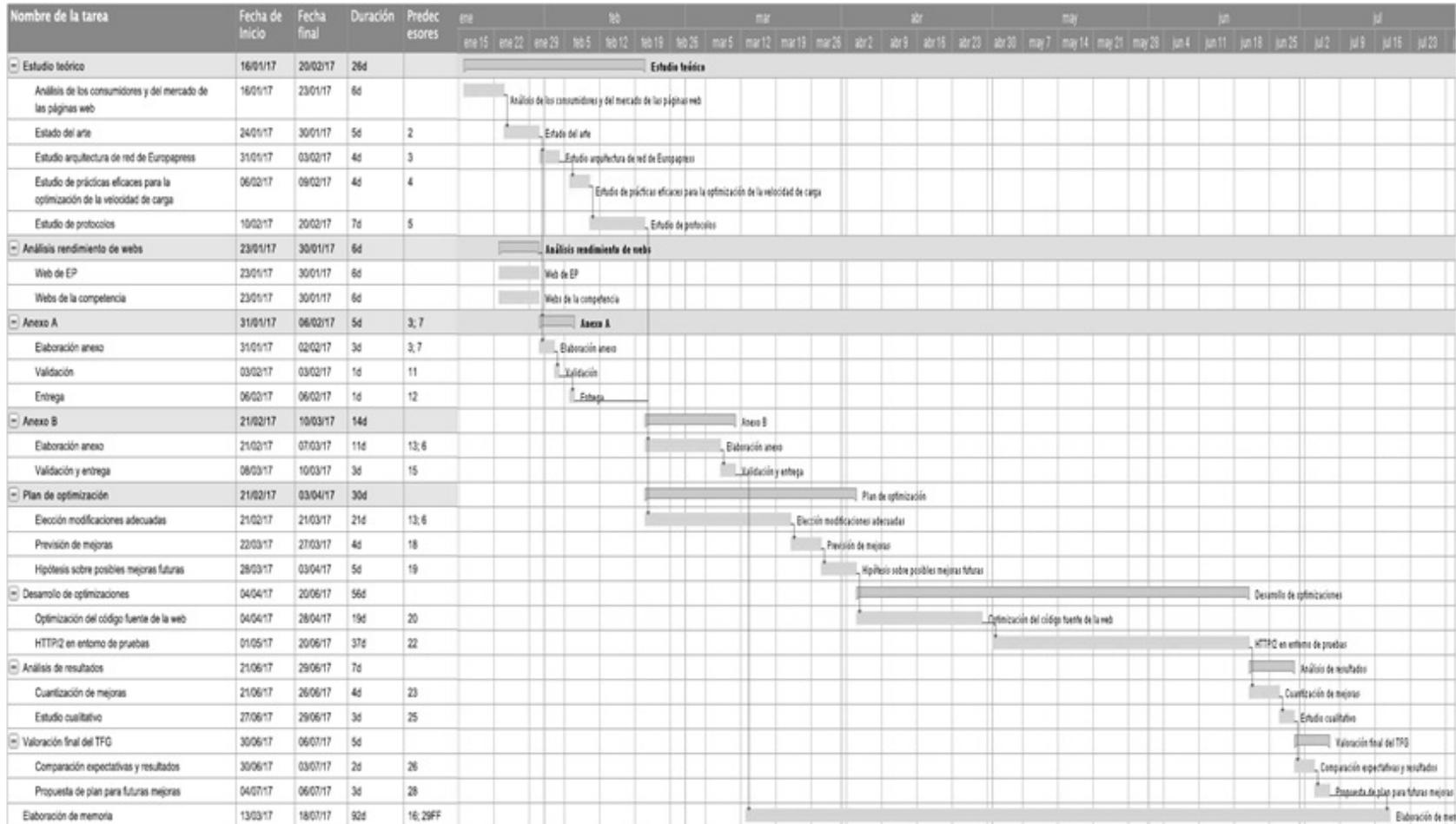


Figura 15. Planificación inicial del proyecto

#### 4.4.2 ESTIMACIÓN ECONÓMICA

Este proyecto va a suponer un gasto económico, que será realizado por la empresa Europa Press, durante la realización de las prácticas en entorno profesional durante los meses de enero de 2017 a julio de ese mismo año.

Los gastos que supone este proyecto en materia de personal se desglosan en la Tabla 2 como se muestra a continuación:

<b>Perfil</b>	<b>Horas</b>	<b>Sueldo/hora</b>	<b>Total</b>
<b>Programador en prácticas</b>	<b>300 h</b>	<b>20 €/h</b>	<b>6 000 €</b>
<b>Director de proyecto</b>	<b>20 h</b>	<b>60 €/h</b>	<b>1 200 €</b>
<b>Coordinador de proyectos</b>	<b>10 h</b>	<b>60 €/h</b>	<b>600 €</b>
<b>TOTAL</b>	<b>430 h</b>		<b>7 800 €</b>

*Tabla 2. Costes de personal*

Pero este proyecto no supone solamente una inversión en cuanto a material humano, sino que la empresa deberá hacerse responsable de los gastos que puedan suponer los diversos materiales relacionados con las distintas herramientas necesarias para la consecución de los resultados que pretende conseguir este proyecto.

Los detalles de estos gastos se muestran en la Tabla 3 a continuación:

*DEFINICIÓN DEL TRABAJO*

<b>Recurso</b>	<b>Precio</b>
Licencia Visual Studio	500 €
Licencia BrowserStack	200 €
Licencia GSuite	200 €
Piezas de Hardware	1000 €
Documentación	100 €
Miscelánea	300 €

*Tabla 3. Costes materiales*

## Capítulo 5. DIAGNOSTICO WEB

### 5.1 ESTUDIO TRÁFICO

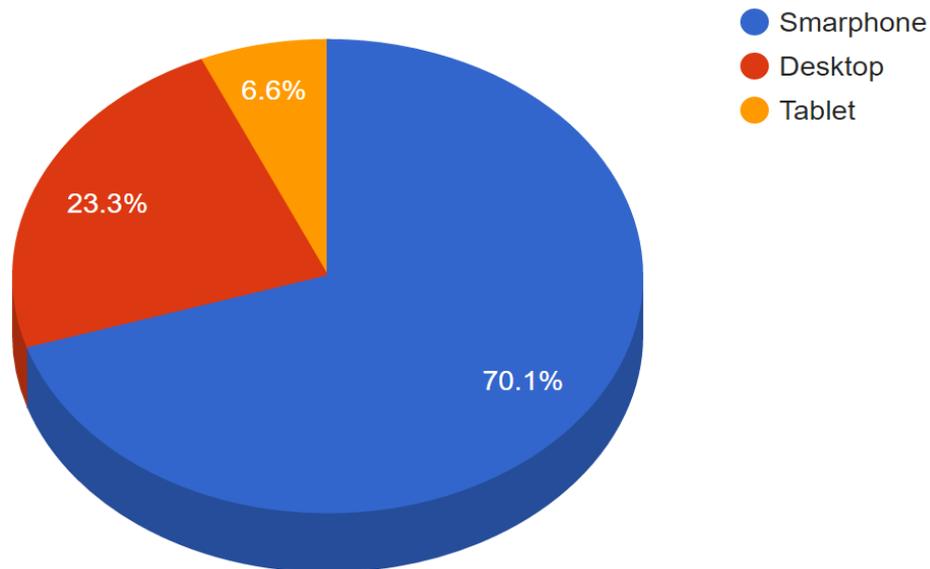
Si queremos diseñar un plan de mejoras para nuestra web, uno de los parámetros que vamos a necesitar analizar es quien son nuestros usuarios y de que manera se comportan. Para este estudio vamos a utilizar la herramienta de Google Analytics y ver desde que tipo de dispositivos se realiza el tráfico a la web.

Dentro de esta herramienta, tenemos un histórico de cómo se ha producido el tráfico en nuestra web. Gracias a esto, vamos a escoger un periodo de un mes para asegurarnos que los resultados sean lo suficientemente significativos, emplearemos los datos de tráfico desde el día 15 enero de 2017 al 15 febrero de este mismo año. Los resultados obtenidos son presentados en Tabla 4:

Categoría del dispositivo	Numero de sesiones	Porcentaje de representación
Smartphone	10,5 M	70,00%
Desktop	3,5 M	23,33%
Tablet	1 M	6,66%
<b>TOTAL</b>	<b>15 M</b>	<b>100,00%</b>

*Tabla 4. Categorización de las visitas (15/01/2017 hasta 15/02/2017)*

**Distribución del número de visitas en función del dispositivo**



*Figura 16. Distribución del número de visitas en función del dispositivo en la web de Europa Press  
(15/01/2017 hasta 15/02/2017)*

Gracias a la Figura 16, podemos observar como las visitas provienen principalmente de dispositivos móviles. Pero este dato no es del todo descriptivo de la situación en cuanto rentabilización de los usuarios. Como ya introdujimos, la visita como tal no define totalmente lo que es importante para las páginas web. Cabe la necesidad de saber cómo se comportan los usuarios una vez están en la web. Aquí entra en juego la permanencia y el *engagement* de cada usuario que accede a la web. Cuanto mayor sea el tiempo que pasa navegando por la web y visualizando distintas páginas del dominio, mayor será la recompensa económica por ello. Veamos entonces cuanto tiempo pasa cada usuario en la web, en función del tipo de dispositivo que esté empleando para ello.

Categoría del dispositivo	Páginas/sesión	Duración media de la sesión
Smartphone	1,27	00:01:47
Desktop	3,84	00:21:43
Tablet	1,72	00:02:46
<b>Media global</b>	<b>1,89</b>	<b>00:06:25</b>

Tabla 5. Categorización de la duración de sesiones

En la Tabla 5 nos encontramos una situación inversa en cuanto el tipo de dispositivo dominante. Son ahora las conexiones desde *desktop* las más determinantes. Si se suma el tiempo que han pasado estos usuarios navegando la web durante ese mes, obtenemos un total de 1250K horas de navegación por únicamente 312k M en móvil y 48k en tablets. Esto nos deja una situación como se representa en la Figura 17

Distribución del tiempo global de duración de sesión en función del dispositivo

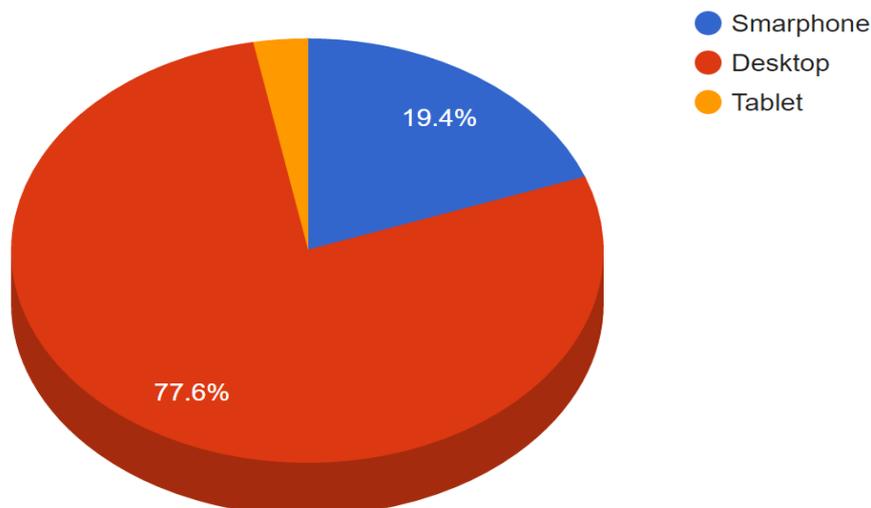
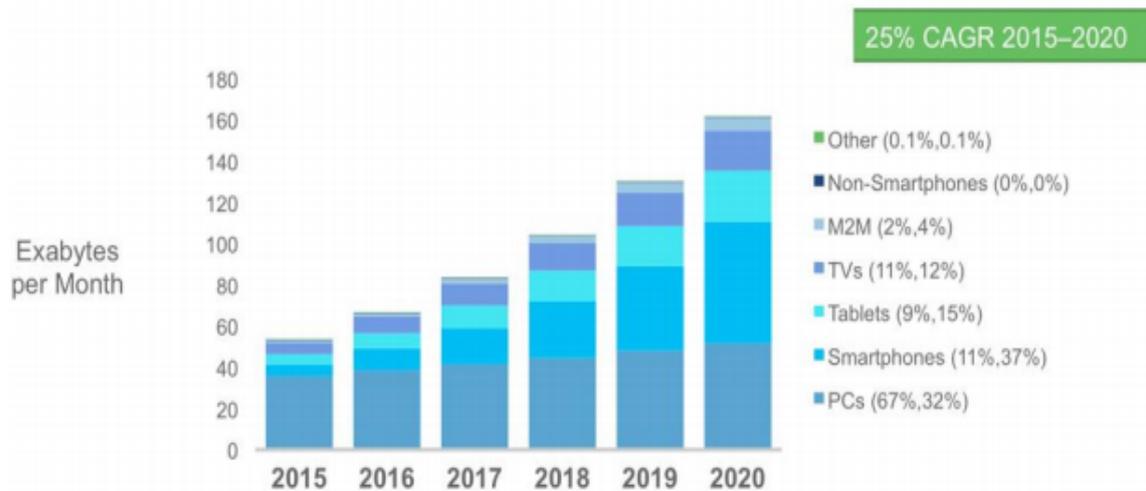


Figura 17. Distribución del tiempo global de duración de sesión en función del dispositivo en la web de Europa Press

Pero esta no es una realidad que sea única a nuestra empresa. Haciéndose eco de las previsiones hechas por equipos de investigación de Cisco, podemos observar cual es la posible evolución del peso de mercado de los dispositivos móviles.



Figures (n) refer to 2015, 2020 device share.  
Source: Cisco VNI Global IP Traffic Forecast, 2015–2020

Figura 18. Crecimiento del tráfico de datos en función del dispositivo [61]

Si seguimos las previsiones de Cisco sobre el crecimiento del tráfico según los dispositivos, Figura 18, se observa como esta situación de tener más cantidad de tráfico proveniente de dispositivos móviles no va a hacer más que incrementarse, por eso debemos tener una versión para el tráfico móvil lo más optimizada posible. Cada vez la gente consume datos de manera más móvil, consumen datos donde y en cuanto pueden mediante estos dispositivos tan extendidos.

## 5.2 ESTUDIO RENDIMIENTO

Hemos visto como es necesario tener en cuenta dos tipos de tráfico fundamental que acceden a nuestra página web. Tenemos, tanto tráfico desde ordenadores de sobremesa, que serían los que engloban la navegación clásica, por así llamarla, como usuarios que se conectan

desde dispositivos móviles, siendo esta principalmente gente que se conecta desde un Smartphone o una Tablet.

Por esto vamos a realizar un estudio de rendimiento de estos dos formatos de la página.

### **5.2.1 VERSIÓN DESKTOP**

Veamos entonces cual es el rendimiento que tiene nuestra web en los dispositivos de sobremesa. Para ello vamos a utilizar la herramienta de WebPageTest.

Veamos cómo se comporta la portada, cuya URL se corresponde con [www.europapress.es](http://www.europapress.es):

<b>Métrica</b>	<b>Tiempo en s</b>
<b>Tiempo de primer byte</b>	<b>0,150 s</b>
<b>Comienzo de render</b>	<b>4,289 s</b>
<b>DOM cargado</b>	<b>6,932 s</b>
<b>Tiempo de carga</b>	<b>14,152 s</b>

*Tabla 6. Estudio rendimiento portada desktop inicial*

Obtenemos que el tiempo de carga total corresponde a 14,152 s, lo que es un tiempo de carga muy grande.

Si observamos cómo se carga la página se puede claramente ver como la estructura visual de la página se carga antes de ese tiempo global de carga, como se puede ver en la Figura 19:

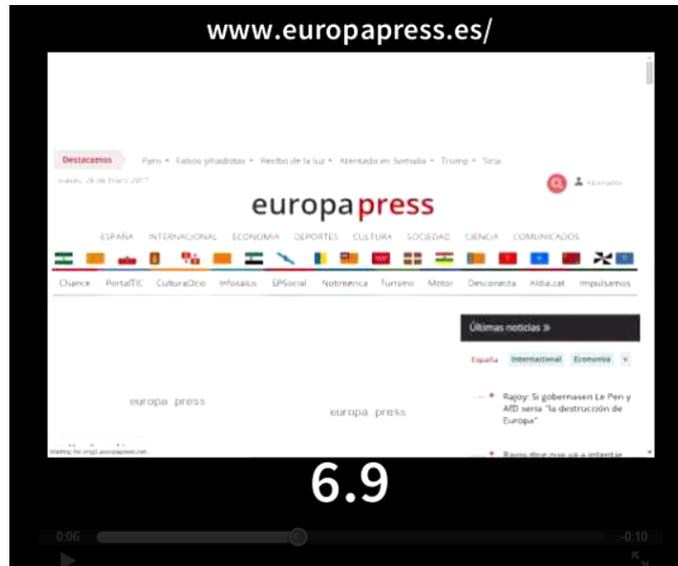


Figura 19. Captura del momento de carga del DOM y estilos

En esta figura, vemos como tanto los elementos del DOM como los estilos CSS, que componen el primer punto de vista de la web, han sido completamente cargados. No obstante, las fotos de las noticias se encuentran todavía cargando. Parece entonces que uno de los datos que tenemos que estudiar es de que modo afectan estas fotos al tiempo de carga de la web, ya que son estas las que están impidiendo que antes de 7 segundos el usuario tenga la sensación de que la web ha sido cargada.

Observemos entonces el desglose de contenido de la web:

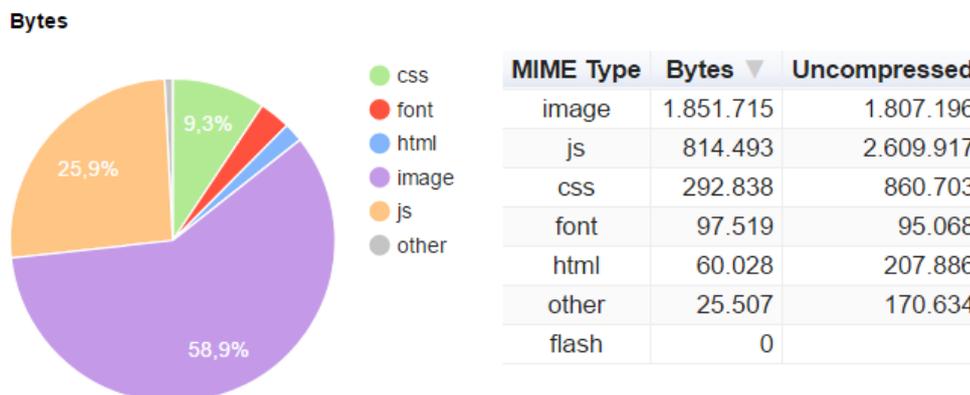


Figura 20. Desglose del contenido de la página

En esta Figura 20, se puede observar como el contenido principal que caracteriza la mayor parte de los datos de la web es el contenido de imágenes. Al ser archivos de gran cantidad de bytes, el uso de un gran número de imágenes afectará en gran medida al tamaño total de la página. Además, podemos deducir que al estar en una página cuyo contenido se basa en publicar noticias y esa estructura se basa en el uso de un titular más una imagen, la carga de imágenes es muy grande, al componerse la web de gran cantidad de ellas.

Observemos la carga de los 25 primeros elementos en detalle:

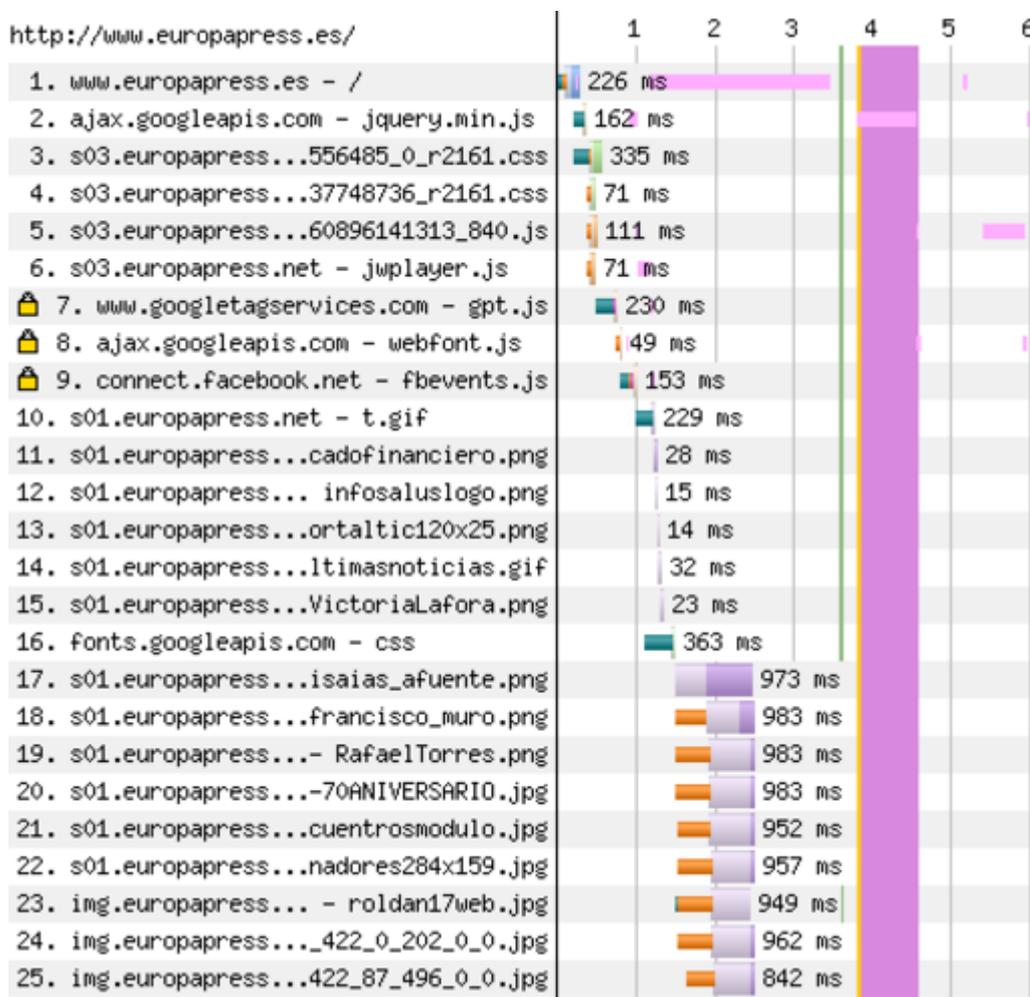


Figura 21. Detalle de los 25 primeros elementos

En la Figura 21, se vuelve a observar como son las imágenes las que mayor impacto tienen en el rendimiento de la web. De los primeros 25 elementos, 15 son de archivos de contenido

visual (extensiones .gif, .png y .jpg). Mientras el resto de contenidos tardan en cargar en tiempos del orden del milisegundo, las imágenes por su contra llegan a rondar el segundo de tiempo de carga. Podemos ver cómo estos contenidos se cargan mediante protocolo HTTP/1, sobre sockets no seguros, por ejemplo, para las *requests* 7,8 y 9, se pinta un candado amarillo, el cual sirve para indicar que el socket se ha ejecutado por https, lo cual no aparece en los sockets de imágenes.

Además, si tomamos una imagen y analizamos los detalles de su *request*, Figura 22 ,se confirma como se ha realizado mediante una conexión HTTP/1 estándar



Request #68	Details	Request	Response	Object
<b>URL:</b> http://img2.europapress.net/fotoweb/fotonoticia_20170126090027_644x400.jpg				
<b>Host:</b> img2.europapress.net				
<b>IP:</b> 95.100.248.139				
<b>Error/Status Code:</b> 200				
<b>Priority:</b> Low				
<b>Client Port:</b> 63161				
<b>Request Start:</b> 4.310 s				
<b>Initial Connection:</b> 18 ms				
<b>Time to First Byte:</b> 968 ms				
<b>Content Download:</b> 43 ms				
<b>Bytes In (downloaded):</b> 71.8 KB				
<b>Uncompressed Size:</b> 71.5 KB				
<b>Bytes Out (uploaded):</b> 0.4 KB				

Figura 22. Detalles de una request correspondiente a una imagen

Podemos entonces decir que no estamos obteniendo buenos resultados de carga al estar utilizando gran número de imágenes, cuya velocidad de carga es larga y al utilizar el protocolo HTTP/1 en la gran parte de las *requests* de carga de contenido.

Además de hacer este estudio de cómo se realiza la carga de la portada, vamos a ver que es lo que opina Google de nuestra página usando su herramienta llamada *PageSpeed Insights*, obtenemos el resultado representado en la Figura 23:

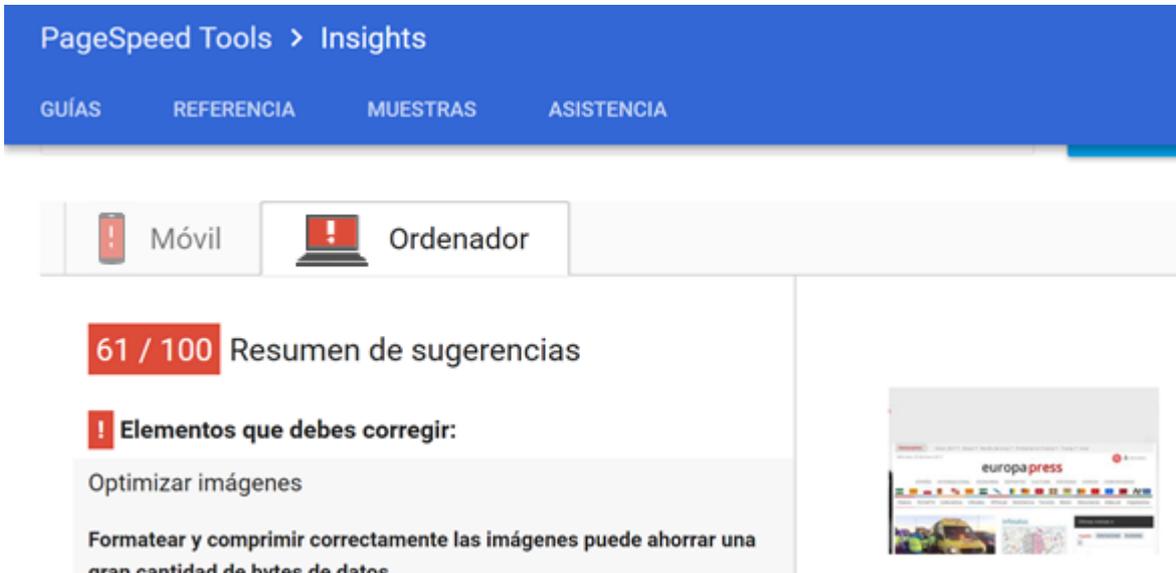


Figura 23. Resultado PageSpeed Insight

Este resultado es muy negativo ya que se según el baremo de Google nuestra página se enmarca en la categoría de las páginas que no están optimizadas según sus cánones.

### 5.2.2 VERSIÓN MÓVIL

Para el análisis de la versión móvil, se va a realizar el test sobre una noticia de última hora, a diferencia de hacerlo sobre la portada home de la web.

Obtenemos los resultados siguientes:

Métrica	Tiempo en s
Tiempo de primer byte	0,3626
Comienzo de render	6,287
DOM cargado	7,867
Tiempo de carga	10,048

Tabla 7. Datos rendimiento de carga noticia móvil antigua



Figura 24. Fotograma estudio de rendimiento noticia móvil antigua

A grandes rasgos los resultados obtenidos en móvil parecen ser parecidos a los encontrados en la versión de sobremesa. Además, tenemos los mismos defectos principales que hemos dicho anteriormente para la versión de *desktop*.

Los resultados obtenidos en la herramienta de análisis de velocidad de Google, son peores que los obtenidos en la versión de *desktop*, como se ve en Figura 25:

PageSpeed Tools > Insights

GUÍAS REFERENCIA MUESTRAS ASISTENCIA

Móvil Ordenador

43 / 100 Resumen de sugerencias

**4 Elementos que debes corregir:**

- Eliminar el JavaScript que bloquea la visualización y el CSS del contenido de la mitad superior de la página  
[Mostrar cómo corregirlo](#)
- Especificar caché de navegador  
[Mostrar cómo corregirlo](#)
- Habilitar compresión  
[Mostrar cómo corregirlo](#)
- Optimizar imágenes  
[Mostrar cómo corregirlo](#)

**3 Elementos que puedes plantearte corregir:**

- Minificar JavaScript  
[Mostrar cómo corregirlo](#)
- Minificar HTML
- Minificar CSS

europapress

Theresa May pide en su carta a la UE negociar a la vez el divorcio y la futura relación de Reino Unido

Internacional  
May apela a la unidad en Reino Unido en este "momento histórico" sin "marcha atrás"

Internacional  
Comienza el "Brexit" | Directo: Tsai: "No es un día feliz, ya es echamos de menos"

Figura 25. Móvil PageSpeed insight

### 5.3 COMPETENCIA

Una vez analizada la situación de nuestra web, es necesario otorgar a estos resultados de un contexto para poder deliberar si la situación de nuestra página es algo que se repite en los distintos dominios que pertenecen a la misma cuota de mercado, con los cuales competimos en cuanto a compartir un mismo *target* de usuarios.

Vamos a realizar los mismos análisis que hemos hecho anteriormente con nuestra web, con algunas webs de los distintos periódicos de mayor tirada, en sus versiones de *desktop*. Resumimos estos resultados a continuación:

<b>MEDIO</b>	<b>Tiempo de carga</b>
<b>elpais.com</b>	<b>19,241 s</b>
<b>elmundo.es</b>	<b>11,255 s</b>
<b>washingtonpost.com</b>	<b>5,981 s</b>

*Tabla 8. Análisis de la competencia.*

Estudiando la competencia, Tabla 8, podemos observar como el caso que tenemos en nuestra web no es muy distinto a las webs que nos son más cercanas. Si nos comparamos con los grandes medios de España, podemos observar que nuestra portada *desktop* es superior en cuanto a velocidad a la que tiene el diario El País, pero es 3 segundos más lenta en comparación con el resultado obtenido en la web de El Mundo.

Podemos entonces deducir que estos problemas son comunes al resto de páginas web de este sector, por problemas por los scripts de los anuncios y la ingente cantidad de imágenes, entre otros. Estamos entonces en una situación similar a la que existe en el sector donde nos movemos, el de los medios de prensa en España.

Pero si buscamos entre los medios que suelen destacar en cuanto a valoración de sus páginas web, en el ámbito internacional, podemos ver cómo es posible obtener una web de noticias exitosa que tenga una velocidad de carga rápida. La página web del Washington Post es vista como uno de los referentes dentro del sector de las páginas web de noticias, y, como se puede apreciar en la Tabla 8. Análisis de la competencia., esto está más que justificado. Esta web consigue mantenerse por debajo de los 8 segundos de tiempo de carga total y además a la hora de crear la sensación de que la carga a terminado a los usuarios se mantiene en aproximadamente 4 segundos.

Vamos a utilizar esta web como referencia y entrar en detalle:

Nada más entrar en la web, ya podemos ver la gran diferencia que tiene esta web con el resto de las que hemos analizado, pues esta corre sobre HTTPS, ver Figura 26.

 The Washington Post (WP Company LLC) [US] | <https://www.washingtonpost.com>

Figura 26. HTTPS del Washington Post

Esto es posible ya que todas sus conexiones se hacen sobre sockets encriptados, como vemos en la Figura 27, donde todas las *requests* están acompañadas por el icono del candado amarillo, que señala la existencia de conexión segura:

```
2. www.washingtonpost.com - /
3. www.washington...onts.0.latest.css
4. www.washington...onts.1.latest.css
5. www.washington...onts.2.latest.css
6. www.washington... - 9e1acbfb3f.css
7. www.washingtonpost.com - style.css
8. www.washington...m - b1811100e1.js
9. www.washington...com - head.min.js
10. www.washington...m - 95f829e22e.js
11. www.washington...m - 70b9918770.js
12. www.washingtonpost.com - head.js
13. www.washington...ion_denied_v1.png
14. www.washington...post-white-2x.png
15. www.washington...post_black_32.png
```

Figura 27. Request seguras en Washington Post

Pero si nos preguntamos porque hay tanta diferencia con la velocidad al estar con HTTPS, es necesario observar con más detalle cada una de estas *requests*, tomamos como ejemplo el de una imagen .png, Figura 28:

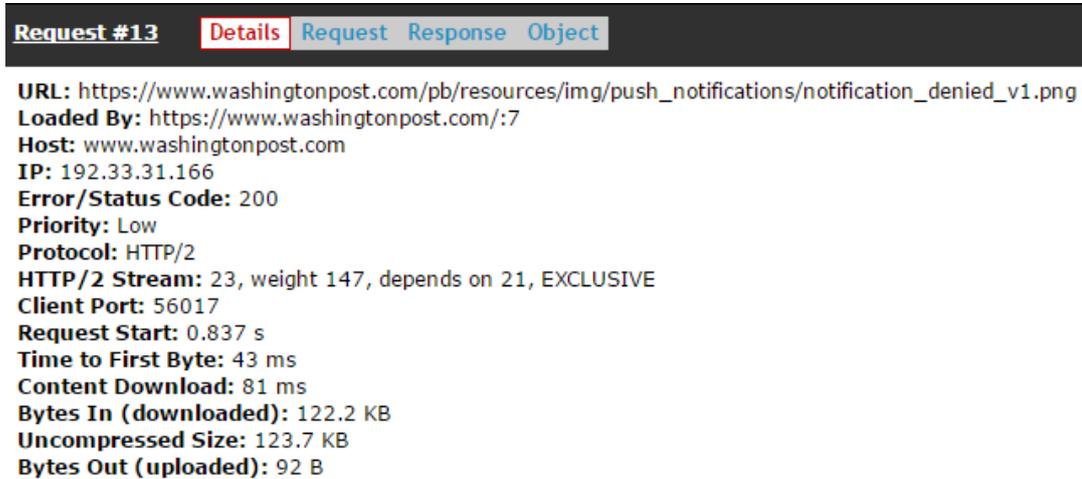


Figura 28. Request de una imagen del Washington Post

En esta figura podemos ver, como a diferencia de lo que analizamos en la Figura 22, esta *request* ya no va por HTTP/1, sino que corre por el protocolo HTTP/2, al igual que hacen el resto de las peticiones en esta web. Es por esto que podemos decir que el uso de HTTPS permite crear una ventaja competitiva en cuanto a velocidad de carga, ya que las comunicaciones podrán realizarse con HTTP/2, traduciéndose esto en una mejora significativa de la velocidad de carga de la web. Además, si observamos su formato, la sobrecarga de imágenes parece ser más liviana en comparación con los formatos que se usan en el resto de las webs de noticias.

## Capítulo 6. SISTEMA DESARROLLADO

### 6.1 ANÁLISIS DEL SISTEMA

La página web de Europa Press, como hemos explicado anteriormente, se basa en un proyecto programado en lenguaje Visual Basic .NET. La estructura fundamental de cualquier página de la web se basa en un sistema modular que podríamos dividir en tres elementos principales:

- *MasterPage*: Estas páginas maestras conforman lo que sería el esqueleto de cada página. Definen la estructura principal y en ella se definen zonas de contenido que se rellenarán con los distintos módulos o archivos de contenido con extensión .aspx
- Archivos de contenido: Estos archivos, con extensión .aspx, especifican siempre cual es la *MasterPage* en la cual se van a desarrollar, son los contenidos que aportan contenido al esqueleto. Por su parte también tienen un modelo modular, con sus propios *placeholders*, los cuales harán la llamada a los distintos módulos para que rellenen esos espacios de contenido. Estos archivos corresponden con las distintos modelos de páginas y entre ellos pueden compartir la misma *MasterPage*
- Módulos: Estos módulos son archivos con extensión ascx que sirven para instanciar una lógica concreta. Por ejemplo, en nuestra web existe un módulo llamado MasLeidas.ascx el cual sirve para implementar la lógica por la cual, en el caso de instanciar este módulo en un archivo aspx, este implemente de forma dinámica el contenido de un módulo que pinte en el HTML una zona donde se pinten las noticias más leídas en función de, por ejemplo, el canal de la web en el que nos encontremos. Estos módulos son los que dan forma a los contenidos repartidos a lo largo de la página, la rellenan de contenido específico según una lógica y son usados a lo largo de toda la web.

### Master pages at run time

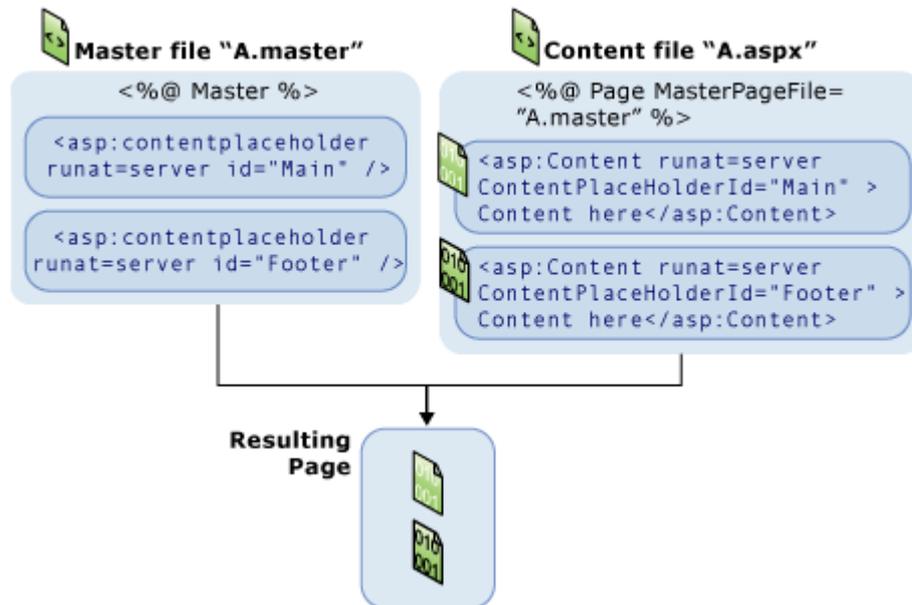


Figura 29. Estructura .master-.aspx[13]

Para tener un ejemplo de este tipo de archivos, tenemos en el ANEXO A el código de la Master Page usada en la portada de todos los canales, que se corresponde con la que se ha desarrollado en la realización de este proyecto, con la creación del nuevo formato de Portada. Si observamos la Figura 29, podemos ver como al inicio de cada archivo se especifica cual es la master que se va a utilizar como esqueleto para la hoja de contenido

A la hora de conectarnos a la web, el acceso a estos distintos archivos se define en un fichero llamado .htaccess. Este es un fichero de configuración sobre nuestro servidor Apache. La función de este archivo es la de realizar el encaminamiento de las *request* a los archivos aspx que se necesite ejecutar. Por ejemplo, tenemos:

```

RewriteCond %{HTTP:Host} www\.europapress\.es
RewriteRule ^/chance/[\w-/?]*(\d+)/(\d+)/$ /chance/noticias.aspx?ch=$1&pag=$2
[NC,L]
  
```

Este código se divide en dos partes y crea una regla de reescritura de URL entrante. El RewriteCond especifica en que, en el caso de que se cumpla esta condición, se entre en esta regla. En este caso se especifica que, si la URL que llega tiene como host el dominio de europapress.es, se entre en la regla.

La RewriteRule en sí se divide en dos. Primero tenemos la condición de URL, que especifica la URL literal que se tiene que cumplir. Para estas condiciones se hace uso de expresiones regulares que nos permiten crear un patrón de búsqueda que abarque una gran cantidad de opciones en una misma regla, recogiendo todos los casos que nos interesen. A partir de esta expresión regular se pueden coger distintos parámetros que se usen a la hora de traducir la URL.

En este ejemplo tenemos la expresión `^/chance/[\w\-\-]*?(\d+)/(\d+)/$`, esta expresión se divide en cinco partes:

- `^` indica que es el comienzo de la expresión regular
- `/chance/` es la primera parte evaluada de la URL. Aquí simplemente se pide que coincida con la palabra chance.
- `/[\w\-\-]*?(\d+)/` es la segunda parte de la URL que se evalúa. Esta expresión significa que se está esperando que llegue una cadena de caracteres seguida de un número, estando separados estos por un guion. El dato numérico no es obligatorio por tener `*?` delante. `(\d+)` es la parte importante de esta sección, `\d+` significa que recoge cualquier dígito. Al ir entre paréntesis, lo que se está indicando que es un grupo de captura, lo que permite guardar este dato y pasarlo como parámetro a la dirección donde redirijamos esta *request*.
- `/(\d+)/$` es la última condición evaluada. En esta se espera un dígito el cual va a ser guardado como la segunda variable

Por ejemplo, la URL <http://www.europapress.es/chance/cineymusica/2/> entraría en esta expresión.

En la segunda parte de la RewriteRule, `/chance/noticias.aspx?ch=$1&pag=$2`, se está diciendo que se ejecute el archivo `noticias.aspx`, que se encuentra en la carpeta `chance`, al cual le estamos pasando dos parámetros, `ch` y `pag`, equivaliendo `$1` con el primer parámetro que se ha capturado

En el ejemplo de URL <http://www.europapress.es/chance/cineymusica/2/>, lo que estaría pasando es que este archivo va recibir como valor del parámetro `ch` *nothing* y para el parámetro `pag`, el valor será 2. Este caso es el de paginación de las portadas de los subcanales del canal `chance`. El parámetro `ch` corresponde al canal mientras que el parámetro `pag` se refiere al número de paginación de esa portada. Aunque estemos pasando el valor *nothing* para el parámetro `ch`, no pasa nada puesto que la detección de a que canal se corresponde está implementada en la lógica del programa, pero no vamos a entrar en detalle ya que está fuera del código del `.htaccess`.

Finalmente, encontramos `[NC, L]`, esta parte de la regla es conocida como los *flags*. Estos *flags* especifican que se debe de hacer cuando se cumpla la condición. En este caso son dos, `NC` quiere decir que esta condición no distingue entre mayúsculas o minúsculas y el *flag* `L` que especifica que si esta regla se ha cumplido se ha llegado al fin de la evaluación de esta URL, ninguna regla más va a ser procesada.

Siendo conocedores de la estructura fundamental del sistema, podemos saber dónde y de que manera es posible introducir las modificaciones en el proyecto.

## **6.2 DISEÑO**

Teniendo en cuenta lo analizado en el Capítulo 5. y los conceptos teóricos estudiados en los capítulos previos de este trabajo, vamos a diseñar un plan de mejoras de la página web.

Aquí presentamos a grandes rasgos cuales van a ser las modificaciones que se van a introducir en la web. En el apartado 6.3 Implementación, se entrará en mayor detalle en cada desarrollo, con su justificación correspondiente.

Uno de los problemas que hemos visto en el análisis de la portada de Europa Press, es la gran cantidad de imágenes y distintos problemas de carga debidos a la estructura de esta. Por este motivo, se ha decidido crear un nuevo formato de portada. Este formato nuevo va a implementar mejoras tanto en el aspecto visual como en el aspecto de rendimiento. Se va a controlar el número de imágenes que se muestren, intentando cumplir de manera efectiva la necesidad de dar contenido visual con cada titular a la hora de hacer una web de noticias visualmente atractiva para los usuarios, gracias a la reducción del contenido. Se considera que el antiguo formato estaba demasiado sobrecargado, por lo que se va a buscar un diseño más minimalista, que no saque mucha información condensada en un pequeño espacio. Buscamos así aumentar la calidad a costa de reducir en cantidad.

Al encontrar problemas a la hora de la carga, por culpa de las imágenes, un método que es muy útil en estos casos es el de implementar lógica JavaScript para evitar cargar una ingente cantidad de imágenes, en el caso de que no sea necesaria su descarga por no llegar a ser visualizadas en ningún momento. Es por esto que se va a implementar en las imágenes que se encuentran por debajo del primer punto de vista de la web la función llamada *Lazy load* de imágenes. Esto nos va a suponer un ahorro en carga de imágenes, ya si se prevé que un usuario no llegue jamás a visualizar ese contenido no van a cargarse.

Como hemos visto en el estudio de la competencia, se ha observado como las páginas que presentan un mejor rendimiento tenían implementado HTTPS. Gracias a tener todas sus conexiones corriendo sobre sockets seguros, se beneficiaban de la posibilidad de operar siguiendo el protocolo HTTP/2, lo que se traducía en una mejora muy notable en los tiempos

de carga. Es por esto que vamos a buscar implementar a lo largo de toda nuestra página web el máximo número de conexiones HTTPS. Para ello en un primer momento vamos a ejecutar estos cambios sobre los recursos que más afectan nuestra velocidad de carga las imágenes. Para posteriormente, implementarlo en el resto de elementos. Para ello vamos a tener que emplear CDN de contenido, los cuales se explicarán posteriormente.

En cuanto a la versión móvil vamos a realizar tres modificaciones para mejorar su rendimiento.

La primera de ellas será una prueba que no busca directamente que la página sea más rápida en cargar por completo, sino en engañar al ojo humano mediante trucos con CSS, priorizando la carga del primer punto de vista. Usaremos CSS introducido en línea en la cabecera del HTML para solventar un problema de largo calibre que existe con los archivos CSS en la versión móvil de la web.

Además, se va a implementar un nuevo formato para las conexiones móvil usando el *framework* AMP en nuestra web. Esto se hará únicamente para el formato de noticias y no con las portadas o el resto de las páginas de contenidos de la versión móvil.

Retomando la optimización de las imágenes en la web, se implementará el uso de un formato de imágenes desarrollado por Google, el WebP, que se complementará con la implementación de un sistema de imágenes responsivas *reales* que mejoren el rendimiento en función de las dimensiones de los dispositivos móviles.

## **6.3 IMPLEMENTACIÓN**

### **6.3.1 FORMATO NUEVO**

La portada antigua que se usaba en la dirección <http://www.europapress.es>, ver Figura 30, se caracterizaba por un diseño en el que dominaba el contenido de imágenes, por el cual se priorizaba mostrar una imagen siempre que existiera un titular. Si hacemos un recuento del número de imágenes que se emplea en este formato nos encontramos con un dato que ronda el centenar de elementos de este tipo. Esta sobrecarga se debe en gran medida a que este formato se compone por tres columnas, en las cuales se hace el display de todas las noticias mediante el formato de un titular acompañado por una imagen.

Como vimos en el estudio de rendimiento de la web, esto generaba una gran cantidad de datos que se deben cargar, ya que además al ser imágenes, su tamaño con respecto a otros elementos, como por ejemplo pudiera ser una hoja de estilos CSS, es mucho mayor.

Ante este problema se plantea crear una portada que, en vez de tres columnas, se componga de dos, siendo la principal compuesta por el formato foto-titular, y la segunda empleada para añadir distintos módulos como pudiera ser un banner de noticias o el módulo de última hora. Además, no se va a forzar a que todos los titulares tengan fotos que los acompañen.

Con este cambio conseguimos reducir aproximadamente de media un 50% el número de imágenes que se deben cargar en el render de esta portada, este dato no es siempre el mismo como lo era el anterior puesto que ahora es posible hacer un display de una noticia que no tiene imagen.

Veamos a continuación una comparación entre estos dos formatos, Figura 30. Formatos portada Europa Press :

### Antiguo Formato

### Nuevo Formato

Figura 30. Formatos portada Europa Press

Para realizar esta nueva portada hemos tenido que seguir el patrón estructural de código de la web, que hemos explicado en el análisis del sistema. Para ello, se ha creado un nuevo archivo. master, correspondiente con el código contenido en el ANEXO A, una hoja de contenido llamada Default.aspx, que a su vez se compone de distintos módulos que se han reutilizado de la antigua portada y un diverso número de módulos de funcionalidad.

Además, se ha modificado el .htaccess para redirigir la url de europapress.es a este Default.aspx. No solamente se han generado estos archivos, este cambio conlleva mucho trabajo de diseño gráfico con un largo desarrollo de hoja de estilos CSS y actualización de los scripts que se utilicen en este nuevo formato.

Pero este no ha sido el único cambio realizado en los formatos. Una de las dificultades a la hora de trabajar con este proyecto es su tamaño. Dentro de la web de Europa Press, existe una estructura que se divide en distintos portales, los cuales se componen de canales y a su vez, estos canales pueden contener subcanales, esto se puede ver en la Figura 31.

Estructura de Portales y Canales del grupo Europa Press

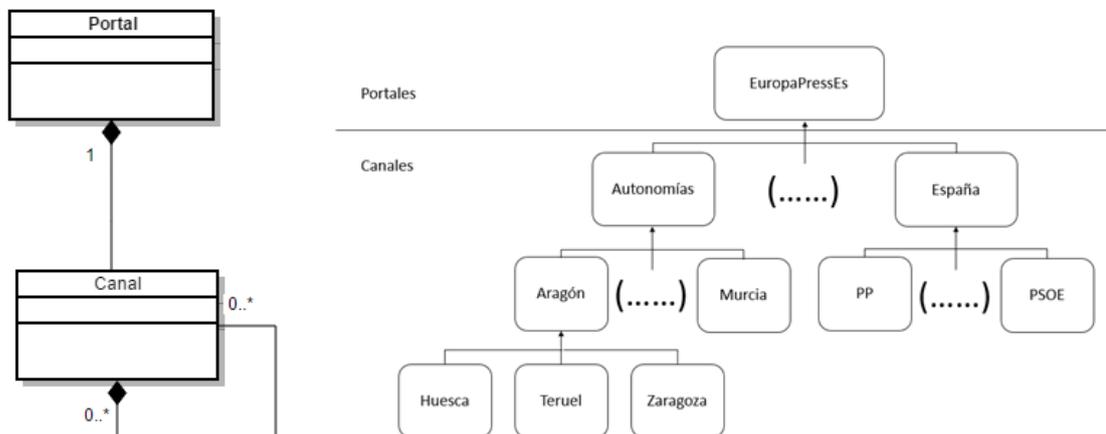


Figura 31. Estructura de Portales y Canales del grupo Europa Press

Estos canales no comparten un mismo formato, lo que crea una gran dificultad en materia de *troubleshooting*. Si alguno de los formatos tenía algún problema por algún fallo de uno de sus módulos personales podía acarrear problemas de rendimiento en la web. Tener tal número de páginas distintas, que hacen llamadas a sus propios módulos, creaba un sistema

muy cargado en elementos, lo que suponía muchas veces el retraso del arreglo de fallos de los canales menos importantes o de los fallos menos críticos. Es por esto que, aprovechando la migración del formato de la portada principal, se ha decidido aplicar la .master diseñada para todos estos canales. A diferencia de la portada general, estos utilizan como hoja de contenido el archivo Portada.aspx. De esta manera, se renuevan además formatos anticuados, como por ejemplo el del canal epturismo, ver Figura 32

Antiguo Formato



Nuevo Formato



Figura 32. Cambio formato portada del canal Turismo

### 6.3.2 *LAZY LOAD*

A la hora de conectarse a una página web, todo se carga de manera secuencial, procediendo a seguir el código del archivo HTML de cada página. Esto puede conllevar algunos tiempos de ralentización de la totalidad del contenido de la página, que pueden estar siendo provocados por elementos que tal vez no lleguen a ser visualizados por los usuarios. Mediante jQuery, existe un método bastante extendido llamado *lazy load*. Este patrón de diseño provoca una posposición de la carga de elementos, en este caso de imágenes, hasta que no se sepa que vayan a ser utilizados. En la web de Europa Press, existía un problema por el cual, al iniciar la carga de la página, se establecía un tiempo de demora debido a un número de imágenes de gran tamaño que se cargaban antes que algunos recursos importantes. El problema consistía que se estaba retrasando la carga de la página por parte de elementos que se encontraban en el pie de página. La mayoría de estas imágenes pertenecían a un módulo de imágenes llamado fotos del día, localizado al final de la web. Para ello se establece entonces que las imágenes que estén implementadas mediante *lazy load* se van a cargar en cuanto el usuario haga *scroll* a una posición que se encuentre a cierta distancia de cada una de ellas, dejando un margen suficiente para que, si se realiza el *scroll* de manera continuada hacia este elemento, el usuario no pueda apreciar que este recurso no estaba cargado desde el principio cuando llegue a la posición de este.

A partir de esta técnica, los usuarios que estén únicamente interesados en ver la primera plana de la portada no van a tener que descargar este contenido puesto que, para visualizar únicamente el primer punto de vista no queremos cargar datos innecesarios del pie de página.

Antes de que las imágenes se carguen, lo que se realiza es emplazar un gif [14] temporal. Por defecto este gif, es una animación de carga, pero como buscamos optimizar en velocidad de carga hemos decidido buscar cual podría ser el gif de menor tamaño compatible con esta funcionalidad, ya que damos por hecho que en ningún caso el usuario va a poder ver que las imágenes *lazy* no están cargadas cuando lleguen hasta ellas. Para ello se va a utilizar el gif más pequeño que se puede generar, el cual consiste en un punto incoloro, puede aparecer en distintos tipos de colores dependiendo del decodificador de GIF que se use, siendo por

defecto de color negro. Su tamaño es de tan solo 26 bytes, lo que supone un ahorro significativo de datos.

Para implementar esta tecnología hemos usado el código JavaScript del ANEXO B. Esta funcionalidad debe de ser incluida en todos los aspx que deseemos.

Para que esta función haga su cometido es necesario, a la hora de crear cada elemento que vaya a ser afectado por esto, definirlo con la estructura que tenemos en el código a continuación:

```

```

En este ejemplo vemos como la imagen que queremos que no se cargue, debe contener tres atributos especiales. Dentro del tag HTML <image> tenemos estos tres elementos:

- Src : a la hora de crear la imagen lo que nos interesa es que la fuente de donde se carga la imagen, no sea la imagen que debe de verse cuando se llega a esa parte del DOM. Como se puede ver, se ha introducido la referencia al gif que explicamos anteriormente. Entonces lo que está ocurriendo es que este gif está en el lugar donde debería ir una imagen de mucho mayor tamaño
- Data-src : este atributo es el que va a permitir al código JS saber que imagen es la que necesita cargar e introducir en el atributo src en cuanto el usuario alcance la zona determinada de carga de elementos implementados con *lazy load*. Lo que hace entonces es ver el contenido referenciado en este atributo como la imagen real que debe de pintarse si el usuario alcanza ese punto.

- Class = "lazy" : esta clase sirve de referencia al jQuery para identificar que esta imagen debe de ser afectada en cuanto se llega a la coordenada de carga de imágenes del tipo *lazy*.

Además, es importante especificar en estas imágenes sus dimensiones, mediante el uso de los atributos *width* y *height*. Esto se debe a motivos de SEO y de experiencia de usuario. Al introducir una imagen de pequeño tamaño que no corresponde con la que debería de ir, es necesario indicar cuales son las dimensiones que va a tener ese contenido ya que, en el caso de que esa imagen se cargue de manera ineficaz, se puede producir cambios en las posiciones de los elementos de la web en cuanto se carga la imagen, al cambiar la imagen de ese contenedor, que tenía el tamaño del *gif* hasta ese instante.

### 6.3.3 HTTPS

Para poder realizar la migración a HTTP/2, es necesario que las páginas web funcionen mediante conexiones encriptadas, instanciando HTTPS en la página web. Esto supone que todos los recursos se tienen que cargar en sockets seguros, para que la web sea considerada HTTPS.

El cambio que vamos a realizar es en la manera de carga de las imágenes y otros recursos estaticos. Las imágenes en Europapress se encuentran en el servicio de *cloud* de Amazon web Services.

Para conseguir un mejor rendimiento en la descarga de estas imágenes, a la vez de que se haga con HTTPS, vamos a implementar una solución que consiste en el uso de una Red de entrega de contenidos, o CDN.

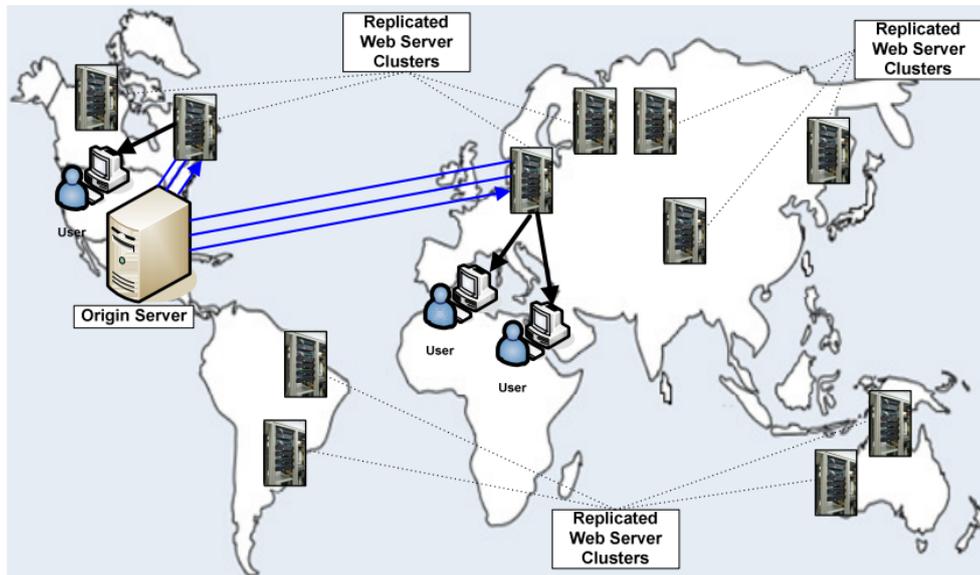


Figura 33. Estructura abstracta de un CDN

Como observamos en la Figura 33. Estructura abstracta de un CDN, la CDN se compone de una red distribuida de servidores localizados en diversas partes de la geografía mundial, lo cual permite tener distintos puntos de conexión, donde todo el contenido deseado está cacheado. Esto otorga una mejoría en cuanto a la posible localización de cada usuario, conectándose cada vez al servidor de contenido más cercano a él. Esto otorga entonces una mejor latencia, ya que ya no existe un único punto de conexión por el cual, en el caso de una conexión desde una posición lejana al servidor, nos encontremos con tiempos de RTT muy altos. Además, gracias a este CDN, podemos distribuir contenido mediante SSL, lo cual ellos se encargan de proveer.

Para ello, se ha tenido que hacer un estudio de mercado el cual ha dejado tres alternativas. Va ser necesario deliberar entonces que servicio CDN es el que más nos conviene contratar. Las opciones que hemos decidido estudiar son las de las empresas Level 3, Amazon y Akamai.

Para ello nos hemos puesto en contacto con estas empresas y hecho una petición de prueba de sus servicios, pidiendo que hagan temporalmente de CDN de todas las imágenes de la web.

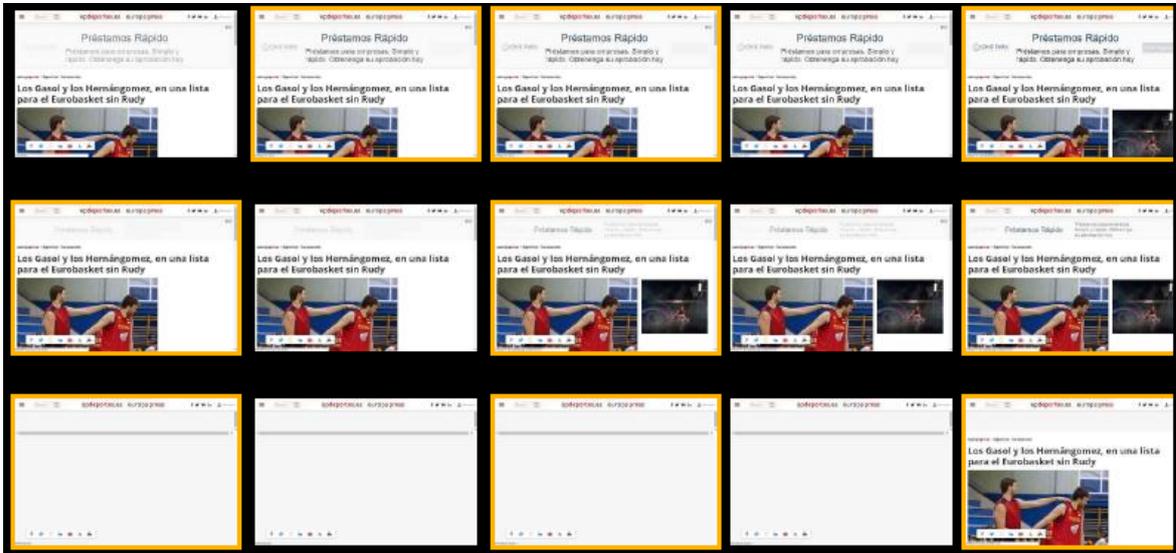
Hagamos una comparativa previa de los servicios que otorga cada una de ellas, con los datos teóricos.

CDN	AKAMAI	LEVEL 3	AMAZON
<b>Compresión GZIP</b>	✓	✓	✓
<b>HTTPS</b>	✓	✓	✓
<b>HTTP/2</b>	✓	⊗	✓
<b>Estadísticas en tiempo real</b>	⊗	✓	⊗
<b>Precio</b>	<b>Estándar</b>	<b>Económico</b>	<b>Elevado</b>

*Tabla 9. Comparativa de distintos CDNs*

El análisis presentado en la Tabla 9. Comparativa de distintos CDNs no es suficiente para realizar esta decisión. por el momento, aun ofreciendo una solución económica, Level3 parece el menos interesante ya que no ofrece el servicio HTTP/2 que nos interesa en esta situación. Necesitamos entonces hacer test sobre los resultados que nos van a dar realmente estos sistemas, es por ello que hemos creado una página .aspx que nos permita realizar pruebas con cada tipo de servicio.

Este módulo de pruebas permite introducir cualquier url de un HTML que pertenezca a la web e ir cambiando desde que CDN queremos que se carguen de las imágenes de ese HTML. Vamos a usar este módulo junto a la herramienta Web Page Test para comparar el comportamiento de cada CDN.



*Figura 34. Comparativa CDNs*

En la Figura 34. Comparativa CDNs, podemos observar una de las pruebas que hemos realizados. Las primeras dos filas corresponden con recursos cargados desde Akamai y Amazon, y la última corresponde con el servicio Level 3. Como era de esperar por el estudio teórico que se ha hecho sobre estos servicios, Level 3 era el que daba un peor servicio en cuanto a rendimiento de carga de los datos, al no dar soporte para HTTP/2. De media, hemos encontrado como sus imágenes suelen tardar en cargarse aproximadamente un segundo más que las de los competidores. Al tener como prioridad la mejora de nuestra web, esta diferencia tiene un orden de magnitud demasiado grande para no tenerla en cuenta, aunque se oferte por un precio por debajo de las otras opciones. Entonces nos cabe determinar entre Amazon y Akamai. Después de realizar un número de test, no observamos que exista una gran diferencia en el tiempo de carga de las imágenes de estos dos servicios, por lo que la elección entre estos dos servicios se va a decantar por cual ofrece el precio más competitivo, puesto que el resto de servicios que ofertan son muy similares. Finalmente, se ha decidido contratar el servicio de Akamai, al ser la opción más económica.

Originalmente, se tenían más de un servidor de entrega de contenidos. Los dos *hostnames* de descarga de las imágenes en toda la web eran *img.europapress.net* y *img2.europapress.net*. Con esto lo que se conseguía era la posibilidad de hacer carga con

*domain sharding*. Esto va a ser modificado ya que, como nuestro objetivo con este cambio es el de implementar el protocolo HTTP/2 en la carga de imágenes, esta práctica ya no es necesaria. Por lo cual, solamente utilizaremos un host para todas las imágenes que se carguen en la página.

Puesto que este servicio va a ser facilitado por Akamai, es necesario entonces cambiar a un *hostname* de este servicio. El nuevo *hostname* va a ser entonces `imgeuropapress.akamaized.net`.

Habrà que afectar este cambio a todas las imágenes del proyecto. Por ejemplo, pasamos de `http://img.europapress.net/fotoweb/fotonoticia_20170406145004_640.jpg` a `https://imgeuropapress.akamaized.net/fotoweb/fotonoticia_20170406145004_640.jpg`.

Esta migración también se va a realizar para contenidos estáticos, esto engloba por ejemplo a los *favicons*, que se corresponden con los iconos asociados a cada página, que los navegadores suelen pintar en las pestañas, Figura 35. Favicon de Europa Press

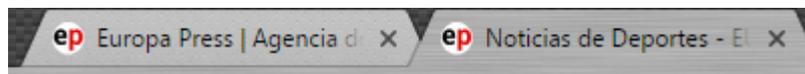


Figura 35. Favicon de Europa Press

Por ejemplo, `http://s01.europapress.net/imagenes/estaticos/favicons/ep3/icon/favicon-32.ico` ahora es:

`https://s01europapress.akamaized.net/imagenes/estaticos/favicons/ep3/icon/favicon-32.ico`  
y `http://s03.europapress.net/css/estilos_1152921504640401413_33554432_r2254.css` ahora es:

`https://s03europapress.akamaized.net/css/estilos_1152921504640401413_33554432_r2254.css`.

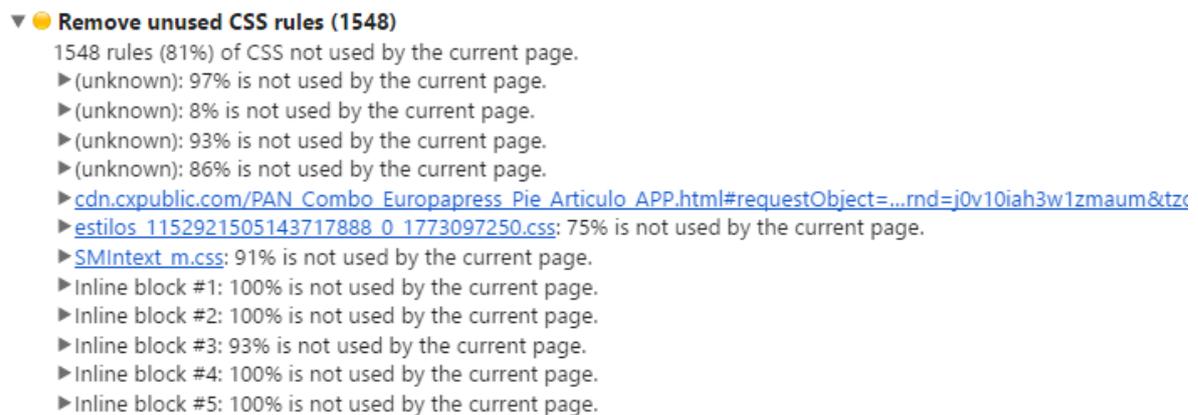
### 6.3.4 CSS INLINE

Hemos visto con el estudio del protocolo propuesto por Google AMP, como para acelerar la carga de la página móvil se propone el establecimiento de código CSS *inline*, ya que buscan no parar la carga de la página por tener que cargar hojas de estilos. Aconseja el uso de CSS

*inline* en el mismo HTML plano, en su cabecera, siempre y cuando su tamaño sea inferior a 500kB. Siguiendo estas recomendaciones, se ha planteado la opción de desplegar un mecanismo parecido al propuesto por AMP en las versiones móvil de la web.

Esta necesidad aparece ya que en la actualidad la gestión del CSS en la versión móvil de la página web no es óptima, ya que, en gran parte de los formatos de toda la web, incluso en los diversos portales, se carga una misma hoja de estilos con todos los estilos que existen en la mayoría de los estilos de la web. Esto genera una carga desorbitada de datos inútiles a la hora de hacer el render de la página, ya que a la hora de generar nuevo contenido se añade directamente en esta hoja, sin eliminar los estilos anteriores que se quedan obsoletos. Esta hoja de estilos se llama `estilos.css`, tiene un tamaño de 439 kB y supone el documento de mayor tamaño del proyecto.

Mediante al uso de la herramienta de Google Chrome de Audits, ver Figura 36, podemos observar la cantidad de datos de las hojas de estilo que no llegan a usarse a la hora de hacer el render de la página



*Figura 36. Google Audit sobre la portada móvil*

Si estudiamos todos estos datos podemos calcular cual es la media de utilización de los estilos CSS que tenemos en nuestra web móvil, en los diversos formatos que existen.

Obtenemos que se está utilizando de media menos del 75% de estos estilos en nuestras páginas. Este dato es prueba de cómo no se está haciendo una buena gestión de la carga de estilos CSS.

Para ilustrar esta situación veamos los datos de distintos formatos de la web móvil:

<b>Formato</b>	<b>Tamaño CSS</b>	<b>Porcentaje de uso</b>	<b>Datos innecesarios</b>
<b>Portada</b>	<b>434 kB</b>	<b>9,00%</b>	<b>399 kB</b>
<b>Noticia</b>	<b>144 kB</b>	<b>25,00%</b>	<b>108 kB</b>
<b>Videos</b>	<b>139 kB</b>	<b>25,00%</b>	<b>104 kB</b>
<b>TOTAL</b>	<b>717 kB</b>	<b>15,00%</b>	<b>611 kB</b>

*Tabla 10. Estadísticas de uso del contenido CSS descargado*

En la Tabla 10. Estadísticas de uso del contenido CSS descargado, podemos observar como a la hora de realizar una conexión a una página de los formatos del portal Europa Press, estamos teniendo en la cabecera una carga de una hoja de estilos con mucha información que no tiene importancia, debido a la cantidad de datos que no se van a emplear a la hora de realizar el render de la página. Estamos pues incurriendo en un error muy crítico en cuanto al rendimiento de una página web. Estamos parando la carga de la página web para descargar un archivo no optimizado en cuanto al uso de datos.

Como respuesta a esto, vamos a seguir el modelo CSS *inline*. Buscamos entonces evitar la pausa de descarga de la página web poniendo los estilos necesarios en el primer punto de vista de la página en el archivo HTML. La particularidad de este modelo va a ser la de evitar que cuando se renderice la página aparezcan elementos sin estilos en el primer punto de vista del usuario. Estamos jugando con la percepción de los usuarios, ya que, aunque no estén cargados todos los estilos, el usuario tiene la sensación de que se ha completado la carga de

toda la página web, aunque el resto de los elementos de la página no se le hayan afectado los estilos correspondientes. La velocidad de carga de una web se vincula a la velocidad aparente y este método se basa en únicamente beneficiar este parametro. Se puede dar el caso de que la página tarde un tiempo X en cargar la totalidad de elementos , siendo X un número fuera de los cánones aceptables de percepción de velocidad humana de carga, pero al realizarse la carga completa del primer punto de vista en un tiempo inferior a X, el dato de carga completa no es importante en el análisis de la velocidad de la web, siempre y cuando la diferencia entre el tiempo X y el de la carga del primer punto de vista sean lo suficientemente cercanos para que los usuarios finales pierdan la sensación de carga entre estas dos etapas. Estamos básicamente optimizando la carga del primer punto de vista. Para ello vamos a dividir los estilos de cada formato en dos bloques de estilos CSS. Primero tendremos unos estilos considerados como críticos, estos estilos van a ser los que necesitemos para que este mecanismo no sea apreciable por los visitantes, son los estilos del primer punto de vista. Estos se van a introducirse en el tag <HEAD> del archivo HTML de manera *inline*, para tener cargados estos estilos antes de renderizar la página.

El resto de los estilos se introducirán al en el pie de página, cargando el archivo de hoja de estilos que contiene todos los estilos.

Como defecto de este enfoque, seguimos teniendo el problema de la carga de estilos no empleados, pero esto es necesario ya que el mantenimiento y la inclusión de los formatos se realizan en archivos CSS distintos, que actualizan los diseñadores de la web, y no existe un mecanismo de distinción exhaustivo de estilos en función de los formatos.

Por lo que se introduce aquí una posible problemática para futuros trabajos, que consistiría en un mecanismo para únicamente cargar los estilos que se vayan a utilizar en función del formato .aspx que se esté empleando de otros parámetros, en vez del estado actual que carga una gran cantidad de estilos innecesarios para evitar que haya elementos sin estilo.

Este cambio se ha realizado en la totalidad de los dominios de Europapress, lo que supone el diseño de 36 hojas de estilos *inline* para los diversos formatos que componen el proyecto Europapress móvil.

Se ha desarrollado la función `InlineCSSIncluder`. Esta función es llamada al ejecutar la `Page_Load` de la `MasterPageM.master.vb`, que corresponde con la master page que se emplea en la versión de móvil

```
Private Sub InlineCSSIncluder (canal As Canal)

    Dim CSS As String = ""
    Dim Path As String = Request.CurrentExecutionFilePath

    Select Case Path
        Case "/appmob/noticias.aspx" 'Portada EP

            Select Case canal.PortalId

                Case Portal.Portales.Hacerfamilia
                    CSS = /*código css inline*/

            /*31 casos distintos según el tipo de .aspx y el portal y/o canal*/

        End Select

    Page.Header.Controls.Add (New LiteralControl (CSS))

End Sub
```

### 6.3.5 AMP

En el Capítulo 3. , hemos visto los beneficios de emplear el *framework* desarrollado por Google. Es una buena oportunidad para diseñar páginas simples que sigan este canon para conseguir un rendimiento de velocidad de carga bueno a la vez que se consigue un mejor posicionamiento en la estructura de Google.

Para ello, vamos a crear una nueva rama en el proyecto llamada AMP. En ella va a ser necesario replicar la estructura introducida en el apartado 6.1 Análisis del Sistema, de master page, hoja de contenido y diversos módulos.

A la hora de crear una página AMP, tenemos que seguir las reglas de este *framework*. No es raro encontrar errores a la hora de generar contenido AMP, puesto que es una tecnología nueva y existen muchos elementos de HTML que no tienen soporte en AMP. Para poder

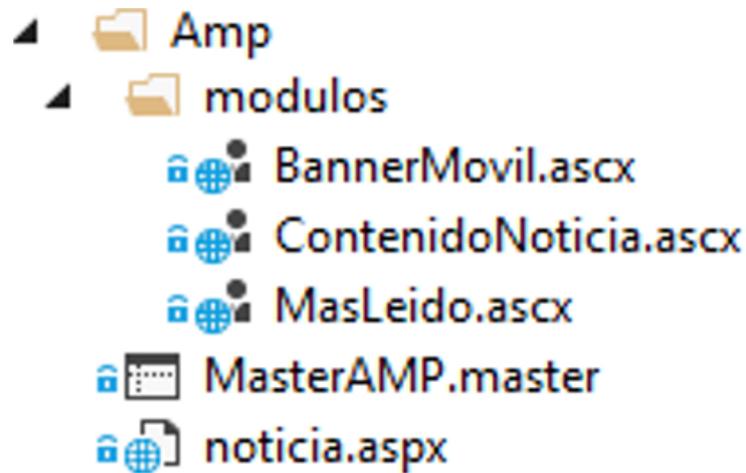
saber cuáles son los componentes compatibles con este formato hay que referirse a los documentos oficiales de Google [15]. A la hora de desarrollar, guiarse por esta guía es esencial para poder saber si los elementos que queremos implementar funcionarán en este formato.

La mayoría de las distintas etiquetas HTML tienen implementadas en este *framework* elementos equivalentes en cuanto a funcionalidad, que han sido redefinidos con el propósito de simplificarlos y que funcionen de manera más rápida, según las directrices de este estándar. Por ejemplo, si utilizásemos el *tag* `<img>` en una página AMP, esta nos daría un error. Lo que deberíamos estar usando en vez de esta etiqueta, es `<amp-img>`.

Uno de los temas más importantes a la hora de implementar este formato es la limitación del JavaScript. El hecho de no poder utilizar JavaScript limita mucho la versatilidad de las páginas. Aunque se pueda usar JavaScript asíncrono, por ejemplo, con el tema de los banners de publicidad, tenemos una limitación grande. Debemos entonces sacar toda la potencia posible del lenguaje CSS para tener una web animada y excluir algunas utilidades que no sean imprescindibles.

Al estar trabajando con una web que busca un retorno sobre la inversión a partir de los anuncios, los banners publicitarios cobran una gran importancia. AMP propone usar su etiqueta `<amp-ad>` ya que no se puede usar JavaScript, el cual es el estándar para la publicidad web. Esta etiqueta lo que implementa es una zona en la cual se carga un *iframe* donde se ejecuta el JS de ese anuncio. Esto supone que dentro de una página AMP, el contenido que va a cargarse último es el patrocinado. Esto supone que, a la hora del diseño, vamos a intentar no poner estos anuncios en el primer punto de vista para evitar que el usuario aprecie que tardan en cargar.

Esta implementación va a consistir en crear un formato de noticia independiente. No vamos a crear una versión entera en AMP de la web, puesto que la complejidad que tendría ese proyecto es demasiado alta, tampoco vamos a crear una portada desde la cual acceder a todo el contenido. Por el momento, vamos a crear noticias a las cuales se acceda desde Google a través de búsquedas concretas.



*Figura 37. Archivos implementación versión AMP de la web*

En la Figura 37, podemos observar los archivos que componen la estructura AMP implementada.

Es una estructura simplificada en comparación al formato noticia de la versión de móvil. Se compone en la cabecera por un menú de navegación de los contenidos de la web que permite acceder a los canales de la web en su versión móvil. Luego tenemos el contenido de la noticia, que se carga mediante el archivo ContenidoNoticia.ascx, el cual se compone de dos tipos de módulos en medio. Las noticias destacadas y los banners de publicidad, se gestionan con los otros dos archivos de la carpeta de módulos. La estructura principal de la noticia se compone del MasterAMP, anexado en el anexo C, y la hoja de contenido noticia.aspx. Podemos ver el resultado final en la Figura 38 y Figura 39. <amp-ad> dentro del contenido de una noticia



Figura 38. Ejemplo de noticia en formato AMP

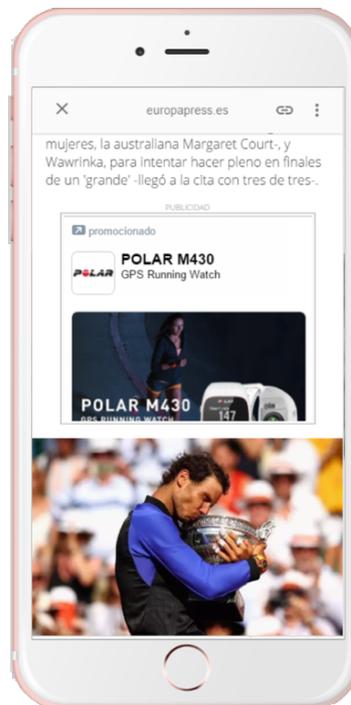


Figura 39. <amp-ad> dentro del contenido de una noticia

Finalmente, al ser un nuevo formato, es necesario crear nuevas reglas en el sistema de redirección de las peticiones. Hay que implementar todos los *rewrites* para que se redirija al nuevo módulo AMP, a continuación, presentamos las reglas introducidas:

```

RewriteCond %{HTTP:Host} amp\.europapress\.es
RewriteRule ^/otr-press/firmas/([a-z]+)/[\w-]*(\d{14})\.html?
/Amp/noticia.aspx?ch=01022&cod=$2&$3 [NC,L]

RewriteCond %{HTTP:Host} amp\.europapress\.es
RewriteRule ^/otr-press/cronicas/[\w-]*(\d{14})\.html?
/Amp/noticia.aspx?ch=01024&cod=$1&$3 [NC,L]

RewriteCond %{HTTP:Host} amp\.europapress\.es
RewriteRule ^/otr-press/tribunas/[\w-]*(\d{14})\.html?
/Amp/noticia.aspx?ch=01023&cod=$1&$3 [NC,L]

RewriteCond %{HTTP:Host} amp\.europapress\.es
RewriteRule ^/otr-press/[\w-]*(\d{14})\.html? /Amp/noticia.aspx?ch=274&cod=$1&$3
[NC,L]

RewriteCond %{HTTP:Host} amp\.culturaocio\.com
RewriteRule ^/([a-z-]+)/noticia-.*-(\d+)\.html /Amp/noticia.aspx?ch=${canales-
cultura-ocio:$1}&cod=$2 [NC,L]

RewriteCond %{HTTP:Host} amp.europapress\.es
RewriteRule ^/economia/[\w-]*(\d{14})\.html?(?:\?(.*))?$
/Amp/noticia.aspx?cod=$1&$2&ch=136 [NC,L]

RewriteCond %{HTTP:Host} amp.europapress\.es
RewriteRule ^/economia/(?:[\w-/*-]*-)?(\d+)/[\w-]*(\d{14})\.html?(?:\?(.*))?$
/Amp/noticia.aspx?ch=$1&cod=$2&$3 [NC,L]

RewriteCond %{HTTP:Host} amp\.europapress\.es
RewriteRule ^/salud/.*-([0-9]+)/noticia-.*-(\d+)\.html?
/Amp/noticia.aspx?ch=74&cod=$2 [NC,L]

```

## 6.3.6 GESTIÓN DE IMÁGENES

### 6.3.6.1 WebP

Google ha creado un formato de imágenes web, llamado WebP [16]. En la web, predomina el uso de imágenes con el formato JPEG, y según los datos que facilita Google, el formato WebP es entre un 25 – 34% más pequeño. Es por esto que nos interesamos a su implementación en las imágenes de la web.

Para ello, hemos implementado en nuestro servicio de imágenes, el algoritmo por el cual las imágenes se transforman a este formato. Este servicio funciona en los servidores, y en cuanto se hace una petición al servicio de *cloud* y este no tiene la imagen guardada, se pone en contacto con el servidor, el cual crea la imagen correspondiente según el formato solicitado y se la devuelve para que la guarde con esta extensión.

Una vez implementado el algoritmo, observemos dos imágenes en su resolución natural, de los dos formatos que nos interesan, correspondientes a la Figura 40 y a la Figura 41.



*Figura 40. Imagen para móvil con WebP*



*Figura 41. Imagen JPG*

Podemos observar dos cosas importantes, tanto a nivel de resolución como tamaño son distintas. Teniendo una mayor resolución y calidad de imagen supondríamos que la Figura 40 ocupa mayor espacio de datos, pero en realidad si analizamos en detalle las propiedades de cada imagen, la Figura 40 ocupa 24,4KB mientras que la Figura 41 tiene un peso de 29,1kB.

Se confirma así que, para una misma imagen, usando estos dos formatos, la relación del tamaño entre el que tiene formato WebP con la misma imagen en JPG es de 0,7. Para poder realizar esta comparación es importante tener en cuenta que estas dos imágenes comparten un mismo SSIM [17] o índice de similitud estructural, que hace que se corresponda la calidad percibida de una misma imagen con distintos formatos.

Estamos implementando entonces imágenes de mejor resolución, con la misma calidad a la vez que podemos ahorrar una notable cantidad de datos.

De momento, este formato al ser relativamente nuevo no tiene soporte en todos los navegadores. Para ser precisos, únicamente van a poder ser usadas para las conexiones que vengan de Chrome, Safari o el navegador Android, ver Figura 42.

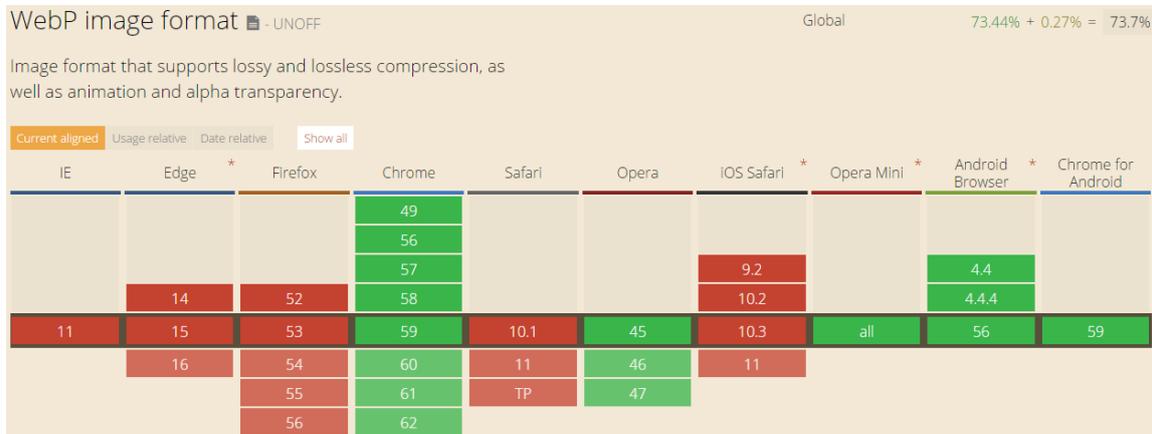


Figura 42. Compatibilidad del formato WebP con los diversos navegadores [18]

Por esta razón, esta implementación solo se ha realizado para dispositivos móviles, como fase de prueba del sistema y a la espera de la compatibilidad en las diversas plataformas.

A continuación, vamos a ver cómo podemos tratar esta situación, añadiendo utilidad de responsividad de la página en función del dispositivo.

### 6.3.6.2 Imagen responsiva

Al utilizar imágenes en una página web desde un dispositivo móvil, se suele cargar una imagen que es la misma para todos los dispositivos de diferentes tamaños y esta ser reescalada para amoldarse a las dimensiones del display de cada dispositivo. Estas imágenes reescaladas son normalmente de una calidad y resolución lo mayor posible para que en casos de dispositivos grandes se vean bien, por lo que se está cargando, en gran parte del tiempo, una cantidad de tamaño innecesario ya que estamos haciendo a continuación un *downgrade* en la resolución, por lo que nos sobra calidad de imagen. No se están optimizando las imágenes según las necesidades de cada usuario. Pero existe una manera de generar un sistema de imágenes responsivas “real”. La ventaja de este sistema es que vamos a conseguir hacer display de imágenes con menor tamaño siendo percibidas con la misma calidad que las anteriormente citadas, de gran tamaño. También evitamos la posibilidad de que exista

algún caso en el que el móvil sea grande y se tenga que reescalar la imagen hacia arriba, dejando una baja calidad.

Esto se hace con el tag <picture> y src, y se hace de la siguiente manera en la versión móvil de la página

```
<picture>
  <source srcset="http://img.europapress.net/fotoweb/fotonoticia_20170317125413_320.webp, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_480.webp 1.5x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_640.webp 2x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_960.webp 3x" type="image/webp" media="(max-width: 320px)">
  <source srcset="http://img.europapress.net/fotoweb/fotonoticia_20170317125413_320.jpg, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_480.jpg 1.5x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_640.jpg 2x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_960.jpg 3x" type="image/jpeg" media="(max-width: 320px)">
  <source srcset="http://img.europapress.net/fotoweb/fotonoticia_20170317125413_360.webp, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_540.webp 1.5x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_720.webp 2x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_1080.webp 3x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_1440.webp 4x" type="image/webp" media="(max-width: 360px)">
  <source srcset="http://img.europapress.net/fotoweb/fotonoticia_20170317125413_360.jpg, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_540.jpg 1.5x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_720.jpg 2x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_1080.jpg 3x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_1440.jpg 4x" type="image/jpeg" media="(max-width: 360px)">
  <source srcset="http://img.europapress.net/fotoweb/fotonoticia_20170317125413_375.jpg, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_750.jpg 2x" type="image/jpeg" media="(max-width: 375px)">
  <source srcset="http://img.europapress.net/fotoweb/fotonoticia_20170317125413_414.jpg, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_1242.jpg 3x" type="image/jpeg" media="(max-width: 414px)">
  <source srcset="http://img.europapress.net/fotoweb/fotonoticia_20170317125413_768.webp" type="image/webp" media="(max-width: 768px)">
  <source srcset="http://img.europapress.net/fotoweb/fotonoticia_20170317125413_768.jpg" type="image/jpeg" media="(max-width: 768px)">
  
</picture>
```

En este código podemos ver dos utilidades principales de este sistema. Como dijimos en el apartado anterior, estamos implementando imágenes con formato WebP en la web móvil.

En este *tag* `<picture>` observamos como cada elemento `<source>` está duplicado, teniendo una versión WebP y otra JPEG, siendo la versión WebP la que se encuentra primero, al priorizar su uso. Con esto estamos dando soporte a las imágenes según que navegador se esté usando, por lo que, si se está usando un navegador compatible, este cargará la primera opción, el formato WebP y de no ser ese el caso se cargará la imagen JPEG.

En cuanto a la responsividad, podemos observar como hay distintas *tag* `<source>` que implementan el atributo *media*, de la misma manera que se haría tradicionalmente en un archivo CSS en los diseños *responsive*, con las *media queries*. Por lo que damos distintas opciones según las dimensiones de los dispositivos. Además, no solo se quiere dar responsividad en función del tamaño, sino que se ha implementado distintas opciones según las resoluciones de los dispositivos. No se va a necesitar la misma imagen para dos dispositivos con el mismo tamaño que presenten distinto tipo de tecnología en sus *displays* [19]. Un ejemplo conocido es la tecnología Retina que está implementada en los *displays* de los dispositivos de la compañía Apple. Estos dispositivos tienen un mayor contenido de píxeles por pulgadas, o PPI, por lo que la calidad en un espacio reducido es mayor en este tipo de dispositivos, por ejemplo, la nitidez de los iconos pequeños se ve con una mayor calidad.

En `<source srcset="http://img.europapress.net/fotoweb/fotonoticia_20170317125413_320.jpg, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_480.jpg 1.5x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_640.jpg 2x, http://img.europapress.net/fotoweb/fotonoticia_20170317125413_960.jpg 3x" type="image/jpeg"` la imagen que termina con el 2x sería la que cargaría un dispositivo retina. Se está entonces dando distintos tipos de imágenes según la resolución de dispositivo.

Este sistema se implementa para cada imagen, y nos permite que la página sea realmente responsiva, además de evitar posibles cargas de imágenes de calidad más grande de lo que realmente se necesite.

## Capítulo 7. ANÁLISIS DE RESULTADOS

Una vez implementados estos cambios, es necesario hacer un estudio que cuantifique de que manera han afectado cada uno a la página web. Por lo tanto, en este capítulo vamos a analizar el sistema que nos encontramos tras terminar la implementación para poder decidir, para cada cambio, si han sido supuesto una mejora notable para los objetivos buscados. A partir de este análisis vamos a poder deducir en alguno de estos casos, si alguna implementación que no aporte una mejora notable necesite ser eliminada del proyecto.

### 7.1 RESULTADOS VERSIÓN DESKTOP

Primero vamos a observar cómo han afectado estos cambios a la versión *desktop* de la web.

Vamos a realizar un test con la herramienta WebPageTest con las mismas condiciones con las que se realizó el estudio de rendimiento, incluido en el Capítulo 5.

Este test se hace sobre la URL <http://www.europapress.es/> que se corresponde con la portada de la web. Obtenemos:

---

Métrica	Tiempo en s
Tiempo de primer byte	0,328 s
Comienzo de render	1,513 s
DOM cargado	5,63 s
Tiempo de carga	10,368 s

---

*Tabla 11. Resultados test nueva portada desktop*

Obtenemos un tiempo de carga total de aproximadamente 10 segundos. Si observamos el antiguo formato, Tabla 6, podemos observar como la mejoría en el tiempo de carga es de un 30%.

Además, podemos observar un gran cambio en la carga del DOM. Mientras que, en la antigua portada, se cargaba el DOM del primer plano de vista de la página en 6,9 segundos, dejando en el espacio de los *placeholders* de imágenes un gif que indicaba que se estaba procediendo a la carga de la imagen. Con este nuevo formato, el DOM se carga en los 5 primeros segundos, y encima las imágenes se cargan a su vez, Figura 43. Tenemos entonces que la zona de la ventana superior se carga de manera rápida y eficaz. Es una mejora del 19% sobre la velocidad anterior, la cual además se acompaña de una mejor experiencia de usuario, en cuanto que el contenido de imágenes se carga directamente en esta etapa.



*Figura 43. Fotograma estudio de rendimiento de la nueva portada de Desktop*

A la hora de la implementación de la nueva portada, uno de los puntos más importantes era como podríamos hacer que la web tuviera un mayor rendimiento cambiando la manera en la que se usa el contenido visual. Miremos entonces como se divide el contenido que se carga para esta nueva portada, Figura 44.

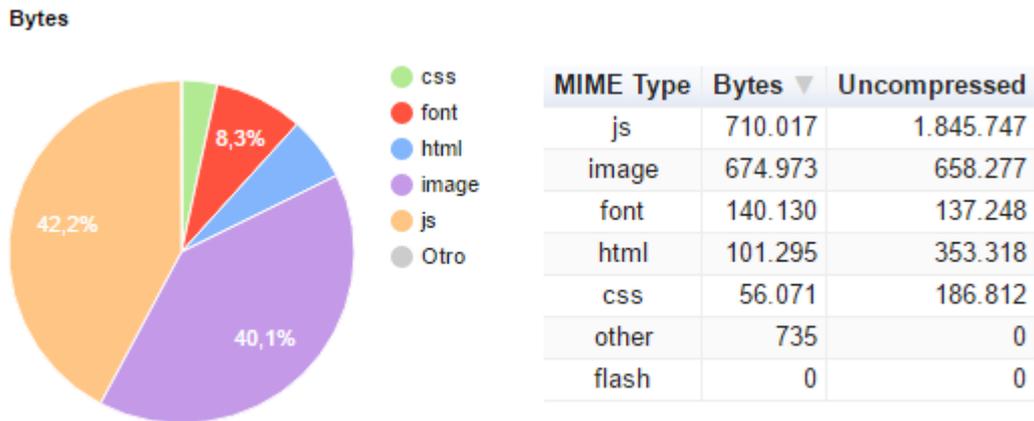


Figura 44. Análisis de contenido en la carga de la nueva portada de Desktop

La situación parece haber cambiado completamente en lo que concierne a la distribución de los elementos de la web. En la anterior versión, Figura 20, las imágenes suponían por sí solas el 60% del contenido de la página, siendo el contenido JS el segundo más importante con el 25%.

Ahora, existe un equilibrio entre los datos JS y los provenientes de las imágenes, suponen el 80% del contenido, 40% cada uno.

Se confirma entonces que este nuevo formato ha rebajado mucho la carga de imágenes. Este cambio es radical ya que ha supuesto una disminución de más de un 60% del tamaño que suponían las imágenes que se cargaban en la portada. Hemos reducido entonces a más de la mitad la cantidad de imágenes que se deben cargar para la portada de la web.

Esto también se ha visto afectado por la instauración del *lazy load* en las imágenes que están por debajo de la primera plana de la portada.

Al analizar el número de fotos que, en vez de cargar su contenido, introducen un gif hasta que no se rebasa una posición con el cursor, nos encontramos que existen un total de 38 imágenes que no se han cargado. A nivel de datos esto supone un ahorro de 894kB, lo que concuerda razonablemente con lo analizado anteriormente.

El *lazy load* ha supuesto el factor principal de que no se hayan cargado tantas imágenes como anteriormente y funciona de manera satisfactoria ya que no se llega a ver como se cargan estas imágenes, aunque se haga un *scroll* rápido.

Además, gracias a un desarrollo optimizado, se ha conseguido reducir la cantidad de datos CSS que se emplean en esta web, siendo esta cantidad reducida al 80%. Esta reducción tan drástica la atribuimos a que no se estaba gestionando en la antigua portada los estilos de manera correcta, ya que los diseñadores gráficos solo añadían nuevas reglas a la hoja de estilos de la antigua portada, sin tener en cuenta las que no estén empleando, y al datar este formato de mucho tiempo, existía una sobrecarga de información en los estilos que se cargaban.

Si observamos a nivel de los distintos sockets, ahora podemos ver como la mayoría de las conexiones corren por un canal seguro, Figura 45, donde el candado indica que hay una conexión mediante HTTPS

```
2. s03europapress...7296256_r2314.css
3. s03europapress...3991349249_925.js
4. www.googletagservices.com - gpt.js
5. ajax.googleapis.com - jquery.min.js
6. securepubads.g...ubads_impl_121.js
7. noddus.com - card_loader.js
8. s03europapress...2147483648_925.js
9. ajax.googleapis.com - webfont.js
10. cdn.elasticad.n... nativeEmbed.gz.js
11. s01europapress...net - Altadis.png
12. imgeuropapress...004_126_bgeee.jpg
13. imgeuropapress...3_105_105_0_0.jpg
14. imgeuropapress...illo_99x99_v2.jpg
```

Figura 45. Conexiones en la nueva portada desktop

Al realizar esta implementación de imágenes que corren sobre un canal seguro, conseguimos que se use el protocolo HTTP/2 para la carga de imágenes, como se puede ver en la Figura 46, con el servicio del CDN

**Request #24** Details Request Response Object

**URL:** https://s01europapress.akamaized.net/imagenes/desayunos/patrocinadores/kpmg.png  
**Loaded By:** http://www.europapress.es/:1001  
**Host:** s01europapress.akamaized.net  
**IP:** 23.67.250.139  
**Error/Status Code:** 200  
**Priority:** Low  
**Protocol:** HTTP/2  
**HTTP/2 Stream:** 9, weight 147, depends on 7, EXCLUSIVE  
**Client Port:** 50779  
**Request Start:** 1.316 s  
**Time to First Byte:** 8 ms  
**Content Download:** 0 ms  
**Bytes In (downloaded):** 3.0 KB  
**Uncompressed Size:** 2.7 KB  
**Bytes Out (uploaded):** 51 B

Figura 46. Request HTTP/2

Hemos visto pues como ha mejorado la velocidad de carga de este formato de manera sustancial. Finalmente, vamos a utilizar la herramienta de análisis de rendimiento web de Google, para ver si los cambios realizados han supuesto una mejor valoración y si siguen los estándares que se deben de seguir según esta empresa para tener una web competitiva.

PageSpeed Tools > Insights

GUÍAS REFERENCIA MUESTRAS ASISTENCIA

Móvil Ordenador

**Needs Work**  
73 / 100

A esta página le faltan algunas optimizaciones de rendimiento comunes que pueden resultar en una experiencia de usuario lenta. Consulta las recomendaciones que aparecen a continuación.

**Optimizaciones posibles**



Figura 47. Análisis WebPage Tools Insights sobre la nueva portada desktop

Podemos ver en la Figura 47, como ya no estamos en niveles críticos según esta herramienta. Hemos pasado de una puntuación de 61/100 a 73/100, pasando de un nivel malo a un estado en el cual esta herramienta sugiere tan solo algunos cambios, lo cual es un buen índice ya que estamos trabajando con una web de noticias con mucho contenido.

Podemos considerar el cambio realizado en esta portada como muy positivo, cumpliéndose los objetivos que buscábamos con estos cambios.

## **7.2 RESULTADOS VERSIÓN MÓVIL**

Ahora observemos cuales han sido los resultados de los cambios introducidos en la versión móvil de la web.

Como hicimos un estudio previo sobre una noticia de última hora, en este caso vamos a analizar otra noticia de última hora para estar en unas condiciones lo más similares posibles.

Vamos a analizar la noticia <http://m.europapress.es/economia/energia-medio-ambiente-00183/noticia-iberdrola-suscribe-convenio-cepyme-asesorar-eficiencia-energetica-pymes-autonomos-20170615130620.html>, obtenemos:

---

<b>Métrica</b>	<b>Tiempo en s</b>
<b>Tiempo de primer byte</b>	<b>0.365 s</b>
<b>Comienzo de render</b>	<b>2.336 s</b>
<b>DOM cargado</b>	<b>5.734 s</b>
<b>Tiempo de carga</b>	<b>7.651 s</b>

---

*Tabla 12. Datos rendimiento de carga noticia móvil antigua*

Como en el caso del *desktop*, se observa una mejora sustancial de la velocidad de carga. Pasamos de un tiempo de carga de la noticia de 10 segundos a 7 y medio, lo que supone una mejora del 25%.

Además, el tiempo de carga de los elementos visuales principales pasa de 7,8 a 5,7 segundos, otra vez una mejora del 25%.



Figura 48. Fotograma estudio de rendimiento noticia móvil nueva

La implementación del CSS *inline* en la práctica resulta efectiva ya que, si se estudia la manera de carga de la web, parece que la totalidad de los estilos están cargados mientras que en realidad no se ha cargado la hoja de estilos que contiene los estilos que no se ven en el primer punto de vista de la Web. Por el contrario, antes se podía observar como tardaba más en cargarse el contenido del DOM ya que era necesario cargar la hoja de CSS antes que el contenido de la página ya que este contenido se implementaba en la cabecera de la página HTML. El efecto de sensación de carga es efectivo, ya que no se aprecia que falte contenido

al cual haga falta aplicar un estilo y se obtiene antes la primera parte de contenido de la página.

Además, podemos observar como al haber hecho el test empleando como parámetro de navegador Google Chrome, se está cargando la imagen en formato WebP como explicamos en el último apartado del capítulo anterior. Aparte, también estamos beneficiándonos de la implementación del CDN Akamai que hemos visto para *desktop* teniendo distintos recursos transmitidos mediante HTTPS + HTTP/2

En cuanto al formato móvil, la implementación por la cual se espera unos grandes resultados es la del formato de noticia AMP. Para realizar este estudio vamos a utilizar la misma noticia que hemos empleado anteriormente para la nueva versión móvil en su formato AMP.

Primero vamos a utilizar el *Web Page Insight* de Google para obtener una puntuación de rendimiento de esta página.

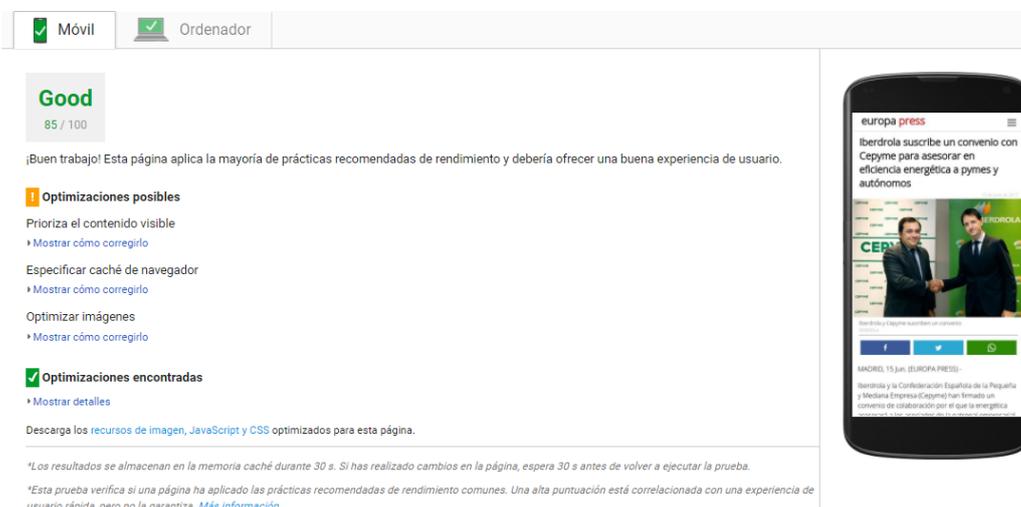


Figura 49. Análisis WebPage Tools Insights sobre la nueva portada desktop

En la Figura 49 podemos observar cómo hemos obtenido un resultado muy positivo. Como era de esperar, Google detecta este formato dentro sus cánones como un formato que cumple

la mayoría de directrices que proponen. Esto se debe a que al implementar AMP, hemos tenido que seguir una serie de requisitos muy estrictos, que no dan margen para implementar sistemas muy complejos. Estamos trabajando única y solamente con lo que Google nos permite desarrollar. Este *framework* es muy cerrado por lo que, para tener una página operativa, se debe seguir lo que nos dice Google, solamente tenemos sus herramientas y si no se hace así, vamos a incurrir en tener páginas plagadas de errores e incompatibilidades.

Veamos entonces cual es la diferencia que nos vamos a encontrar si comparamos las dos versiones para móvil de esta noticia, ver Figura 50



Figura 50. Resultado de la demo comparativa del tiempo de carga de las dos versiones de las noticias para móvil (AMP contra m.)

Podemos observar como entre estas dos versiones de la noticia existe una gran diferencia de velocidad. La versión AMP gana ampliamente en este apartado a la versión normal de móvil, siendo esta aproximadamente el doble de rápida.

<b>Tiempos en segundos</b>	<b>Versión AMP</b>	<b>Versión móvil</b>
<b>Tiempo de “first byte”</b>	<b>0,408 s</b>	<b>0,365 s</b>
<b>Comienzo de renderización</b>	<b>1,826 s</b>	<b>2,336 s</b>
<b>Elementos visuales completados</b>	<b>2,436 s</b>	<b>5,734 s</b>
<b>CARGA TOTAL</b>	<b>5,545 s</b>	<b>7,651 s</b>

*Tabla 13. Comparativa de tiempos de carga entre AMP y móvil*

La Tabla 13 nos permite apreciar la eficacia del *framework* AMP para cargar los datos rápido. Aunque la página llegue a tomar 5 segundos en cargarse por completo, el usuario ya tiene los elementos visuales cargados en torno a los dos segundos, lo que es un resultado lo suficientemente rápido para que el usuario no tenga la sensación de estar esperando para obtener el contenido. Esto crea una experiencia de usuario muy positiva.

Uno de los resultados que caben destacar, es que dentro de los 5 segundos que tarda en cargar la noticia, el elemento que provoca que no se considere que la página esté cargada por completo a los 2 segundos, tiempo que tarda en cargarse gran parte del contenido visual, es el contenido patrocinado. Como explicamos en la Figura 39, uno de los defectos principales de este sistema es cómo funciona la carga de anuncios, siendo estos muchos más lentos que el resto del contenido. Lo que puede crear situaciones donde se vean huecos vacíos donde se esté cargando los anuncios. Por esto hemos introducido estos anuncios dentro del contenido de la noticia, Figura 39, para evitar que el usuario experimente esta situación, como ocurre en otras webs, como se aprecia en la Figura 51. Esta limitación ha supuesto que por ejemplo

se haya decidido no introducir anuncios en el la cabeza de página como se suele hacer con frecuencia en el resto de la página



*Figura 51. Ejemplo de un anuncio amp mal empleado*

Ante este problema, solo nos queda estar atentos a las próximas actualizaciones que introduzca este nuevo *framework*, por el cual Google está realizando una apuesta importante como demuestra la primera conferencia AMP que se desarrolló en marzo de este año y donde se habló de como se está trabajando para mejorar estas limitaciones.

Esta apuesta también se demuestra en uno de los resultados más interesantes que nos ha dejado el despliegue de este tipo de noticias en nuestra web. Actualmente, al tener un contenido AMP validado y en funcionamiento, estas noticias están siendo indexadas por el motor de búsqueda de Google, como demuestra la Figura 52

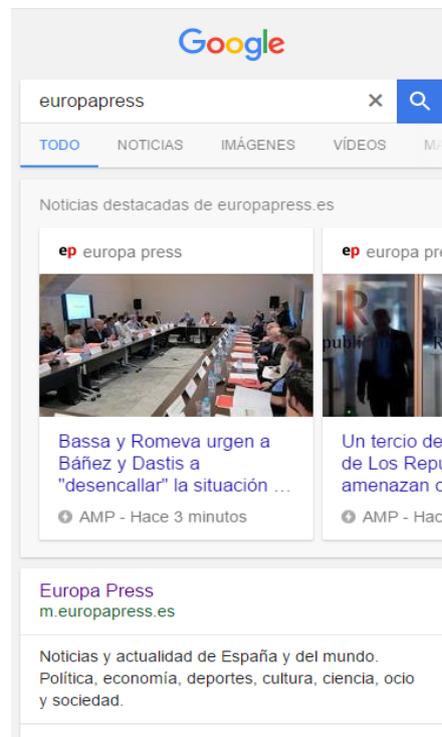


Figura 52. Noticias AMP de europapress indexadas en Google

Esto supone dos situaciones muy positivas para la web. La primera es en el plano SEO, ya que ser al indexadas estas noticias como noticias destacadas, se está consiguiendo gran visibilidad para la web y se puede ganar mucho tráfico por referencia de Google.

Es importante tener en cuenta que, a la hora de conseguir tráfico en nuestra página, el motor de búsqueda de Google es el parámetro más importante para ello. Gracias a la herramienta Google Analytics, podemos observar desde donde nos llega el tráfico. Observemos entonces cuales han sido las fuentes desde las cuales los usuarios se han conectado a la página.

Fuente/medio	Sesiones	Rebotes
	15.280.996 % del total: 100,00 % (15.280.996)	11.295.255 % del total: 100,00 % (11.295.255)
1. ■ google / organic	6.382.579	40,15 %
2. ■ m.facebook.com / referral	2.685.533	19,78 %
3. ■ (direct) / (none)	2.169.710	13,06 %
4. ■ Facebook / Social Instant Article	1.387.493	11,12 %
5. ■ t.co / referral	661.082	4,43 %
6. ■ flipboard.com / referral	451.945	3,38 %
7. ■ facebook.com / referral	230.917	1,22 %
8. ■ twitter.com / social	165.744	1,05 %
9. ■ google / cpc	134.657	0,58 %
10. ■ europapress.es / referral	126.924	0,42 %

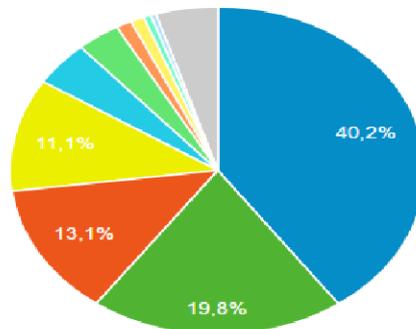


Figura 53. Análisis de las fuentes de tráfico

En la Figura 53, podemos observar como la mayoría de las sesiones que se crean en nuestra web proceden de referencias de Google. El 40% de nuestras visitas dependen de la indexación de Google, por lo que es realmente crítico tener en cuenta las propuestas de Google y seguir sus directrices.

Además de la importancia de ganar tráfico, con la implementación de AMP nos podemos beneficiar de la AMP Caché, ver Figura 54.



Figura 54. Galería de noticias AMP cacheadas por Google

Con esta caché estamos consiguiendo una experiencia de usuario que crea la sensación de estar en una aplicación nativa. A la hora de navegar dentro de esta galería, el usuario no puede en ningún momento apreciar la carga de las noticias. En esta figura, lo que está ocurriendo es que el hosting de las noticias se produce por parte de Google. Para conseguir que el usuario no aprecie la carga aprovecha que están realizando ellos mismo el hosting, para poder realizar un *prerendering* de estas noticias.

Por lo que, en este caso, lo que ha ocurrido es que a la hora de buscar europapress en el buscador de Google, a la vez que se ha realizado la búsqueda, se ha forzado la carga de las noticias. Por este mecanismo, se crea una herramienta por la cual podemos leer las principales noticias de nuestra web que Google decida hacer de host.

Este resultado nos viene bien por el momento, ya que como no tenemos una portada implementadas en AMP, este mecanismo actúa de sustituto ante esta situación. Además, al tener la mayoría del tráfico procedente de Google, su funcionalidad está garantizada.

### **7.3 RECOPIACIÓN DE TOTALIDAD DE RESULTADOS**

Para tener acceso a la globalidad de los resultados del proyecto resumidos en un mismo sitio, creamos la tabla a continuación:

	<b>Versión móvil</b>	<b>Versión móvil AMP (lo comparamos con la versión móvil estándar antigua)</b>	<b>Versión DESKTOP</b>
<b>Tiempo carga global versión antigua</b>	<b>10,368 s</b>	<b>10,368 s</b>	<b>14,152 s</b>
<b>Tiempo carga global versión versión mejorada</b>	<b>7,651 s</b>	<b>5,545 s</b>	<b>10,368</b>
<b>Mejora</b>	<b>25%</b>	<b>46%</b>	<b>27%</b>
<b>MEJORA MEDIA EN LA WEB</b>		<b>33%</b>	

*Tabla 14. Resumen global de resultados de carga total*

## **Capítulo 8. CONCLUSIONES Y TRABAJOS FUTUROS**

### **8.1 CONCLUSIONES**

Este proyecto nació a partir de una necesidad empresarial en la que se buscaba mejorar una situación en la cual existía una falta de optimización. El objetivo era el de realizar una serie de modificaciones que se tradujeran en una mejora del rendimiento de carga de la web de Europa Pres. Por este motivo, la razón principal de este trabajo siempre fue la de obtener resultados observables, más allá de conseguir hacer un análisis de la situación en la que se encontraba y de que manera se podía mejorar. Todo el análisis inicial y las alternativas propuestas para conseguir mejoras no servirían de nada si no se tradujera en resultados.

Como se demuestra en el Capítulo 7. , se ha llegado a la consecución de los objetivos del proyecto. A partir de todo lo desarrollado en el proyecto, la página de Europa Press se ha convertido en una web que tiene una mejor velocidad de descarga respecto al inicio de este trabajo. Esto ha llevado consigo la adaptación a las últimas tecnologías web, que buscan la optimización de los recursos y la manera en la que trabaja el mecanismo de carga de una página web.

Pero no solamente ha supuesto una mejora en su rendimiento, ya que, al implementar estos cambios, se han optimizado otros apartados que son importantes para cualquier plataforma de servicio *online*. Estos son principalmente dos aspectos, e SEO y la seguridad. A la hora de crear los nuevos formatos, se ha tenido que realizar bastante trabajo para respetar las condiciones que se requieren para conseguir un buen posicionamiento. Además, con la introducción del formato AMP se ha visto como se está consiguiendo mejoras de exposición por parte de Google, para atraer mayor tráfico desde dispositivos móviles. En el marco de la seguridad, al estar buscando el uso del protocolo HTTP/2, ha sido necesario la introducción de comunicaciones encriptadas. El objetivo final, será tener la web corriendo completamente mediante el binomio HTTP/2 con HTTPS, lo que supone una mejora en la manera de tratar las conexiones.

Por esto, este proyecto ha cumplido su objetivo de conseguir la reforma de la web mediante los cambios que han sido necesarios para que todo cargue más rápido.

Pero, dentro de este entorno de trabajo, estos cambios no significan que se haya logrado el mejor caso posible. Dentro de las limitaciones del trabajo, nos encontramos con que hemos tenido disponible un periodo de tiempo limitado, que no nos ha permitido llegar a una optimización completa. Existen algunas alternativas que vamos a presentar en el siguiente apartado de futuras mejoras, que deja una hoja de ruta posible, en el caso de que se quiera seguir con la mejoría de la velocidad de carga de la web. Además, al estar en un entorno de trabajo de grandes dimensiones, en muchos casos para realizar cambios y a la hora de presentar alternativas, los procesos se producen de una manera lenta para conseguir ser aprobados y siempre hay que tener en cuenta al resto del sistema para poder ver la posibilidad de adoptar estos cambios.

## **8.2 TRABAJOS FUTUROS**

Como punto final de este proyecto, vamos a desarrollar una propuesta sobre cuáles son los cambios que puedan suponer una continuación sobre lo que aquí se ha desarrollado, que supongan una mejoría, aprovechando las lecciones y beneficios obtenidos. Proponemos las siguientes opciones:

Hemos podido observar como adoptando el formato AMP se han conseguido unos rendimientos de velocidad y de SEO bastante importantes. Es por eso que podría llegar a ser interesante plantearse crear una versión AMP de la totalidad de la web, y no solamente de las noticias. Esto supondría un gran trabajo por el tema de limitación de JS, pero el beneficio que se puede obtener a partir de esto es lo suficientemente grande como para considerar la implementación de este sistema.

En cuanto al protocolo HTTP/2, vimos como existían alguna web de noticias que ya lo tenían todo adaptado para correr mediante este protocolo. Como dijimos al inicio del proyecto, esto es una transición lenta que lleva un tiempo consigo, como el caso citado

anteriormente del NY Times. Pero este trabajo supondría poder tener HTTPS en la web y beneficiarse de HTTP/2, para obtener tiempos de carga muy cortos, del orden de los que vimos en Tabla 8, con el ejemplo del Washington Post.

Finalmente, uno de los aspectos que más han llamado la atención en este proyecto es como se está gestionando las hojas de estilo dentro de la web. En la actualidad se tienen situaciones donde al cargar las hojas de estilos, la amplia mayoría de los estilos no llegan a ser empleados.

Esto se debe a que se carga una hoja de estilos principal, a la cual se añaden casos especiales como podemos ver en el código siguiente, en la función de carga de CSS:

```
Select Case portalActual
    Case Portal.Portales.Chance : ArchivosCSS = CSSExternos.Chance
    Case Portal.Portales.Turismo : ArchivosCSS = CSSExternos.Turismo
    Case Portal.Portales.Portaltic : ArchivosCSS = CSSExternos.PortaltIC
    Case Portal.Portales.Culturaocio : ArchivosCSS = CSSExternos.CulturaO
cio
    Case Portal.Portales.Hacerfamilia : ArchivosCSS = CSSExternos.HacerFa
milia
    Case Portal.Portales.Desconecta : ArchivosCSS = CSSExternos.Desconect
a
    Case Portal.Portales.Cienciaplus : ArchivosCSS = CSSExternos.CienciaP
lus
    Case Portal.Portales.CampusVivo : ArchivosCSS = CSSExternos.CampusViv
o
    Case Portal.Portales.AldiaCat : ArchivosCSS = CSSExternos.Aldia
    Case Portal.Portales.Notimerica : ArchivosCSS = CSSExternos.Latam
    Case Portal.Portales.EsAndalucia : ArchivosCSS = CSSExternos.EsAndalu
cia
    Case Else : ArchivosCSS = CSSExternos.General
End Select
```

Lo que se propone, sería el desarrollo de un sistema de hojas de estilos que, en vez de ser gestionado de la manera anterior, se haga mediante el *framework* LESS [20]. Mediante este *framework* se podrían crear hojas de estilo para cada funcionalidad, las cuales podrían luego ser importadas en el archivo .less de cada caso. Una de las características de LESS es la posibilidad de importar otros archivos LESS. A continuación, vemos un ejemplo donde al

inicio del archivo LESS se le dice que coja los estilos del *body* de la web, los estilos del menú de navegación, las variables que nos dan distintos colores usados en toda la web y el estilo del buscador:

```
@import "Body.less";  
@import "_navigation.less";  
@import "_variables.less";  
@import "buscador-principal.less";
```

Esto sería como generar CSS de manera modular, y permitiría tener un sistema que genere hojas de estilo más organizado para evitar encontrarnos casos donde se están cargando estilos sin necesidad.

## Capítulo 9. BIBLIOGRAFÍA

- [1] Gomez, “Why Web Performance Matters: Is Your Site Driving Customers Away?” (2010)
- [2] DoubleClick by Google, “The Need For Mobile Speed” (2016)
- [3] International Telecommunication Union (ITU), “ICT Facts and Figures 2016” (2015)
- [4] Internet Live Stats, “Estadísticas de usuarios de internet en el mundo según la ITU”, Hipervínculo (2017)
- [5] Deloitte, “The impact of web traffic on revenues of traditional newspaper publishers” (2016)
- [6] PwC, “Transitioning from a print past to a digital future – triggering new strategies and wide divergences between markets” (2016)
- [7] Webpagetest Documentation, Hipervínculo
- [8] Akamai, “State of the Internet Q3 2016 report” (2016)
- [9] Mike Belshe, “More Bandwith doesnt matter (much)” (2010)
- [10] Ilya Grigorik, “High Performance Browser Networking” - O'Reilly (2013)
- [11] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper”, (2016)
- [12] Google, “Google AMP Cache”, Hipervínculo (2017)
- [13] Microsoft, “ASP.NET Master Pages”, Hipervínculo (2017)

- [14] Steve Souders, “Even Faster Websites” – O’Reilly (2009)
- [15] Google, “AMP Components/Tags”, [Hipervínculo](#)
- [16] Google, “WebP: A new image format for the Web”, [Hipervínculo](#) (2016)
- [17] IEEE , “Image Quality Assessment: From Error Visibility to Structural Similarity” (2004)
- [18] Caniuse.com, “Compatibilidad del formato WebP con los distintos navegadores”, [Hipervínculo](#)
- [19] Sebastian Gabriel ,“Guía PPI” [Hipervínculo](#)
- [20] Bass Jobsen, “Less Web Development Essential” (2014)
- [21] Zona Research, “The economic Impacts of Unnacceptable Web Site Download Speeds” (1999)
- [22] Fiona Fui-Hoon Nah,” A study on tolerable waiting time: how long are Web users willing to wait?” College of Business Administration-University of Nebraska - Lincoln (2004)
- [23] Discussion Paper- Harvard Kennedy School, “What Newspapers Don’t Know about Web Traffic Has Hurt Them Badly – But There is a Better Way” (2015)
- [24] IETF, Hypertext Transfer Protocol Version 2 (HTTP/2), [Hipervínculo](#)(2015)
- [25] Zhou Munyaradzi<sup>1</sup>, Giyane Maxmillan<sup>2</sup>, Mutembedza Nyasha Amanda, “Effects of Web Page Contents on Load Time over the Internet.” Midlands State University, Faculty of Science and Technology
- [26] Varun Sapra and Abram Hindle, “Web Servers Energy Efficiency Under HTTP/2” - PeerJ (2016)

- [27] Imperva, “HTTP/2: In-depth analysis of the top four flaws of the next generation web protocol” (2016)
- [28] Nginx, “HTTP/2 for Web Application Developers” - (2015)
- [29] Bo Han, Shuai Hao, Feng Qian, “MetaPush: Cellular-Friendly Server Push For HTTP/2” (2015)
- [30] Simic, Bojan, “The Performance of Web Applications: Customers Are Won or Lost in One Second” November 2008
- [31] Equation Research on behalf of Gomez, Inc. “Why the Mobile Web is Disappointing End-Users. A study of consumers’ mobile web experiences” (2009)
- [32] IETF, “Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing”, Hipervínculo (2014)
- [33] IETF, “HPACK: Header Compression for HTTP/2”, Hipervínculo (2104)
- [34] IETF, “HTTP Over TLS”, Hipervínculo (2000)
- [35] Darpa internet program, “Transmission control protocol”, Hipervínculo (1981)
- [36] IETF, “Transport Layer Security (TLS): Application-Layer Protocol Negotiation Extension”, Hipervínculo (2014)
- [37] W3C, “A vocabulary and associated APIs for HTML and XHTML” Hipervínculo (2014)
- [38] IETF, “TCP Extensions for High Performance”, Hipervínculo (2014)
- [39] Internet protocol darpa internet program protocol specification Hipervínculo (1981)

- [40] HTTP/1 Semantics and Content , Hipervínculo
  
- [41] HTTP/1 Caching, Hipervínculo
  
- [42] Steven Seow, “Designing and Engineering Time: The Psychology of Time Perception in Software” (2008)
  
- [43] O'Reilly & Associates, Inc., “HTTP the definitive guide” – O'Reilly (2002)
  
- [44] Steve Souders, “High performance web sites” - O'Reilly (2007)
  
- [45] Robert B. Miller, “Response time in man-computer conversational transactions” - Fall Joint Computer conference (1968)
  
- [46] Clement Nedelcu, “Nginx HTTP Server Second Edition” – Packt publishing (2013)
  
- [47] Karl Seguin, “The little Redis Book” (2012)
  
- [48] Josiah L. Carlson, “Redis in action”- Manning (2013)
  
- [49] Tiago Macebo & Fred Oliveira, “Redis Cookbook” - O'Reilly (2011)
  
- [50] Dimitri Aivaliotis, “Mastering Nginx” - Packt publishing (2013)
  
- [51] Nginx Inc., “Nginx Modules reference” (2014)
  
- [52] Nginx Inc., “101 Nginx Secret Heart”
  
- [53] Internet Society, “Brief History of the Internet”, Hipervínculo
  
- [54] WAN-IFRA, “World press trends” (2015)

- [55] Deloitte, “The impact of web traffic on revenues of traditional newspaper publishers” (2016)
  
- [56] Google, “WebP: A new image format for the Web”, Hipervínculo (2016)
  
- [57] Google, “WebP Compression Study”, Hipervínculo (2016)
  
- [58] Al-Mukaddim Khan Pathan and Rajkumar Buyya, “A Taxonomy and Survey of Content Delivery Networks”, University of Melbourne (2008)
  
- [59] Guía uso del tag picture, Hipervínculo

## ANEXO A

Código de MP\_Portada.master, hoja master implementada en todas las portadas de los distintos canales

```
<%@ Master Language="VB" Inherits="MP_Portada" CodeBehind="MP_Portada.master.vb"
%>

<%@ Register TagPrefix="webep" TagName="ModuloHTML" Src="modulos/ModuloHTML.ascx"
%>
<%@ Register TagPrefix="webep" TagName="MenuPrincipalSeccion" Src="modulos/MenuPr
incipal_Seccion.ascx" %>
<%@ Register TagPrefix="webep" TagName="Stats" Src="~/modulos/TagEstadisticas.asc
x" %>
<%@ Register TagPrefix="webep" TagName="Copyright" Src="~/modulos/Copyright.ascx"
%>
<%--
<%@ Register TagPrefix="webep" TagName="GAM_Head" Src="~/modulos/GoogleAdManager_
Head.ascx" %>--%>
<%@ Register TagPrefix="webep" TagName="GTM_Head" Src="~/modulos/GoogleTagManager
_Head.ascx" %>
<%@ Register TagPrefix="hacerfamilia" TagName="GAM_Head_HacerFamilia" Src="~/hace
rfamilia/Modulos/GoogleAdManager_Head_HacerFamilia.ascx" %>
<%@ Register TagPrefix="webep" TagName="Scripts" Src="~/modulos/Head_Scripts.ascx
" %>
<%@ Register TagPrefix="webep" TagName="CSS" Src="~/modulos/Head_CSS.ascx" %>
<%@ Register TagPrefix="webep" TagName="OcultarAdmin" Src="~/Modulos/OcultarAdmin
.ascx" %>
<%@ Register TagPrefix="webep" TagName="Pie_2014" Src="~/Modulos/Pie_2014.ascx" %
>
<%@ Register TagPrefix="webep" TagName="SigueRRSS" Src="~/Modulos/SigueRRSS.ascx"
%>
<%@ Register TagPrefix="webep" TagName="NavegacionFixed" Src="~/modulos/Navegacio
nFixed/NavegacionFixed.ascx" %>
<%@ Reference Control="~/modulos/Banner.ascx" %>
<%@ Reference Control="~/modulos/MenuSecciones.ascx" %>
<%@ Register Src="~/Modulos/MenuSeccionesAdmin.ascx" TagPrefix="webep" TagName="M
enuSeccionesAdmin" %>
<%@ Register TagPrefix="webep" TagName="BannerGTM" Src="~/modulos/BannerGTM.ascx"
%>
<!DOCTYPE html>
<html lang="<%= _Lang %>" xml:lang="<%= _Lang %>" xmlns="http://www.w3.org/1999/xhtml
" xmlns:og="http://opengraphprotocol.org/schema/" xmlns:fb="http://www.facebook.c
om/2008/fbml" prefix="og: http://ogp.me/ns#">
<head runat="server">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta http-equiv="x-dns-prefetch-control" content="on" />
    <asp:Placeholder ID="facebookPageMetas" runat="server" />
```

```

<asp:Placeholder ID="DNSPrefetch" runat="server"></asp:Placeholder>
<link rel="dns-prefetch" href="//fonts.googleapis.com/" />
<link rel="dns-prefetch" href="//ajax.googleapis.com/" />
<link rel="dns-prefetch" href="//google-analytics.com/" />
<link rel="dns-prefetch" href="//b.scorecardresearch.com/" />
<link rel="dns-prefetch" href="//p.jwpcdn.com/" />

<script type="text/javascript">
    WebFontConfig = {
        google: {
            families: ['Open+Sans:400,600,600italic,700']
        }
    };

    (function (d) {
        var wf = d.createElement('script'), s = d.scripts[0];
        wf.src = 'https://ajax.googleapis.com/ajax/libs/webfont/1.6.16/webfont.js';
        s.parentNode.insertBefore(wf, s);
    })(document);
</script>
<script type="text/javascript">var _sf_startpt = (new Date()).getTime()</script>

<!--
- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -
-->
<!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
<!--[if lt IE 9]>
    <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
</[endif]>

<!-- Facebook Pixel Code -->
<script>
    !function (f, b, e, v, n, t, s) {
        if (f.fbq) return; n = f.fbq = function () {
            n.callMethod ?
                n.callMethod.apply(n, arguments) : n.queue.push(arguments)
        }; if (!f._fbq) f._fbq = n;
        n.push = n; n.loaded = !0; n.version = '2.0'; n.queue = []; t = b.createElement(e); t.async = !0;
        t.src = v; s = b.getElementsByTagName(e)[0]; s.parentNode.insertBefore(t, s)
    }(window,
        document, 'script', 'https://connect.facebook.net/en_US/fbevents.js')
;

    fbq('init', '1748569405419693');
    fbq('track', "PageView");</script>
<noscript></noscript>
  <!-- End Facebook Pixel Code -->
  <script>
    (function () {
      var dht = document.createElement('script');
      dht.type = 'text/javascript';
      dht.async = true;
      dht.src = ('https:' == document.location.protocol ? 'https://' : 'http://') + '1tol.bbva.com/1tol/dht.js';
      var s = document.getElementsByTagName('script')[0];
      s.parentNode.insertBefore(dht, s);
    })();
  </script>

  <asp:Placeholder ID="StartHeader" runat="server" />
  <asp:Placeholder ID="PHFavicon" runat="server" />
  <meta http-equiv="Content-Language" content="es" />
  <meta name="country" content="Spain" />
  <meta name="author" content="Europa Press" />
  <meta name="organization" content="Europa Press S.A." />
  <meta name="copyright" content="Europa Press" />
  <meta name="locality" content="Madrid, España" />
  <meta name="distribution" content="global" />
  <asp:Placeholder ID="LinkRelIndex" runat="server"></asp:Placeholder>
  <link href="https://plus.google.com/101131134978819552698" rel="publisher" />
  <webep:CSS runat="server" ID="CSS2" />
  <!--
<webep:GAM_Head ID="GoogleAdManager" runat="server" SlotsIniciales="home136_columna2,home627_columna2" />-->
  <webep:GTM_Head ID="GoogleTagManager" runat="server" ModoSingleRequest="false" />
  <hacerfamilia:GAM_Head_HacerFamilia ID="GoogleAdManagerHacerFamilia" runat="server" Visible="false" />
  <asp:ContentPlaceHolder ID="Header" runat="server" />
  <webep:Stats ID="TagGoogle" runat="server" Provider="GoogleUniversal" PorcentajeEncuestas="0" />
  <!-- <meta name="viewport" content="width=device-width,minimum-scale=1,initial-scale=1" />-->
  <meta name="viewport" content="width=1080" />
  <webep:Scripts runat="server" ID="ScriptsHome" Visible="false" Indent=" " Modulos="jQuery_v2, jQuery_NavegacionFixed, JS_Portada, navegacion" />
</head>
<body class="CuerpoResponsive">
  <webep:Stats ID="Comscore" runat="server" Provider="Comscore" PorcentajeEncuestas="0" />
  <asp:Placeholder ID="SchemaOrganizationYPerson" runat="server"></asp:Placeholder>
  <asp:ContentPlaceHolder ID="RRSS" runat="server" />
  <webep:NavegacionFixed ID="NavegacionFixed" runat="server" />
  <asp:ContentPlaceHolder ID="ContentMenuHome" runat="server" />
  <form id="form1" runat="server">

```

```

<div id="PublicidadSuperior" class="Canvas" runat="server">
  <div id="BannerCabecera" class="publi">
    <asp:ContentPlaceHolder ID="BigBanner" runat="server" />
  </div>
</div>
<webep:MenuPrincipalSeccion ID="Menu" runat="server" Visible="false" />
<div class="full-page-width-noticia">

  <div class="container center-content-container">
    <asp:ContentPlaceHolder ID="adminMetaData" runat="server" />
    <webep:OcultarAdmin ID="ocultaradmin" runat="server" />
    <webep:MenuSeccionesAdmin runat="server" ID="MenuSeccionesAdmin"
  />

    <asp:ContentPlaceHolder ID="ContenidoSuperior" runat="server" />
    <asp:ContentPlaceHolder ID="ContenidoSuperiorBanner_980" runat="s
server" />

    <asp:ContentPlaceHolder ID="ContenidoSobreColumnaCentral" runat="
server" />

    <%-- Inicio Home --%>
    <asp:ContentPlaceHolder ID="ContenidoPrimeraPlana" runat="server"
  />

    <asp:ContentPlaceHolder ID="ContenidoCentralHome" runat="server"
  />

    <asp:ContentPlaceHolder ID="ContenidoInferior" runat="server" />
    <%-- Fin Home --%>
    <%-- Inicio Portadas --%>
    <div class="row flex">
      <asp:ContentPlaceHolder ID="CPH_ColumnaIZQ" runat="server" />
      <asp:ContentPlaceHolder ID="CPH_ColumnaCentral" runat="server
" />

    </div>
    <div class="row">
      <asp:ContentPlaceHolder ID="ContenidoBajoNoticia" runat="serv
er" />

    </div>
    <%-- Fin Portadas --%>
  </div>
</div>

<footer id="footer" role="contentinfo" class="footer<%= _ClaseFooter %>">
  <div class="row">
    <div class="twelve columns">
      <webep:Pie_2014 ID="Pie_2014" runat="server" />
      <webep:Copyright ID="Copyright1" runat="server" />
    </div>
  </div>
</footer>
<webep:Scripts runat="server" ID="Scripts" Indent=" " Modulos="navegac
ion, jQuery_LazyLoad, jQuery_NavegacionFixed, JS_Portada " />
<webep:Scripts runat="server" ID="ScriptsHome2" Visible="False" Modulos="
jQuery_Royal_Slider" />

```

```

<!--<webep:Scripts runat="server" ID="ScriptsHome3" Visible="False" />--
<webep:Scripts runat="server" ID="Scripts2" />

<div id="lazyloadDiv" runat="server" visible="True">
  <script type="text/javascript">

    // <![CDATA[
    //jQuery.noConflict();
    $(function () {
      $("img.lazy").lazyload({
        effect: "fadeIn", threshold: 800, failure_limit: 9999
      });
    });
    $(window).load(function () {
      $("html,body").trigger("scroll");
    });
    // ]]>
  </script>
</div>
<asp:PlaceHolder ID="EndBody" runat="server" />
<asp:ContentPlaceHolder ID="CPH_EndBody" runat="server" />

<!--Código HTML de política de cookies -->
<div class="cookiesms" id="cookie1">
  Esta web utiliza cookies propias y de terceros para analizar su naveg
  ación y ofrecerle un servicio más personalizado y publicidad acorde a sus intere
  ses. Continuar navegando implica la aceptación de nuestra <a href="http://www.eur
  opapress.es/politica-cookies.html">política de cookies</a> -

  <button onclick="controlcookies()">Aceptar</button>
  <div class="cookies2" onmouseover="document.getElementById('cookie1')
  .style.bottom = '0px';">Uso de cookies</div>
</div>
<script type="text/javascript">
  if (localStorage.controlcookie > 0) {
    document.getElementById('cookie1').style.display = 'none';
  }
</script>
<!-- Fin del código de cookies -->

<!-- ad6 -->
<webep:BannerGTM ID="BannerGAMM_Under" runat="server" NombreSlot="Footer_
Publi" Tipo="TamannoPersonalizado" TamannoPersonalizado="1,1" />
<!-- ad6 -->

<!-- Inboard Teads -->
<!--
- <webep:BannerGTM ID="InBoard" runat="server" Tipo="TamannoPersonalizado" Tamann
oPersonalizado="1,1" /> -->

```

```
<!-- Inboard Teads -->
</form>

</body>
</html>
```



```
});  
  
}  
  
if (options) {  
    /* Maintain BC for a couple of versions. */  
    if (undefined !== options.failurelimit) {  
        options.failure_limit = options.failurelimit;  
        delete options.failurelimit;  
    }  
    if (undefined !== options.effectspeed) {  
        options.effect_speed = options.effectspeed;  
        delete options.effectspeed;  
    }  
    $.extend(settings, options);  
}  
  
/* Cache container as jQuery as object. */  
$container = (settings.container === undefined ||  
    settings.container === window) ? $window : $(settings.container);  
  
/* Fire one scroll event per scroll. Not one scroll event per image. */  
if (0 === settings.event.indexOf("scroll")) {  
    $container.on(settings.event, function () {  
        return update();  
    });  
}  
  
this.each(function () {  
    var self = this;  
    var $self = $(self);  
  
    self.loaded = false;  
  
    /* If no src attribute given use data:uri. */  
    if ($self.attr("src") === undefined || $self.attr("src") === false) {  
        if ($self.is("img")) {  
            $self.attr("src", settings.placeholder);  
        }  
    }  
  
    /* When appear is triggered load original image. */  
    $self.one("appear", function () {  
        if (!this.loaded) {  
            if (settings.appear) {  
                var elements_left = elements.length;  
                settings.appear.call(self, elements_left, settings);  
            }  
            $("                .one("load", function () {
```

```

        var original = $self.attr("data-
" + settings.data_attribute);
        $self.hide();
        if ($self.is("img")) {
            $self.attr("src", original);
        } else {
            $self.css("background-
image", "url('" + original + "')");
        }
        $self[settings.effect](settings.effect_speed);

        self.loaded = true;

        /* Remove image from array so it is not looped next time. */
        var temp = $.grep(elements, function (element) {
            return !element.loaded;
        });
        elements = $(temp);

        if (settings.load) {
            var elements_left = elements.length;
            settings.load.call(self, elements_left, settings);
        }
    })
    .attr("src", $self.attr("data-" + settings.data_attribute));
});

/* When wanted event is triggered load original image */
/* by triggering appear. */
if (0 !== settings.event.indexOf("scroll")) {
    $self.on(settings.event, function () {
        if (!self.loaded) {
            $self.trigger("appear");
        }
    });
}

});

/* Check if something appears when window is resized. */
$window.on("resize", function () {
    update();
});

/* With IOS5 force loading images when navigating with back button. */
/* Non optimal workaround. */
if ((/(:iphone|ipod|ipad).*os 5/gi).test(navigator.appVersion)) {
    $window.on("pageshow", function (event) {
        if (event.originalEvent && event.originalEvent.persisted) {
            elements.each(function () {
                $(this).trigger("appear");
            });
        }
    });
}

```

```
    });  
  }  
});  
}  
  
/* Force initial check if images should appear. */  
$(document).ready(function () {  
  update();  
});  
  
return this;  
};  
  
/* Convenience methods in jQuery namespace.          */  
/* Use as $.belowthefold(element, {threshold : 100, container : window}) */  
  
$.belowthefold = function (element, settings) {  
  var fold;  
  
  if (settings.container === undefined || settings.container === window) {  
    fold = (window.innerHeight ? window.innerHeight : $window.height()) + $window.scrollTop();  
  } else {  
    fold = $(settings.container).offset().top + $(settings.container).height();  
  }  
  
  return fold <= $(element).offset().top - settings.threshold;  
};  
  
$.rightoffold = function (element, settings) {  
  var fold;  
  
  if (settings.container === undefined || settings.container === window) {  
    fold = $window.width() + $window.scrollLeft();  
  } else {  
    fold = $(settings.container).offset().left + $(settings.container).width();  
  }  
  
  return fold <= $(element).offset().left - settings.threshold;  
};  
  
$.abovethetop = function (element, settings) {  
  var fold;  
  
  if (settings.container === undefined || settings.container === window) {  
    fold = $window.scrollTop();  
  } else {  
    fold = $(settings.container).offset().top;  
  }  
}
```

```

    return fold >= $(element).offset().top + settings.threshold + $(element).height(
);
});

$.leftofbegin = function (element, settings) {
    var fold;

    if (settings.container === undefined || settings.container === window) {
        fold = $window.scrollLeft();
    } else {
        fold = $(settings.container).offset().left;
    }

    return fold >= $(element).offset().left + settings.threshold + $(element).width(
);
});

$.inviewport = function (element, settings) {
    return !$.rightoffold(element, settings) && !$.leftofbegin(element, settings) &&
        !$.belowthefold(element, settings) && !$.abovethetop(element, settings);
};

/* Custom selectors for your convenience.  */
/* Use as $("img:below-the-fold").something() or */
/* $("img").filter(":below-the-fold").something() which is faster */

$.extend($.expr[":"], {
    "below-the-fold": function (a) { return $.belowthefold(a, { threshold: 0 }); },
    "above-the-top": function (a) { return !$.belowthefold(a, { threshold: 0 }); },
    "right-of-screen": function (a) { return $.rightoffold(a, { threshold: 0 }); },
    "left-of-screen": function (a) { return !$.rightoffold(a, { threshold: 0 }); },
    "in-viewport": function (a) { return $.inviewport(a, { threshold: 0 }); },
    /* Maintain BC for couple of versions. */
    "above-the-fold": function (a) { return !$.belowthefold(a, { threshold: 0 }); },
    "right-of-fold": function (a) { return $.rightoffold(a, { threshold: 0 }); },
    "left-of-fold": function (a) { return !$.rightoffold(a, { threshold: 0 }); }
});

})(jQuery, window, document);

```

## ANEXO C

*Código de MasterAMP.master, masterpage del formato noticia en AMP*

```
<%@ Master Language="VB" Inherits="Amp_MasterAMP" CodeBehind="MasterAMP.master.vb" %>

<%@ Register TagPrefix="webep" TagName="MenuLateral" Src="~/Modulos/MenuLateral.ascx" %>
<!DOCTYPE html>
<html amp lang="es">
<head runat="server">
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,minimum-scale=1,initial-scale=1">
    <script async src="https://cdn.ampproject.org/v0.js"></script><style amp-boilerplate>body{-webkit-animation:-amp-start 8s steps(1,end) 0s 1 normal both;-moz-animation:-amp-start 8s steps(1,end) 0s 1 normal both;-ms-animation:-amp-start 8s steps(1,end) 0s 1 normal both;animation:-amp-start 8s steps(1,end) 0s 1 normal both}@-webkit-keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}@-moz-keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}@-ms-keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}@-o-keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}@keyframes -amp-start{from{visibility:hidden}to{visibility:visible}}</style><noscript><style amp-boilerplate>body{-webkit-animation:none;-moz-animation:none;-ms-animation:none;animation:none}</style></noscript>
    <style amp-custom>
        /*Aquí van las líneas de código CSS inline*/
    </style>
    <script async custom-element="amp-font" src="https://cdn.ampproject.org/v0/amp-font-0.1.js"></script>
    <script async custom-element="amp-analytics" src="https://cdn.ampproject.org/v0/amp-analytics-0.1.js"></script>
    <script async custom-element="amp-ad" src="https://cdn.ampproject.org/v0/amp-ad-0.1.js"></script>
    <script async custom-element="amp-sidebar" src="https://cdn.ampproject.org/v0/amp-sidebar-0.1.js"></script>
    <script async custom-element="amp-accordion" src="https://cdn.ampproject.org/v0/amp-accordion-0.1.js"></script>

    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
    <asp:ContentPlaceHolder ID="Header_rel_next_prev" runat="server" />
</head>
<body class="opensans-amp-font-loading">
    <asp:ContentPlaceHolder ID="GA" runat="server"></asp:ContentPlaceHolder>
```

```

<header id="top" class="header-principal">
  <div class="brand-logo">
    <!--
<a href="http://m.europapress.es"><span class="europa">europa</span><span class="
press"> press</span></a>--%>
    <asp:ContentPlaceHolder ID="ContentPlaceHolderLogoMenu" runat="server
"></asp:ContentPlaceHolder>
    <button on='tap:sidebar.toggle' id="menutoggle" class=" navtoogle"></
button>
  </div>
</header>
<!-- <amp-sidebar id='sidebar'
layout="nodisplay"
side="left"> --%>
<webep:MenuLateral runat="server" />
<!--</amp-sidebar>--%>
<main role="main">
  <article>
    <asp:ContentPlaceHolder ID="PH_AMPBody" runat="server">
    </asp:ContentPlaceHolder>
  </article>
</main>
<section class="top">
  <a href="#top">Subir ↑</a>
</section>
<footer>
  <ul class="lista-footer">
    <li class="item-footer"><a class="span-
pie" href="http://www.europapress.es/noticias">Últimas noticias</a></li>
    <li class="item-footer"><a class="span-
pie" href="https://play.google.com/store/newsstand/news/Europa_Press?id=CAowjpSbB
w">Kiosko Google Play</a></li>
    <li class="item-footer"><a class="span-
pie" rel="nofollow" href="http://www.europapress.es/contacto.aspx">Contacto</a></
li>
    <li class="item-footer"><a class="span-
pie" rel="nofollow" href="http://www.europapress.es/avisolegal.html">Aviso legal<
/a></li>
  </ul>
  <div class="brand-logo">
    <span class="europa">europa</span><span class="press"> press</span><s
pan class="dev"></span>
  </div>
  <div class="div-pie">
    <span class="span-pie">Agencia privada líder en España</span>
  </div>
</footer>
  <amp-font layout="nodisplay" font-family="Open Sans" timeout="2000" on-error-
remove-class="opensans-amp-font-loading" on-error-add-class="opensans-amp-font-
missing" on-load-remove-class="opensans-amp-font-loading" on-load-add-
class="opensans-amp-font-loaded"></amp-font>
  <amp-analytics type="comscore">
<script type="application/json">

```

```
{  
  "vars": {  
    "c2": "13025493"  
  },  
  "extraUrlParams": {  
    "comscorekw": "amp"  
  }  
}  
</script>  
</amp-analytics>  
</body>  
</html>
```