



Ingeniaritza Goi Eskola Teknikoa
Escuela Técnica Superior de Ingeniería
Bilbao

eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

INGENIARITZA GOI ESKOLA
ESCUELA SUPERIOR DE INGENIERÍA
BILBAO

Trabajo de Fin de Máster

DE

**Optimización de la velocidad y rendimiento en
servicios web**

Titulación *Máster en Universitario en Ingeniería de Telecomunicación*

Alumno Asier Unanue Esteban

Tutor de proyecto Juanjo Igarza Ugaldea

Fecha *Septiembre de 2017*

Referencia
Trabajo de Fin de Master

Curso Académico
2016/2017

Optimización de la velocidad y rendimiento de Servicios Web

ÍNDICE

	pág.
I. RESUMEN	6
II. ABSTRACT	7
III. LABURPENA	8
1. INTRODUCCIÓN	9
2. OBJETIVOS	13
3. ALCANCE	14
3.1.1 Alcance del proyecto	14
4. BENEFICIOS SOCIALES, TÉCNICOS Y ECONÓMICOS	17
4.1 Beneficios técnicos.....	17
4.2 Beneficios económicos.....	17
4.3 Beneficios sociales.....	18
5. ESTADO DEL ARTE	19
5.1 Web Performance Optimization.....	19
5.2 Search Engine Optimization (SEO)	19
5.2.1 Posicionamiento interno	20
5.2.2 Posicionamiento externo	22
5.2.3 Content Delivery Networks	22
5.3 Accelerated Mobile Pages.....	26
6. ANÁLISIS DE ALTERNATIVAS.....	29
6.1 Objetivo del análisis	29
6.2 Alternativas.....	29
6.2.1 Infraestructura del servidor	29
6.2.1.1 LAMP.....	29
6.2.1.2 MEAN	31
6.2.2 Tecnologías web optimizadas para terminales móviles	32

6.2.2.1	Accelerated Mobile Pages	33
6.2.2.1.1	Fundamentos de AMP	35
6.2.2.1.2	Configuración del indexado AMP.....	36
6.2.2.1.3	AMP en los buscadores	37
6.2.2.2	Otros frameworks web.....	37
6.2.2.2.1	Fundamentos de Bootstrap.....	38
6.2.3	Servidor de contenido.....	38
6.3	Criterios de selección	40
6.3.1	Infraestructura del servidor	40
6.3.2	Frameworks web para terminales móviles.....	40
6.3.3	Servidor de contenido.....	41
6.4	Selección de la solución	42
6.4.1	Infraestructura del servidor	42
6.4.2	Frameworks web	43
6.4.3	Servidor de contenido.....	44
7.	METODOLOGÍA.....	47
7.1	Modelo de desarrollo	47
7.1.1	Definición de requisitos del sistema	48
7.1.2	Diseño del sistema	48
7.1.3	Diseño del software	50
7.1.4	Codificación	51
7.1.5	Verificación del software.....	51
7.1.6	Verificación del sistema	52
7.1.7	Validación del sistema	52
7.2	Procedimiento de minificación.....	53
7.3	Resultados de la verificación del sistema.....	53
7.3.1	Google Analytics.....	54
7.3.2	GTMetrix.....	57
7.3.3	Waterfall	58
7.4	Procedimiento de despliegue del NDC.....	59
7.5	Conclusiones.....	60
8.	PLAN DEL PROYECTO.....	63

8.1	Introducción.....	63
8.2	Equipo de trabajo	63
8.3	Paquetes del proyecto.....	64
8.4	Duración total	72
8.5	Hitos	73
8.6	Diagrama de Gantt	74
9.	RECURSOS HUMANOS Y MATERIALES	76
9.1	Recursos humanos	76
9.2	Recursos materiales.....	77
10.	Presupuesto	78
10.1	Costes unitarios.....	78
10.1.1	Mano de obra	78
10.1.2	Material amortizable	78
10.1.3	Material fungible	79
10.2	Gastos en Recursos Humanos.....	80
10.3	Gastos en recursos materiales.....	81
10.4	Resumen final de los gastos del proyecto	82
11.	RIESGOS DEL PROYECTO Y PLAN DE CONTINGENCIA.....	83
12.	CONCLUSIONES	85
13.	REFERENCIAS	86

ÍNDICE DE TABLAS

	pág.
Tabla 5–1: Regiones CDN Google Cloud Flare [16].....	24
Tabla 5–2: Regiones Amazon Cloud [17].....	25
Tabla 6–1: MongoDB VS MySQL - Resultados.....	43
Tabla 6–2: MongoDB VS SQL. Comparativa	43
Tabla 6–3: AMP vs Bootstrap. Comparativa	44
Tabla 6–4: VPS vs Host Físico. Comparativa	45
Tabla 7–1: Comparativa GTMetrix - I.....	57
Tabla 8–1: Personal encargado del proyecto.....	64
Tabla 8–2: Paquete de trabajo 1. Horas de trabajo por persona según tarea	65
Tabla 8–3: Paquete de trabajo 2. Horas de trabajo por persona según tarea	66
Tabla 8–4: Duración total del proyecto.....	72
Tabla 8–5: Hitos	73
Tabla 10–9–1: Recursos materiales.....	77
Tabla 10–9–2: Costes en Recursos Humanos.....	78
Tabla 10–9–3: Costes en material amortizable	78
Tabla 10–9–4: Costes en Material Fungible.....	79
Tabla 10–9–5: Costes en Recursos Humanos.....	80
Tabla 10–9–6: Costes en Recursos Materiales.....	81
Tabla 10–9–7: Costes totales del proyecto	82
Tabla 11–1: Evaluación de riesgos	84

ÍNDICE DE FIGURAS

Figura 1—1: Distribución geográfica del servicio [1]	10
Figura 1—2: No CDN vs CDN [3].....	12
Figura 4—1: Porcentaje de accesos por posición en búsqueda [1].....	18
Figura 5—1: CDN [14]	23
Figura 5—2: CDN World [15]	23
Figura 5—3: Búsqueda AMP	27
Figura 5—4: Iframe AMP	28
Figura 6—1: Esquema LAMP [7]	30
Figura 6—2: Esquema MEAN [7].....	31
Figura 6—3: Valicación URL AMP	34
Figura 6—4: Bootstrap Desktop Browsers [19]	38
Figura 6—5: Bootstrap Mobile browsers	38
Figura 6—6: Sesiones por país [1].....	45

I. RESUMEN

El objetivo de este proyecto es conseguir aumentar la velocidad de páginas web y, consecuentemente, mejorar su posicionamiento en los buscadores mediante diversas técnicas, de modo que los potenciales usuarios de las páginas se vean aumentados y su experiencia de uso mejore. Para lograrlo, se analiza el conjunto de técnicas existentes, realizándose las pruebas pertinentes que permitan conocer qué técnicas son las óptimas para llevar a cabo el propósito principal de mejorar la velocidad de las páginas.

En el presente documento, se lleva a cabo el procedimiento necesario para lograr los mejores resultados. En primer lugar, se describe cada una de las técnicas y tecnologías llevadas al estudio con el fin de poder analizar y conocer cuáles son las ventajas e inconvenientes de cada una de ellas. A continuación, se llevan a cabo las mediciones necesarias para conocer cuáles son las técnicas que se deben aplicar. Dichas mediciones consisten en el empleo de herramientas, como por ejemplo GTMetrix, que miden diferentes aspectos durante la carga de una página web, a saber, diferentes tiempos de carga, número de peticiones realizadas al servidor, etc. Una vez obtenidos los datos, se escogen las opciones óptimas y, aplicadas en conjunto, se muestra el resultado final comparado con el inicial. El conjunto de técnicas para la mejora del posicionamiento de páginas web que se emplea en este estudio son recogidas del directorio del World Wide Web Consortium W3C o Consorcio Internacional Web, que es el encargado de publicar las nuevas técnicas que hacen posible a las webs adaptarse a las nuevas plataformas.

Dado que la página web a tratar ya obtiene un número de visitas mucho más elevado desde plataformas móviles que desde escritorio, las tecnologías a aplicar pretenden optimizar el rendimiento para el acceso a través de toda clase de dispositivo móvil. A lo largo de este documento, se puede ver cómo se aplican tecnologías como Accelerated Mobile Pages AMP, se modifican aspectos de la arquitectura de la Red de Distribución de Contenidos RDC mediante Nodos de Distribución de Contenidos NDCs, e incluso se plantea un cambio de la arquitectura de capas de los servidores web.

II. ABSTRACT

The main objective of this project is, using different approaches, to reduce the webpage load time and therefore to improve the search engine rankings in order to gain traffic and get better user experience. To complete this goal, different options are going to be tested and the best ones will be chosen to increase the load time speed of the website which is the main purpose of this project.

In this document, it is explained the procedure to achieve the desired results. On one hand, the different technologies and procedures available are described with their advantages and disadvantages. On the other hand, the results using these procedures are showed and explained so the best solution can be selected. The results are going to be tested using software programs like GTMetrix, a webtool that measures page load time, number of calls made to the webserver, etc. After studying the results mentioned above, the best procedures are selected and applied to the webpage and the final results are compared with the initial starting point. The set of procedures used to study the improvement of search engine rankings are collected from the repository of the World Wide Web Consortium Web W3C, also called International Consortium Web, which is the responsible to maintain and to introduce the global standards that must be followed by the devices connected to the web.

Since the webpage to be studied gets its traffic mainly from mobile platforms the project will be focused on these type of devices. Different solutions will be discussed such as AMP (Accelerated Mobile Pages AMP), CDN (Content Distribution Network) architecture along with the Content Distribution Nodes NDCs, and webserver plugins.

III. LABURPENA

Proiektu honen xedea web orrien abiadura igotzea da, eta honekin batera ondorioz, bilatzaileetan orriek duten posizioa hobetzea, egun eskuragarri dauden teknikak erabiliz. Era honetan web orri horien erabiltzaile kopurua hasi egingo da gainera erabiltzaileon esperientzia hobetuz. Hori lortzeko gaur egun eskuragarri dauden teknikak erabiltzen dira eta dagozkien probak egiten dira teknika egokienak eta konbinazio optimoak zein diren jakiteko, beti asmo bakarra gogoan izanda: orrien abiadura hobetzea.

Dokumentu honetan emaitzak doitzeko eta egokitzeko jarraitu beharreko prozedura garatzen da. Lehenik eta behin, erabili beharreko teknika eta teknologia bakoitza azaltzen da bakoitzaren abantailak eta eragozpenak aztertzeko asmoz. Ondoren, lortutako emaitzak neurtzen dira erabiliko diren teknologiak aukeratzeko. Neurketa hauek adibidez GTMetrix bezalako erreminten erabileran datza, kasu honetan web orrien kargako parametroak neurtzen ditu, hala nola: karga-denborak edo zerbitzariari eginiko eskari kopurua. Datu hauek jaso eta gero, aukera optimoak hautatzen dira, eta multzo egokia hautatuz, lortutako azken balioa hasieran geneukanarekin konparatzen da. Web orrien posizioa hobetzeko ikerlan honetan erabiltzen den teknika multzoa World Wide Web Consortium W3C edo Nazioarteko Web Partzuergoaren direktoriotik jaso da. Partzuergo honen arduren artean web orriak plataforma berrietara egokitzeko teknika berriak proposatzea eta argitaratzea daude.

Erabiliko den web orriak dagoeneko bisita gehien plataforma mugikorretarik jasotzen dituzenez, aplikatuko diren teknologiek edozein motako gailu mugikorretik eginiko atzipenak optimizatzea dute helburu. Dokumentu honetan zehar ikusi ahal izango da nola erabiltzen diren Accelerated Mobile Pages AMP bezalako teknikak, nola aldatzen diren Content Distribution Network CDN-ren arkitekturaren ezaugarriak Content Distribution Nodoen bidez edo nola web zerbitzarien geruzen arkitekturan aldaketa proposatzen den.

1. INTRODUCCIÓN

En la actualidad, las páginas web están adquiriendo cada vez una mayor complejidad debido a la gran cantidad de plugins, publicidad dinámica y animaciones que es posible incluir en ellas. A pesar de que esto resulta positivo para mejorar la interacción con el usuario y ofrecerle una mejor experiencia de usuario, es algo que afecta muy negativamente a los tiempos de carga de la página web.

Por otra parte, el fácil acceso a las tecnologías móviles, tales como smartphones o tablets, supone un crecimiento continuo de usuarios que acceden desde dichos dispositivos. Esto está dirigiendo a los servicios web hacia un nuevo paradigma en el que se pretende conseguir que las páginas sean adaptativas (responsive), es decir, que sirvan para cualquier tipo de dispositivo y tamaño de pantalla. Ello añade aún más elementos de complejidad a la página a fin de que sea compatible con todas las plataformas.

Como resultado de esta tendencia, cada vez es más frecuente que aumenten los tiempos de carga del código de las páginas web. Por ello, algunos actores importantes en el mundo de la tecnología que envuelve internet, como por ejemplo Google u otros proveedores de internet (ISPs), pretenden concienciar a los desarrolladores para que optimicen la velocidad de carga, reduzcan la cantidad de información necesaria para visualizar las páginas y optimicen el ancho de banda global que éstas ocupan desde que salen del servidor hasta que llegan al usuario final.

En el mencionado contexto, este proyecto pretende optimizar la velocidad de carga de páginas web y verificar la eficiencia de las nuevas normas y tecnologías que son propuestas por algunos organismos o entidades como Google.

Conseguir reducir el tiempo de carga de las páginas y optimizar su código es algo fundamental puesto que, de este modo, se mejora la experiencia del usuario a la vez que se requieren redes con menor ancho de banda y dispositivos con menores requisitos hardware. En la imagen que se muestra a continuación, obtenida mediante Google Analytics, se muestran el número de sesiones existentes contra un dominio web, en el periodo de un mes y distribuidas geográficamente.

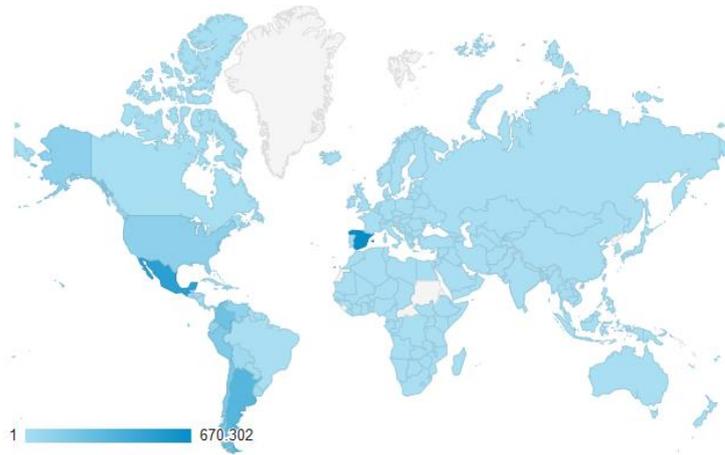


Figura 1—1: Distribución geográfica del servicio [1]

En la actualidad, existen millones de páginas web que tratan diferentes temáticas. Por ello, es posible que muchas de estas páginas coincidan en su objetivo y temas tratados, siendo en el caso que atañe al estudio, el tema en cuestión la salud. Para este proyecto, se emplea un dominio web que consiste de un servicio que recoge a un gran número de doctores y clínicas a lo largo de diferentes países de habla hispana. Además, permite a sus usuarios conocer qué médicos y de qué especialidad se encuentran más cerca de la zona que les interesa o bien, más cerca de su ubicación en ese momento. Por otro lado, ofrece un servicio de carácter informativo que recoge información acerca de miles de enfermedades y diferentes afecciones relacionadas con la salud de las personas. Ambos servicios relacionados con la medicina, son desplegados en países donde existen hispanoparlantes.

El Search Engine Optimization SEO es el conjunto de algoritmos empleado por los buscadores para obtener resultados relacionados con una búsqueda y mostrarlos de manera ordenada. Dichos algoritmos, incluyen nuevas variables o, modifican algunas existentes anteriores, para llevar a cabo la ordenación de los resultados obtenidos para una búsqueda. Una mejora en los tiempos de carga de las páginas web acompañada de la optimización del código de dichas páginas, podría favorecer a los sitios web cuando son posicionados en las listas de resultados.

A fin de lograr una mayor velocidad, se requiere una correcta programación y estructuración del sitio web de modo que se propicien tiempos de carga más bajos. A modo introductorio, algunos de los métodos empleados para disminuir el tiempo de carga de una web son [2]:

- Realizar la carga de fuentes de manera asíncrona. De este modo se consigue cargar la web con una fuente base que es modificada cuando todo el contenido ha sido cargado.
- Eliminar clases definidas no utilizadas por ningún otro recurso de la web, como puede ser un fichero de estilos CSS.
- Llevar a cabo la minificación del contenido servido, evitando así el envío de código contenido innecesario y disminuyendo consecuentemente el tamaño de los ficheros enviados.

Por otra parte, en el caso del sitio web tratado en este proyecto, se posee información mostrada sobre una estructura preestablecida, es decir, la información es obtenida mediante peticiones a diferentes bases de datos (BB.DD) y mostrada sobre plantillas. Actualmente, existen diferentes modalidades de acceso a las BB.DD entre las que se encuentran SQL y el uso de JavaScript, las cuales se presentan y comparan más adelante.

Dejando de lado los aspectos de la programación y otros aspectos software, uno de los puntos más influyentes en el tiempo de carga de una página web es la distancia desde el cliente hasta el servidor. Este aspecto físico es menester considerarlo ya que, a mayor distancia, el Round-Trip Time RTT o tiempo de “ida y vuelta” se ve considerablemente aumentado, no solo por el tiempo de propagación sino por la cantidad de nodos intermedios que ha de atravesar la información para alcanzar su objetivo. Por este motivo, es fundamental poseer un Nodo Distribuidor de Contenidos próximo a los usuarios.

En el caso de estudio, el dominio A pretende dar servicio a un conjunto de países de habla hispana. Entre estos países, existen algunos casos en los que no es posible disponer de un Nodo Distribuidor de Contenidos NDC contratado a una empresa con servicios de tipo Content Delivery Network CDN o Red de Distribución de Contenidos RDC. Es el caso, por ejemplo, del dominio www.dominioA.mx, correspondiente a México y enfocado a usuarios mejicanos. Para que el *dominio A* obtenga el mejor rendimiento posible y aumente su velocidad de carga, es necesario configurar un NDC geográficamente cercano que posea una réplica del contenido a servir. De este modo, las peticiones al directorio mejicano se realizan al nuevo NDC situado en este país, pues es el más próximo (para lo cual, también es necesario anunciar este NDC a los nodos de resolución de nombres DNS).

Optimización de la velocidad y rendimiento de Servicios Web

En la siguiente figura se muestra, a la izquierda una red sin una CDN en funcionamiento, es decir, un único servidor de contenidos para todos los clientes. Por otro lado, a la derecha, se muestra una arquitectura RDC en la que los clientes realizan peticiones a los servidores réplica NDC más cercanos.

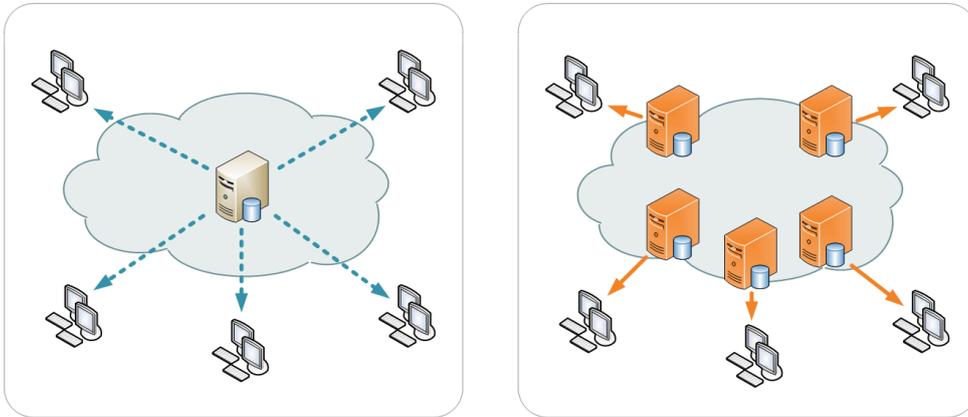


Figura 1—2: No CDN vs CDN [3]

Como se puede observar en la imagen superior, los usuarios son servidos desde un servidor diferente en función de la configuración de los proxy servers, que este caso tiene que ver con la distancia entre cliente y servidor, es decir, con su situación geográfica.

2. OBJETIVOS

El objetivo principal de este proyecto es la mejora de la velocidad de carga de las páginas web. Para lograrlo, el proyecto sigue las siguientes premisas:

- Mejorar la infraestructura. Esto supone la inclusión de nuevos servidores o Nodos Distribuidores de Contenidos NDCs.
- Aplicar tecnologías que permitan la carga asíncrona de los elementos de las páginas web, como pueden ser la carga de fuentes y estilos, publicidad, imágenes, etc.
- Comprimir y reducir la cantidad e información que atraviesa la red para lograr mostrar el contenido al usuario. Para ello se recurre a la minificación, a la optimización de los ficheros CSS, a la compresión de imágenes, entre otros procedimientos.

Consecuentemente, una vez se logra mejorar la velocidad de las páginas web, se obtienen beneficios conseguidos por el buen hacer, por ejemplo:

- Mejorar la experiencia de los usuarios que visiten la página web.
- Mejorar la valoración de posicionamiento de los buscadores, es decir, que los algoritmos SEO otorguen una mayor puntuación a la página web y la coloquen más arriba en el listado de resultados mostrados para una búsqueda.

3. ALCANCE

Este apartado describe el alcance del proyecto y cuáles son los procedimientos que se han de llevar a cabo para satisfacer los objetivos.

3.1.1 Alcance del proyecto

El objetivo general del proyecto consiste en la mejora de la velocidad de carga de páginas web mediante procedimientos aplicables a cualquier página diseñada. Para conseguirlo, es necesario finalizar cada una de las partes que contribuyen a esta mejora. A continuación, se describe cada una de ellas:

- Despliegue e incorporación de un Nodo Distribuidor de Contenido

Para poder conseguir que la página web mejore su velocidad de carga, se debe ampliar la red de distribución de contenidos formada por múltiples servidores NDC que pueden ser desplegados en cualquier punto de interés. El despliegue de un Nodo Distribuidor de Contenido (NDC) está formado de varias fases:

- Creación del NDC

Lo primero que se ha de conseguir es tener operativa una máquina capaz de servir contenido. Para ello, se han de emplear el conjunto de herramientas software que posibiliten lo siguiente:

- Almacenamiento replicado de información, de manera estructurada y mediante bases de datos. La información debe poder ser accedida, modificada y eliminada.
- Se requiere la existencia de un servidor web capaz de atender las peticiones de los clientes y devolverles el contenido obtenido de las bases de datos.
- El servidor mencionado en el punto anterior, debe poseer el código, réplica del existente en el NDC de origen, capaz de ofrecer la interfaz de usuario completada con la información obtenida de la base de datos en base a la solicitud realizada por el cliente.

- Anuncio de nueva posición y dirección IP

Como bien es sabido y se ha mencionado previamente, el contenido de una página web puede estar alojado en uno o varios servidores. Al realizarse una solicitud a un sitio web se da, en primer lugar, un intercambio de mensajes con el nodo de resolución de nombres DNS que tenga configurado el equipo del cliente. El nodo de resolución de nombres devuelve al cliente la dirección IP previamente configurada (si no la conoce la solicita a un servidor de nivel superior), que posee el servidor que sirve el contenido de la web solicitada. Por lo tanto, es necesario que el servidor o servidores web anuncien su dominio y su IP a la red para darse a conocer de modo que el DNS pueda redirigir a los clientes hacia el servidor oportuno.

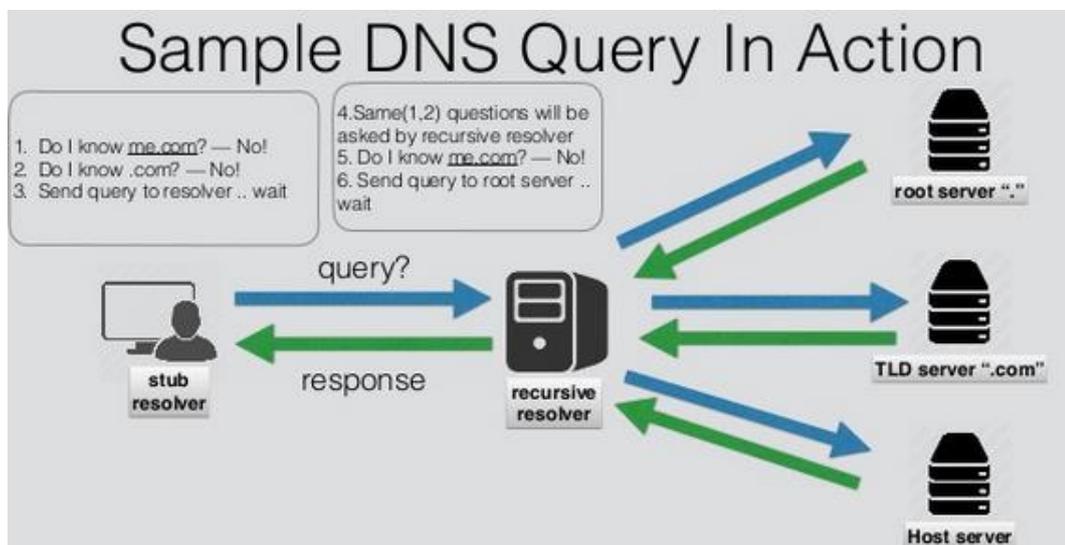


Ilustración 3—1: Procedimiento de resolución de nombres [4]

Antes de que un servidor DNS admita una nueva entrada a su tabla de nombres, es necesario un servidor DNS autoritativo que, en primer lugar, verifique si el nombre de dominio empleado se encuentra en sin asignar y, en tal caso, proceda a distribuir la relación dominio/dirección IP al resto de nodos de resolución de nombres. De este modo, se consigue que cuando un cliente intenta acceder al dominio en cuestión, el servidor de resolución de nombres pueda proporcionarle la IP del servidor de contenidos más cercano.

- Implementación de tecnologías web óptimas

Lo siguiente que se ha de realizar es la modificación del código de servidor existente a fin de adaptarlo a nuevas metodologías de programación pensadas

para mejorar la velocidad de carga de las páginas web. Una vez analizadas las opciones más empleadas en la actualidad y ser comparadas con las nuevas técnicas emergentes, se debe aplicar la óptima según los resultados que se obtengan en la fase de pruebas piloto. Para llevar a cabo la evaluación de las tecnologías óptimas, se recurre a herramientas que han de aportar información acerca de los siguientes puntos:

- Velocidad de carga: depende del tiempo que tarda desde que se realiza la petición al servidor hasta que el cliente web muestra la respuesta al usuario.
- Peticiones al servidor y ausencia de servidores externos: una menor cantidad de peticiones para cargar elementos al servidor supone un menor tiempo de carga. Por otra parte, la ausencia de varios servidores para la carga de contenidos elimina la dependencia de dichos servidores al no existir peticiones externas.
- Tamaño del contenido y optimización del mismo: Cuanto menor sea el tamaño del contenido, menor ancho de banda se requiere para su transferencia a los clientes. Existen diversas técnicas que permiten, para una misma respuesta, reducir su tamaño y optimizarla. Para ello se emplean métodos de compresión de contenidos como, por ejemplo, imágenes y vídeos. Por otro lado, se puede comprimir el código de las páginas si éste es minificado.

4. BENEFICIOS SOCIALES, TÉCNICOS Y ECONÓMICOS

En este proyecto cuyo objetivo es la mejora de la velocidad de carga de diferentes servicios web mediante la aplicación de todas las estrategias existentes en la actualidad se pueden obtener diferentes beneficios: técnicos, económicos y sociales.

4.1 Beneficios técnicos

El proyecto propone el estudio de diferentes aspectos novedosos en el ámbito de la programación web. Pretende, además de verificar la eficacia real de las nuevas estrategias para poder ser aplicada en el desarrollo de cualquier página web posteriormente.

La consecución de la optimización de páginas web supone:

- Reducción de los tiempos de carga.
- Reducción del número de peticiones realizadas para cargar un contenido.
- Reducción de la tasa de datos requerida para la carga de la página web.
- Reducción de la carga de trabajo de los servidores.
- Mejora del posicionamiento web en los buscadores.

4.2 Beneficios económicos

La optimización web supone una serie de mejoras y beneficios económicos que afectan, no solamente a las empresas desarrolladora de las páginas web sino también a los usuarios de las mismas. Dichos beneficios son los siguientes:

- Se requieren servidores con menor capacidad reduciendo su coste.
- Se mejora la experiencia de usuario favoreciendo la fidelización y logrando que la web en cuestión sea considerada previamente a una búsqueda.
- Con la mejora del posicionamiento en los buscadores se obtiene una mayor cantidad de tráfico de modo que aparezcan nuevos modos de financiación como puede ser la publicidad o la inclusión de servicios Premium. En la siguiente gráfica se muestra cómo la posición en los buscadores es imprescindible para obtener mayor cantidad de visitas.

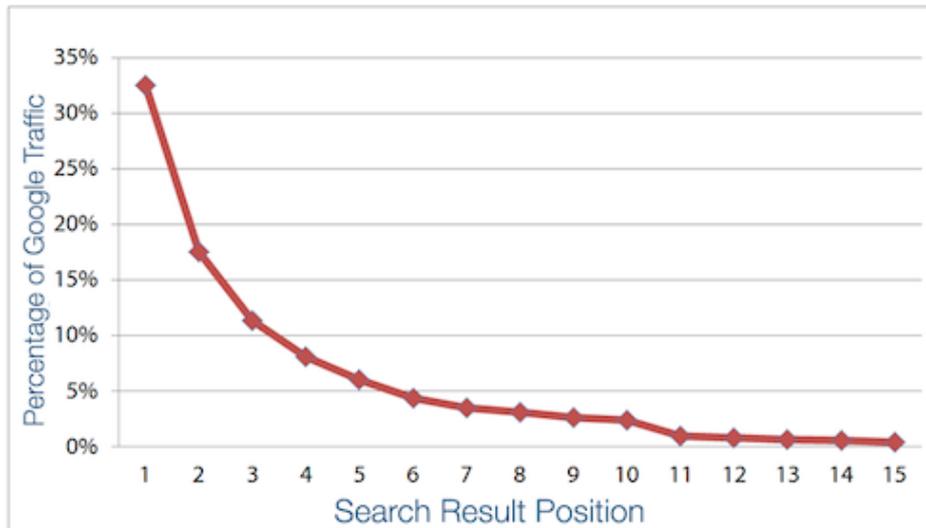


Figura 4—1: Porcentaje de accesos por posición en búsqueda [1]

Actualmente, la posición media del dominio que se trabaja en este proyecto se encuentra de media, entre la posición 5 y 6.

Además, 120 millones de mexicanos poseen, para el dominio web tratado en este proyecto, un nuevo NDC geográficamente cercano supone un nuevo punto de acceso al servicio que debe atraer una mayor cantidad de usuarios que incrementen los beneficios.

- Reducción de la tasa de datos requeridos para cargar una misma página web. De este modo, los usuarios reducen su consumo de datos a la par que su factura de móvil.

4.3 Beneficios sociales

Además de los beneficios técnicos y económicos mencionados en los puntos anteriores, la ejecución de las medidas propuestas en este proyecto aporta también algunos beneficios sociales detallados a continuación.

- Aumento del número de usuarios a los que se les muestra la posibilidad de acceder a un servicio de calidad.
- Se pone a disposición de una mayor cantidad de usuarios información o servicios que pueden serles de utilidad.

5. ESTADO DEL ARTE

5.1 Web Performance Optimization

El Web Performance Optimization WPO es de suma importancia. Los datos recogidos en el siguiente párrafo demuestran que la pérdida de usuarios debida al tiempo de carga de las webs se puede traducir en varios millones de pérdidas.

Para Bing, añadir dos segundos al tiempo de carga de su página supone una pérdida de un 4,3% de usuarios. Además, Google explica que una página 400 milisegundos más lenta que otra generan una pérdida de 0.59% de búsquedas por usuarios y lo que ello conlleva en cuanto a ingresos por publicidad [5].

El tiempo de carga óptimo debe estar por debajo de los 4 segundos. Sin embargo, el tiempo medio de carga oscila entre 1 y 8 segundos. El motivo por el cual se pierden más beneficios es que cerca del 60% de los usuarios abandonan las páginas web que tardan en cargar más de 3 segundos y un 25% adicional si tarda en cargar más de 4 segundos [6].

5.2 Search Engine Optimization (SEO)

Se conoce como Search Engine Optimization al conjunto de procedimientos que permiten optimizar las búsquedas de contenido web de manera que los buscadores optimicen sus resultados. Estos procedimientos consisten en una serie de procesos técnicos que han de efectuar cambios tanto en la estructura como en la información que muestran las páginas web. Con ello se pretende mejorar la visibilidad de una página web en las listas de resultados obtenidos que muestran los buscadores existentes.

Las técnicas a desarrollar incumben tanto cambios en la programación, como en el diseño y el contenido de las páginas web. Estos cambios han de ser realizados en base a una serie de directrices y consejos de buenas prácticas promovidos periódicamente por buscadores como Google, Yahoo o Bing. No obstante, las directrices no siempre son de carácter “fijo”, lo cual significa que, algunas directrices sí son realmente publicadas por los buscadores, pero otras sin embargo, son sospechas de la comunidad de desarrolladores basadas en

pruebas empíricas. Las técnicas pueden ser separadas en dos tipos: de posicionamiento interno [7] y de posicionamiento externo [8].

5.2.1 Posicionamiento interno

Las técnicas de posicionamiento interno hacen referencia a las mejoras que el desarrollador de una página web ha de aplicar sobre el contenido y los aspectos como la apariencia o la accesibilidad que le afectan. El conjunto de directrices emitidas por los buscadores varía a lo largo de los años, no obstante, a continuación, se muestran algunas de las que aún permanecen vigentes:

- El diseño de la web ha de ser responsivo o adaptable. Esto implica que el contenido de las páginas web debe mostrarse adecuadamente independientemente del tamaño de la pantalla del dispositivo que reproduce dicho contenido. Las páginas web que no adapten su contenido a las características de los dispositivos de los clientes son fuertemente penalizadas en los resultados de búsqueda [8].
- La página web debe facilitar su propio rastreo. Cuando un buscador lanza una “araña” en búsqueda de contenidos relacionados con una búsqueda analiza millones de páginas web. Es fácilmente comprensible que, si el desarrollador web colabora en la búsqueda de estos contenidos mediante el uso de marcadores o etiquetas de contenido, el tiempo de búsqueda por parte del buscador disminuye y, consecuentemente, la página web será mostrada más fácilmente [9].
- Se debe evitar el contenido duplicado lo antes posible. El mismo contenido publicado bajo diferentes nombres es fácilmente detectable y, aunque a priori proporcione mayor número de visitas, es fácilmente detectado y penalizado [10].
- Los contenidos deben ser de calidad y mejorar la experiencia del usuario. Para lograr mejorar dicha experiencia, es necesario facilitar el uso de la página web, así como disminuir el tiempo de carga. Por otro lado, facilitar la navegación entre contenidos mediante enlaces internos dentro de una misma página web es altamente positivo para la experiencia del usuario. No obstante, la página no debe constar con excesivos enlaces de salida aunque sean de carácter interno ya que debido a lo que se conoce como “Link Juice”, la importancia de los enlaces existentes en una página web se reparte entre ellos [11].
- El contenido debe ser original, no copiado u obtenido de otras páginas web. La realización de esto puede conllevar a infligir el anterior consejo, creándose contenido duplicado y, consecuentemente, obteniendo una penalización [8].

- La estructura y el diseño de la página web debe estar realizado pensando en el usuario. Es por ello que ambos son continuamente modificados y adaptados a las nuevas directrices, realizando las variaciones que son consideradas óptimas para mejorar la funcionalidad, el acceso y la atención captada por parte de la página hacia el usuario.
- Los títulos deben ser únicos. Además, el título etiquetado como `<h1>` es de vital importancia para que la página sea mostrada cuando un usuario realiza una búsqueda de un tema determinado [8].
- La página web debe ser lo más ligera posible, evitando el contenido flash, iframes o la ejecución de códigos JavaScript que ralenticen su funcionamiento. Sin embargo, evitar por ejemplo el uso de JavaScript se hace altamente complicado dado que aumenta las posibilidades y funcionalidades que para algunos sitios web pueden ser consideradas muy importantes [8].
- Evidentemente, la página web debe estar alojada en un servidor rápido y fiable.
- Las URL deben ser amigables y significativas para la búsqueda. Es interesante que las URLs contengan palabras clave o keywords sobre las que se desea posicionar el contenido. Esto puede conllevar un procedimiento de traducción mediante el fichero `htaccess` del directorio en cuestión.
- Se recomienda que el sitio web tenga instalado un certificado SSL y emplee enlaces HTTPS para redirigir a otros contenidos web [12].
- En cuanto al contenido, debe estar correctamente etiquetado tal y como se ha mencionado en uno de los puntos anteriores. Para ello, pueden emplearse métodos como el uso de la negrita o cursiva para destacar las palabras clave relacionadas con el tema, emplear etiquetas `<meta>` con las palabras claves del contenido o el uso de cabeceras `<h1>`, `<h2>`, etc. que destaquen los apartados más importantes del sitio web mostrado. Sin embargo, el uso de keywords no debe ser excesivo [8].
- Debe evitarse el empleo de imágenes para los menús o texto en general. El texto ha de ser codificado como texto y a ser posible nunca en imágenes a excepción de tipografías no estándar [8].
- Todas las imágenes deben contener el atributo ALT ya que Google no es capaz de reconocer las fotos y es necesario indicárselo. Esto cobra especial importancia especialmente si la imagen, además, es un enlace a otro sitio web [8].
- Emplear etiquetas de HTML5 además de acatar su estructura de secciones. La etiqueta más importante de una página web es el `<title>`. Cada sitio web debe poseer un `<title>` único que describa el contenido de la página y contenga las keywords más representativas. A continuación, la etiqueta `<description>` puede

ser de una o dos frases que describan el contenido de la página. El contenido de esta etiqueta es muy importante ya que suele ser el resumen empleado por los buscadores cuando un sitio web aparece mostrado en sus resultados de búsqueda.

- Emplear “migas de pan” a través de un Content Management System CMS, que permita tener el control de los contenidos de diferentes bases de datos y relacionarlos entre sí. Gracias a estas migas de pan es posible subir o bajar de nivel en la estructura de la información mostrada en el sitio web. Así, por ejemplo, si la página web posee migas de pan del tipo “Enfermedades → Alergias → Alergia al polen” ha de ser posible, desde la página de Alergia al polen, volver al nivel de Enfermedades a través de una miga [13].

5.2.2 Posicionamiento externo

En este apartado se recogen los métodos empleados para conseguir que el sitio web deseado se mencionado o citado en otros sitios web. Cuantas más menciones online mayor es el número de potenciales visitas. Para conseguir aumentar este número de menciones se recomienda lo siguiente [10]:

- Conseguir enlaces desde páginas webs cuya temática sea similar a la nuestra de manera que los contenidos de ambas páginas sean complementarios.
- La participación en las redes sociales continúa siendo desde hace varios años un aspecto importante en el posicionamiento.
- Registrarse en foros y participar de ellos pueden suponer un aporte sustancial de visitas al sitio web deseado. La consecución de esto no consiste en publicar enlaces al sitio web en cada dominio dado que, de cara al posicionamiento, la importancia de las URL es inversamente proporcional al número de veces que aparecen en un sitio web, en este caso el foro. Para lograr una correcta “publicidad” del sitio web en foros, el usuario debe poseer en su firma el enlace al sitio web que desea promocionar.
- Aunque se consigan enlaces de diferentes sitios web, no todos son valorados de igual manera. Conseguir enlaces desde sitios de prestigio o “Trust-Rank” debe ser un objetivo.

5.2.3 Content Delivery Networks

Una Content Delivery Network (CDN) o red de entrega de contenidos, es una red formada por diferentes equipos que poseen copias de archivos y/o

disponen de contenido exclusivo en función de la zona en que se encuentran (o bien replicado) a fin de mejorar la experiencia del usuario que emplea un servicio. Su principal objetivo, es disminuir los tiempos de carga de los contenidos, así como dotar al servicio de redundancia de contenidos.



Figura 5—1: CDN [14]

En la actualidad existen diferentes proveedores de redes de entrega de contenido, entre los cuales se encuentran Google Cloud Flare, Amazon, y Akamai. Este conjunto de empresas mencionadas anteriormente (y otras), despliegan redes con la forma mostrada en la siguiente imagen:



Figura 5—2: CDN World [15]

En la actualidad, las empresas mencionadas poseen sedes en los siguientes lugares:

➤ **Google Cloud Flare**

Tabla 5-1: Regiones CDN Google Cloud Flare [16]

Regiones	Nodos por zona
Oregon	2
California	3*
Iowa	4
Montreal	3*
N. Virginia	3*
S. Carolina	3
Londres	3*
Belgica	3
Holanda	2*
Frankfurt	3*
Finlandia	3*
Mumbai	3*
Singapur	2
São Palo	3
Tokio	3
Taiwan	3
Sydney	3*

Nota: Están señalizadas con * aquellos nodos que aún no se encuentran desplegados.

➤ **Amazon Cloud**

Tabla 5-2: Regiones Amazon Cloud [17]

Norteamérica	Europa	Asia	América del Sur	Australia
Ashbum	Amsterdam,	Chennai, India	São Paulo, Brasil	Melbourne
Atlanta	Berlín, Alemania	Hong Kong,	Río de Janeiro	Sídney
Chicago	Dublín, Irlanda	Manila, Filipinas		
Dallas	Fráncfort,	Nueva Delhi,		
Hayward	Londres,	Osaka, Japón		
Jacksonville	Madrid, España	Seúl, Corea Sur		
Los Ángeles	Marsella,	Singapur		
Miami	Milan, Italia	Taipei, Taiwán		
Minneapolis	Múnich,	Tokio, Japón		
Nueva York	Praga,			
Palo Alto	Estocolmo,			
Filadelfia	Viena, Austria			
San José	Varsovia,			
Seattle	Zúrich, Suiza			
South Bend				
St. Louis				
Montreal				
Toronto				

5.3 Accelerated Mobile Pages

Accelerated Mobile Pages AMP, es un sistema que contiene una serie de normas desarrolladas y promovidas por Google con la intención de mejorar la experiencia del usuario que accede a páginas web a través de sus dispositivos móviles disminuyendo drásticamente la velocidad de carga y dividiendo por 10 la cantidad de datos requeridos para cargar la misma información. Pone a disposición de los programadores y sitios web una plataforma distinta a lo conocido en las búsquedas.

Mediante AMP, es posible aumentar la visibilidad y mejorar la experiencia de usuario en páginas web con contenidos puntuales, sencillos y muy concretos. Así pues, está pensado para webs que pueden desarrollar una plantilla de contenido que puede mostrar información de manera dinámica. En la actualidad, algunos de los dominios que emplean AMP de manera oficial son los siguientes [18]:

- Marca
- WOMP Mobile
- Terra
- Milestone
- Teads
- Hearst
- RelayMedia
- Plista
- Slate
- WIRED
- Gizmodo
- The Washington Post

Cuando un usuario realiza una búsqueda en Google, éste muestra una lista de resultados, entre los cuales pueden estar páginas AMP, y además, un carrusel de resultados con una previsualización del contenido para aquellos dominios cuya temática sea la información de la actualidad o noticiarios. En la siguiente imagen se muestra una búsqueda móvil con resultados AMP en formato carrusel horizontal:



Figura 5—3: Búsqueda AMP

La lista de noticias que se muestran en el carrusel son deslizables lateralmente, dando paso al resto de noticias indexadas. Una vez se accede a una de las noticias, se muestra un iframe de Google, es decir, la página web es cargada desde el propio navegador mediante un iframe y sin salirse de los resultados de búsqueda, de manera que al cerrar la página web, se vuelve a los resultados de la búsqueda realizada.

Como se puede observar en la imagen mostrada a continuación, la página web es mostrada dentro de la búsqueda (observar la URL). No solo eso, sino que además, dentro de la propia noticia, es posible acceder a otras noticias que aparecían en el carrusel sin volver a la lista de resultados, con tan solo deslizar lateralmente sobre la pantalla del teléfono móvil.



Figura 5—4: Iframe AMP

Lamentablemente, la posibilidad de aparecer en el carrusel esta, de momento, vetada para todas aquellas páginas web cuyo ámbito no sea el de información diaria.

6. ANÁLISIS DE ALTERNATIVAS

6.1 Objetivo del análisis

A continuación, se realiza un análisis de alternativas. La misión u objeto de dicho análisis es llevar a cabo un estudio profundo de las posibles alternativas que son potenciales soluciones a las problemáticas planteadas en este estudio. Gracias a este análisis, es posible valorar cada una de las posibilidades y determinar cuál de ellas es la óptima.

6.2 Alternativas

6.2.1 Infraestructura del servidor

En este apartado se analizan las diferentes combinaciones de tecnologías existentes para desplegar servicios web. Actualmente, las infraestructuras más empleadas son LAMP (Linux Apache MySQL PHP) y MEAN (MongoDB ExpressJS AngularJS NodeJS), por lo que su estudio es objeto de interés para este proyecto.

6.2.1.1 LAMP

LAMP o Linux Apache MySQL PHP es el conjunto de tecnologías empleadas en la definición de servidores web. La combinación existente entre estas tecnologías es el resultado de la suma entre el bajo coste de las herramientas que hacen uso de las tecnologías y su carácter open source.

La arquitectura que emplea este sistema es la siguiente:

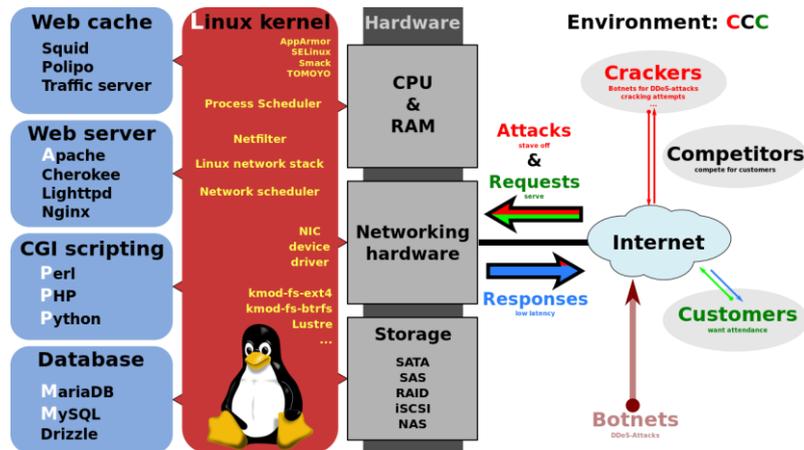


Figura 6—1: Esquema LAMP [7]

En primer lugar, se disponen las bases de datos de las cuales se obtiene la información necesaria para servir contenido o proporcionar algún servicio. Para ello, emplea MySQL o MariaDB.

Lo siguiente es el código CGI scripting o APIs que permite gestionar los datos almacenados en las bases de datos. El código de las APIs se implementa fundamentalmente en PHP, pero también puede emplearse Perl y Python.

A continuación, se dispone un servidor, concretamente Apache, que se encarga de proporcionar a los clientes un medio de comunicación con los recursos (el front end), de recibir las solicitudes y hacer las llamadas pertinentes al código de las APIs existentes para obtener la información necesaria para responder las solicitudes.

Finalmente, se puede disponer de servicios de caché o de gestión de carga que mejoren el rendimiento del tráfico a la vez que disminuyen la carga de trabajo sobre los servidores y bases de datos.

Toda esta infraestructura, es desplegada sobre un sistema operativo Linux, que está disponible para ser accedido como servidor web.

6.2.1.2 MEAN

MEAN o MongoDB ExpressJS AngularJS NodeJS es, al igual que LAMP un conjunto de tecnologías que permiten el desarrollo de sistemas de servicios web.

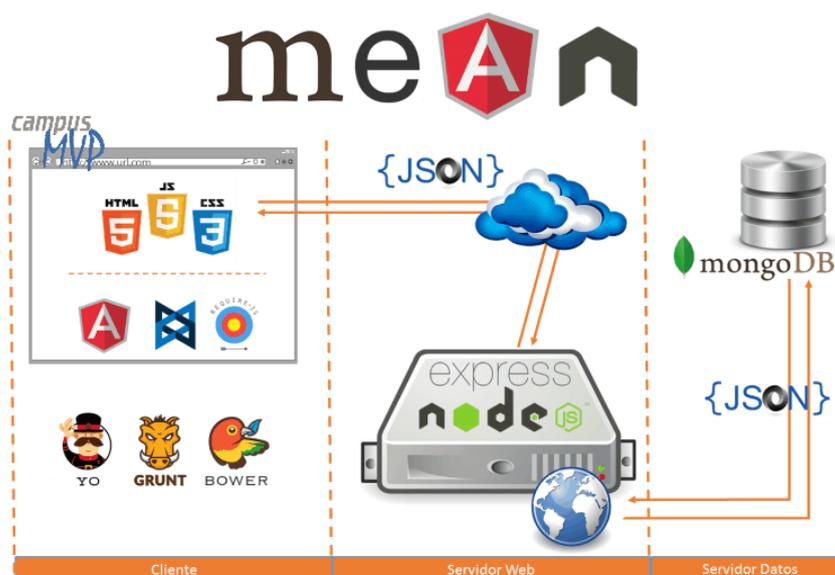


Figura 6—2: Esquema MEAN [7]

En primer lugar, las bases de datos son almacenadas mediante el sistema NoSQL MongoDB. Este sistema almacena los datos en estructuras definidas con notación JSON, lo cual permite una gestión muy eficiente de los datos almacenados. Además, una de sus grandes características es que, a pesar de que la cantidad de información crezca, su funcionamiento no se ve afectado.

Por otro lado, la tecnología ExpressJS es un módulo de NodeJS que funciona sobre la misma plataforma NodeJS. Hace las veces del CGI scripting existente en LAMP, es decir, es el soporte para las aplicaciones web que se encarga de gestionar la información almacenada bajo la solicitud del nivel superior, en este caso AngularJS.

AngularJS es un framework que se emplea para ser el front end del servicio, al igual que el código HTML5/PHP empleado en LAMP. Su gran

ventaja radica en la facilidad con la que pueden aplicársele librerías con funcionalidades en el front end, es decir, las vistas.

Por último, NodeJS es la plataforma sobre la que se encuentran desplegados los anteriores puntos cuyo funcionamiento está basado en JavaScript.

6.2.2 Tecnologías web optimizadas para terminales móviles

En la actualidad, el diseño de una página web adaptativa o *responsive* favorece la experiencia de usuario en las páginas web debido a que es la que se encarga de hacer que el sitio web sea lo más intuitivo y amigable dadas cualesquiera que sean las características del dispositivo en el que se hace uso de ellas.

El diseño adaptativo es una metodología de diseño y desarrollo que tiene como objetivo adaptar el contenido y apariencia de las páginas web dadas las exigencias del mercado actuales. Esto se debe a que un sitio web es ahora accedido desde teléfonos móviles, tablets, ordenadores e incluso televisores y no en todos ellos un único diseño se muestra correctamente.

Gracias al diseño *responsive*, con una única versión del HTML y CSS se puede adaptar una página web a los diferentes dispositivos. Sin embargo, existen tecnologías que, al aplicarlas, permiten a los navegadores modificar la organización del contenido de manera sencilla. Es el caso de tecnologías como bootstrap, que permite al desarrollador -entre otras ventajas- adaptar el sitio web en tiempo real, por ejemplo, si se modifica el tamaño de ventana del navegador en que se visualiza la página web.

Por otro lado, a partir de 2016, Google está impulsando la tecnología Accelerated Mobile Pages que, aun estando sin madurar, tiene un grandísimo potencial y está siendo impulsada por el propio Google.

En este proyecto se presenta la comparativa del funcionamiento existente entre las páginas web adaptativas realizadas empleando HTML5+

CSS + bootstrap, tanto en cuanto a la experiencia del usuario se refiere, funcionalidad, tiempos de carga, etc.

6.2.2.1 Accelerated Mobile Pages

Accelerated Mobile Pages AMP es una nueva forma de compilar las páginas web estáticas que requieren una rápida presentación y que, en ocasiones, contienen múltiples vistas multimedia como pueden ser fotografías, vídeos, gifs, etc. Es una iniciativa promovida por Google a fin de que el contenido cargue más rápido. Esto se consigue, grosso modo, implementando una enorme caché que almacena los contenidos listos para ser servidos previo procesamiento y optimización. Dentro de dicho procesamiento, se tiene en cuenta, no solo la velocidad de carga de las páginas, sino que también la correcta visualización de las mismas en dispositivos móviles. AMP consta de tres partes bien diferenciadas en su arquitectura de funcionamiento:

- Por un lado AMP HTML: Es básicamente HTML pero con restricciones que le permiten aumentar el rendimiento y que, junto a determinadas extensiones permiten aportar funcionalidades adicionales al HTML básico. Este permite generar contenido web de una forma más ligera que con HTML.
- Por otra parte, AMP JS es el encargado de garantizar la rápida presentación de las páginas AMP HTML del punto anterior. Al igual que el lenguaje HTML se encuentra restringido, con JavaScript ocurre lo mismo, por lo que se hace necesaria una librería específica Javascript (AMP JS).
- Finalmente, y de forma opcional, es posible configurar una caché mediante Google AMP Caché, que se encarga de servir las páginas AMP HTML. Esta caché funciona a modo de CDN y como tal, su punto de acceso es un proxy inverso. Su trabajo consiste en recoger las páginas AMP HTML configuradas con este servicio y mejorar su rendimiento. Si se emplea esta configuración, todos los ficheros javascript e imágenes que contienen este tipo de web deben estar contenidas en el propio origen o servidor de caché, a fin de reducir el máximo la dependencia de otros servidores no optimizados y así aumentar la rapidez del servicio.

Además, la caché posee un sistema de validación que se encarga de verificar el correcto funcionamiento de la página y de que no dependa de recursos externos. Si la validación no es correcta, dicha caché no procede a almacenar y servir dichos contenidos.

Si se introduce en la web <https://validator.ampproject.org> la url de la página web que se desee verificar manualmente, se puede observar que, cuando todo es correcto y se cumplen las normas, el estatus de la página es aceptado y pasa a estar disponible en el caché de Google.

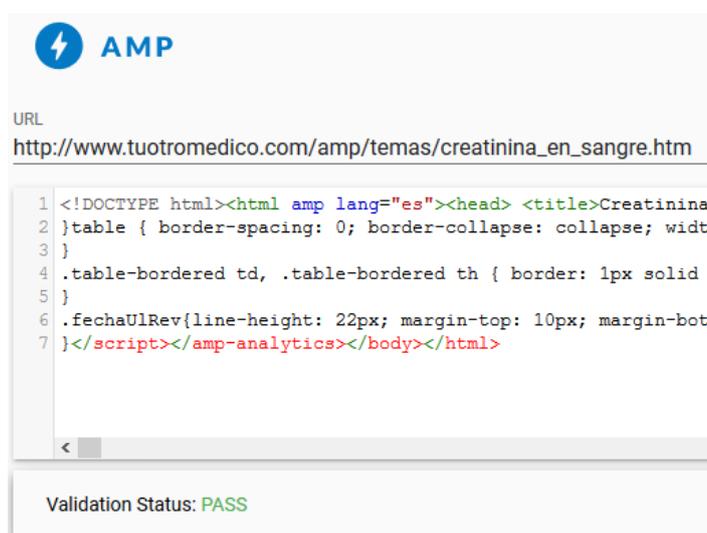


Figura 6—3: Valicación URL AMP

Una vez que la web esta validada Google la cachea junto a todos los recursos requeridos y la muestra. A continuación, se muestra un ejemplo de la página web relativa al dominio a trabajar durante este proyecto.



Ilustración 6—1: Tuotromedico AMP

6.2.2.1.1 Fundamentos de AMP

Para conseguir que las páginas sean lo más óptimas posible se emplean técnicas de implementación Web Performance Optimization (WPO). Estas técnicas pretenden optimizar el rendimiento y los tiempos de carga de las páginas web. Entre las técnicas empleadas, se pueden observar las siguientes:

- Los recursos que emplea la página web son descargados posteriormente al código fuente de la misma. De esta manera, se consigue presentar la página en el menor tiempo posible y que el usuario pueda observarla mientras todos sus elementos terminan de ser descargados. Para que AMP pueda optimizar la carga de elementos pesados, es necesario que éstos especifiquen su tamaño o peso.
- Como ya se ha mencionado en las características de AMP, las peticiones externas están prohibidas y para poder incluir vídeos u otros elementos multimedia, estos deben estar contenidos en el servidor local.
- Se emplea JavaScript asíncrono específico de AMP, revisado previamente por Google y que tiene funcionalidades muy específicas como, por ejemplo, la monitorización de Google Analytics. Es necesario tener muy presente que el código JS bloquea la construcción de la web durante el tiempo en que está siendo ejecutado.

La idea de adaptar los sitios web a los diferentes dispositivos en función del tamaño de las pantallas que lo visualizan es buena, pero posee determinadas características que hacen muy difícil poder adaptar todos los contenidos a todos los dispositivos de la manera más adecuada. Ello se debe a que todas las páginas web existentes hasta el momento del boom de los terminales móviles no están diseñadas con el propósito de adaptarse y por tanto no existe una metodología “mobile-first design”.

6.2.2.1.2 Configuración del indexado AMP

Al implementar AMP en las webs deseadas se ha de tener en cuenta que las páginas AMP deben ser copia de las HTML habituales con ligeras variaciones y etiquetados.

Para establecer la relación entre las versiones HTML y las AMP HTML se debe añadir en todas las páginas que dispongan de versión AMP la siguiente etiqueta:

```
<link rel="amphtml" href="">
```

Dónde el href es la URL de la versión AMP. De este modo, se obtiene lo siguiente:

```
<link rel = "amphtml" href=" <a href='web/amp/' target =  
'_blank'>web/amp</a>" />
```

Del mismo modo, la página AMP HTML debe incluir el enlace a la página de escritorio, de la siguiente manera:

```
<link rel ="canonical" href ="<a href="http://web"  
target="_blank">http://web</a>" />
```

Esto último se ha de realizar para que Google no aplique penalización por contenido duplicado haciendo que la nueva URL no sea indexada.

6.2.2.1.3 AMP en los buscadores

Google hace uso de ventanas horizontales que muestran resultados de páginas que poseen versiones programadas en AMP y que se encuentran validadas. Si se hace uso de uno de los links que Google pone a disposición del usuario, la página que se sirve está contenida en una de las cachés del CDN de Google. Para ello hace uso de un *iframe*, de manera que la página es cargada sin salir de la propia página de Google, pudiendo ser cerrada y, de esta manera, volviendo a los resultados obtenidos en la búsqueda previamente realizada.

Sin embargo, existen determinadas desventajas que pueden no gustar tanto a los usuarios. Entre estas desventajas, se puede observar la imposibilidad de controlar el medio o de realizar un control del tráfico cursado en las páginas web por parte de los servidores que alojan las versiones completas. Además, la limitación existente en cuanto al uso de código JavaScript limita, dependiendo de los casos, algunas funcionalidades existentes.

6.2.2.2 Otros frameworks web

Bootstrap es el framework de código abierto más popular en la actualidad para el diseño de sitios web responsive. Se contempla esta opción frente a otros frameworks debido a su implementación actual en la página web a tratar en el presente proyecto. Recoge un conjunto de metodologías y herramientas que permiten diseñar sitios web con carácter adaptativo y, al igual que en otros frameworks, es posible emplear diferentes tipografías, botones, menús y otros elementos mediante el uso de diferentes etiquetas en el código HTML y sus correspondientes en el CSS.

En la actualidad, bootstrap es soportado por la mayor parte de los navegadores existentes, tanto de escritorio como de dispositivos móviles.

	Chrome	Firefox	Internet Explorer	Microsoft Edge	Opera	Safari
Mac	Supported	Supported	N/A	N/A	Supported	Supported
Windows	Supported	Supported	Supported, IE10+	Supported	Supported	Not supported

Figura 6—4: Bootstrap Desktop Browsers [19]

	Chrome	Firefox	Safari	Android Browser & WebView	Microsoft Edge
Android	Supported	Supported	N/A	Android v5.0+ supported	N/A
iOS	Supported	Supported	Supported	N/A	N/A
Windows 10 Mobile	N/A	N/A	N/A	N/A	Supported

Figura 6—5: Bootstrap Mobile browsers

6.2.2.2.1 Fundamentos de Bootstrap

Para hacer uso de Bootstrap es necesario disponer tanto del código CSS como del código JavaScript que permiten emplear este framework en HTML5. Bootstrap dispone de un conjunto de plantillas en formato LESS, consistente en un lenguaje de estilos que puede ser compilado sobre archivos CSS, siendo posible llevar a cabo su compilación **en tiempo real** por el propio navegador. El uso de estas plantillas se realiza como si de una plantilla CSS normal se tratara, siendo de este modo, implementada en el código HTML5 mediante etiquetas `<link>`.

La característica que lo hace tan poderoso es la capacidad de ser compilado en tiempo real, siendo posible adaptar la plantilla sobre la que se muestra el código a la forma y tamaño de la pantalla en que se muestra el contenido del sitio web.

6.2.3 Servidor de contenido

En este apartado se analiza si es mejor crear un nodo de distribución de contenidos propio o contratar este servicio a un proveedor con red de distribución. En el primer caso, se estudian las posibilidades para llevar a cabo

el despliegue del nodo que son, en primer lugar, la virtualización de un sistema y, por otro lado, contratar una máquina física propia.

En primer lugar, dentro de las diferentes opciones para el despliegue y soporte de un Nodo Distribuidor de Contenido se encuentran las siguientes posibilidades:

- Servidor de contenidos virtualizado y con reserva de recursos (VPS).
- Servidor de contenidos exclusivamente dedicado al servicio, es decir, contratación de un servicio de *hosting dedicado*.

La contratación de un Virtual Private Server VPS consiste en contratar un servicio de virtualización, o bien mediante las tradicionales máquinas virtuales o bien mediante nuevas plataformas de virtualización como Docker. Una máquina virtual es fácilmente contractable y se puede realizar a medida y, según las necesidades, contratar una determinada capacidad de almacenamiento, memoria RAM, capacidad de procesamiento, ancho de banda, etc. Todos estos recursos son dispuestos por una máquina sobre la que se pueden encontrar diferentes sistemas virtualizados de manera simultánea. Este tipo de hosting permite a quien lo contrata tener acceso de superusuario a la máquina virtual, siendo posible desplegar sobre ella todos los servicios que la máquina sea capaz de soportar con los recursos de los que dispone.

La contratación de un servidor dedicado consiste en la adquisición exclusiva de una máquina en la cual se encuentran desplegados únicamente los servicios que el cliente desea y sobre el entorno elegido. Este tipo de servicios son contratados cuando se requiere la seguridad de que los recursos de la máquina son exclusivos para el servicio que se desea desplegar.

Por otro lado, como se menciona en el anterior punto, en la actualidad se dispone de múltiples proveedores de contenido y, en este apartado, se analizan aquellos que se consideran más extendidos: Amazon Cloud, Google Cloud, Akamai y CloudFlare. Sin embargo, ninguno de ellos posee un NDC próximo a todas las localizaciones mundiales como, por ejemplo, México. Por

este motivo se decide que, para este proyecto, lo más óptimo es desplegar un NDC propio próximo a los nuevos clientes en México.

6.3 Criterios de selección

6.3.1 Infraestructura del servidor

En este apartado, se definen los criterios a emplear en la selección de la infraestructura sobre la cual se despliega el servicio web, ya sea LAMP o MEAN.

- Escalabilidad. La capacidad de que el sistema crezca sin la necesidad de modificar su funcionamiento a la vez que su rendimiento se mantiene. Pondera un 25% puesto que es un aspecto importante. Sin embargo, se debe considerar si el aumento del tamaño afecta de gran manera al funcionamiento del sistema en cualquiera de las dos infraestructuras.
- Reutilización de código. Este aspecto es de gran importancia, centrado en uso de recursos para el front end. Pondera un 15% ya que actualmente el sistema se encuentra altamente implementado y esto no reviste mayor importancia.
- Rendimiento en la gestión de datos. Este aspecto evalúa la velocidad con la que los sistemas son capaces de acceder a la información y/o modificarla. Pondera un 60% ya que este aspecto es fundamental para el funcionamiento del servicio. Debe tenerse en cuenta, que el sistema consiste prácticamente en búsquedas.

6.3.2 Frameworks web para terminales móviles

En este apartado, se definen los criterios que determinan el framework a emplear para llevar a cabo el proyecto.

- Requerimientos al servidor: Este aspecto contempla la carga que supone cada página a devolver al usuario que la solicita. En términos numéricos, hace referencia a la cantidad de información a transferir, el número de peticiones a realizar para completar la carga de la página web. Este aspecto pondera un 30%

- Capacidad de adaptación. Este aspecto tiene en cuenta la capacidad del framework de adaptarse al tamaño de la pantalla en que se visualiza el contenido tanto de forma estática como dinámica. Estas consideraciones pueden influir en la experiencia de usuario. Pondera un 20%.
- Tiempo de carga de la página web. Es el aspecto más relevante en cuanto a su influencia en el posicionamiento del sitio web se refiere. Por este motivo, pondera un 50%.

6.3.3 Servidor de contenido

En este apartado, se definen los criterios a emplear en la selección del tipo de servicio a desplegar sobre el cual se despliega el Nodo Distribuidor de Contenidos.

- Seguridad frente ataques. Este aspecto valora la posibilidad de que el servicio sea atacado. Es fundamental evitar en la medida de lo posible ataques sobre el servidor que afecten de cualquier manera a los usuarios. La ponderación de este punto es de un 20%
- Rendimiento de la máquina. Este aspecto recoge la importancia de un buen rendimiento en la máquina que despliega el servicio. La ponderación de este punto es de un 30%
- Escalabilidad. Este aspecto recoge la importancia de poder escalar el sistema y ampliarlo en caso de ser necesario. Este aspecto pondera un 20%.
- Fiabilidad de la información. Este aspecto recoge la importancia de poseer redundancia y duplicidad de la información dependiendo del tipo de servicio contratado e independientemente del entorno en que se desarrolla, en este caso, una red de distribución de contenidos. Este aspecto pondera un 20%.
- Coste. Por último, se valora el coste de las opciones existentes. Este aspecto pondera un 10%.

6.4 Selección de la solución

6.4.1 Infraestructura del servidor

En este apartado se selecciona la solución de la infraestructura a utilizar. Las principales diferencias entre las infraestructuras LAMP y MEAN son las siguientes:

- NodeJS permite escribir aplicaciones para el servidor en lenguaje JavaScript, haciendo posible que el sencillo lenguaje JavaScript, pueda ser empleado no solo desde el lado del cliente. Además, debido a su arquitectura no bloqueante es más rápido y escalable que lo que lo era LAMP.
- Por otro lado AngularJS, es capaz de crear front ends de gran complejidad de manera sencilla haciendo uso de librerías reutilizables, lo cual es una gran ventaja.
- MongoDB es una base de datos cuya principal característica es no ser SQL, es decir, no se relacional. Es un formato que permite almacenar información o documentos en formato JSON. Dado que JSON es un formato fácilmente entendible por JavaScript tanto en el lado del cliente como en el del servidor, la necesidad de librerías externas de serialización y deserialización, es decir, de código externo, desaparece.
- Debido a que MongoDB es una base de datos no relacional, la velocidad de acceso a la información depende de qué tipo de contenido se desee cargar. Por ejemplo, la obtención de información de múltiples tablas acerca de un usuario en MySQL es más lenta en comparación con la obtención de toda esa información de un solo documento o estructura JSON.

A continuación, se muestra una comparación en cuanto a la velocidad de acceso e inserción entre MongoDB y MySQL medidas en milisegundos para 10, 100, 1000 y 10.000 búsquedas en inserciones [20]:

Tabla 6-1: MongoDB VS MySQL - Resultados

Datos a tratar. Unidades	Inserción				Búsqueda			
	10	100	1000	10000	10	100	1000	10000
SQL (ms)	1702	8171	94813	924695	1	10	14	144
MongoDB (ms)	46	167	1013	9956	2	60	677	4539

Como se puede observar en la tabla anterior, los tiempos de búsqueda de información en bases de datos relacionales son mucho más rápidas que en las no relacionales. Además, se verifica que la búsqueda de información bajo tablas relacionales es extremadamente escalable en comparación con las no relacionales. Por tanto, a continuación, se muestra la tabla de puntuaciones sobre las que este documento se basa para seleccionar la infraestructura más adecuada para el proyecto:

Tabla 6-2: MongoDB VS SQL. Comparativa

	Escalabilidad -25%-	Reutilización -15%-	Rendimiento -60%-	Puntuación
SQL	10	5	9	8.65
MongoDB	5	10	6	6.35

En vista de los resultados obtenidos, se considera más adecuado el uso de la infraestructura LAMP con SQL debido fundamentalmente a que los tiempos de búsqueda y escalabilidad en la casuística de uso analizada arrojan mucho mejores resultados que los obtenidos con MEAN y MongoDB.

6.4.2 Frameworks web

A continuación, se evalúan los criterios citados anteriormente a fin de seleccionar la opción óptima.

- **Requerimientos al servidor:** En este punto AMP resulta ser superior a bootstrap puesto que AMP permite servir contenido con mayor agilidad debido a la optimización del código de las páginas web, consigue disminuir la

cantidad de información transferida gracias a su carácter asíncrono por el cual solo se transfiere lo que el usuario demanda.

- Capacidad de adaptación. En cuanto a este aspecto se puede considerar que existe igualdad. Bootstrap es capaz de adaptarse a los tamaños de pantalla según ha sido configurado su fichero de estilos CSS, tanto dispositivos móviles como ordenadores. Por su parte, AMP, es empleada únicamente en terminales móviles ya que, si se accede a una web mediante un ordenador es cargada la versión de escritorio o “canonical”, que puede estar desarrollada con cualquier framework como pudiera ser el mismo Bootstrap.
- Tiempo de carga de la página web. En este aspecto no cabe la menor duda, AMP vence debido a su optimización de código, a que requiere menor tiempo de procesado por parte del cliente web, y además, se encuentra cacheada por Google.

Tabla 6-3: AMP vs Bootstrap. Comparativa

	Requisitos -30%-	Adaptación -20%-	Rendimiento -50%-	Puntuación
AMP	10	10	10	10
Bootstrap	5	10	6	6.5

Por tanto, se concluye que la tecnología óptima para el desarrollo de páginas web es Accelerated Mobile Pages AMP.

6.4.3 Servidor de contenido

En este apartado, se selecciona la posibilidad más óptima para el despliegue de un nodo de distribución de contenidos.

- Seguridad frente ataques. Este aspecto depende de la seguridad de la que se desee dotar al servidor. En cualquier caso, esta seguridad depende en primer lugar de la empresa a la que se le contrata la máquina virtual o física.
- Rendimiento de la máquina. En cualquiera de los casos, virtual o física, este rendimiento depende de los recursos contratados. Sin embargo, si la máquina es virtualizada, es posible que otra máquina ajena, alojada en el mismo host,

se encuentre realizando un proceso pesado que impida garantizar el rendimiento del NDC en determinadas ocasiones.

Como se puede ver en el siguiente gráfico, el segundo país que más sesiones genera es México, y es por ello, que se requiere un mínimo de garantías para el servicio.

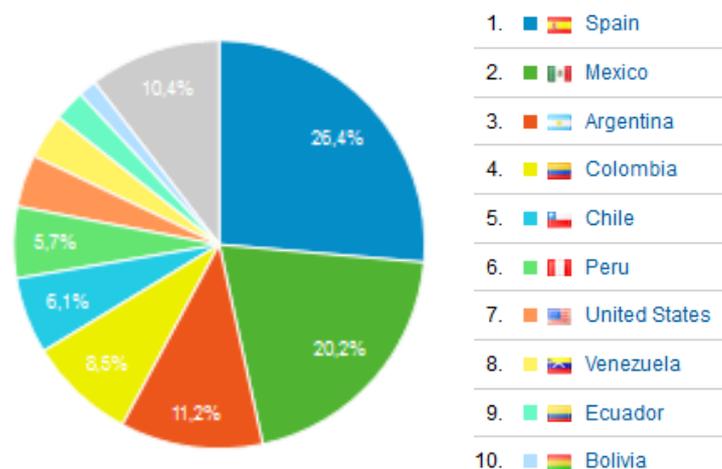


Figura 6—6: Sesiones por país [1]

- Escalabilidad. Como se menciona en el apartado de criterios en cuanto a este apartado, en primera instancia puede parecer que un sistema virtualizado tiene ventaja debido a su capacidad de adaptarse, eso si, dentro del rango que ofrezca el proveedor de hosting. Sin embargo, para un servicio web cuyo número de clientes se mantiene estable, puede que esta ventaja no pueda ser aprovechada.
- Fiabilidad de la información. En ambos casos, los nodos pueden funcionar con un sistema de almacenamiento en la nube.
- Coste. Como se puede ver en el apartado de criterios, resulta mucho más económico disponer de un sistema virtualizado que contratar una máquina completa.

Tabla 6—4: VPS vs Host Físico. Comparativa

	Rendimiento -20%-	Seguridad -30%-	Escalabilidad -20%-	Fiabilidad -20%-	Coste- 10%-	Puntuación
VPS	6	10	8	8	8	10
Host físico	10	10	4	8	4	6.5

En vista de los resultados mostrados en la tabla superior, se concluye que no es necesaria una máquina completa para ofrecer servicios web ya que su principal ventaja es el rendimiento y éste no es tan importante en el mencionado contexto.

7. METODOLOGÍA

En este apartado, se detalla el procedimiento que se sigue en este proyecto para mejorar el rendimiento de las páginas web una vez seleccionadas las mejores opciones del análisis de alternativas y, por tanto, conociendo las tecnologías y frameworks a emplear.

Para llevar a cabo lo mencionado en el anterior párrafo, se muestra el ciclo de vida del proyecto y el consiguiente modelo a seguir, a saber, un modelo lineal, en espiral, incremental o, en V.

7.1 Modelo de desarrollo

Para el desarrollo de este proyecto, en cuanto al apartado software se sigue un modelo en V dado que se trata de un proyecto de pequeñas dimensiones y cuyo equipo de trabajo se ve reducido a tan solo 2 personas. Además, permite mostrar de una manera clara y ordenada el procedimiento en curso y sus resultados al ser un modelo triangular a cualquier nivel. El esquema a seguir es el siguiente:

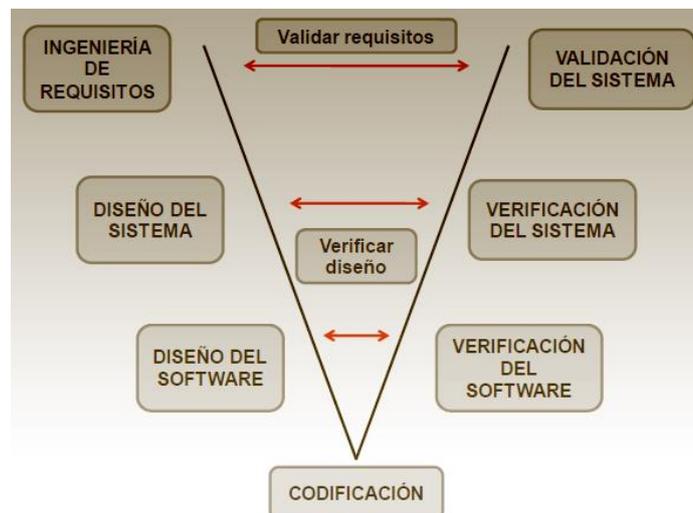


Ilustración 7—1: Modelo de desarrollo en V [23]

Posteriormente, a la conclusión de la validación del sistema, una se verifica que el sistema software desarrollado con los nuevos conjuntos de

normas y tecnologías funciona correctamente, se somete a la minificación del código servido por el servidor a fin de que los navegadores requieran un menor tiempo de procesamiento de dicho código y puedan mostrar más velozmente los contenidos.

Por otra parte, al final de este apartado de metodología, se encuentra el procedimiento a seguir para el despliegue del Nodo de Distribución de Contenidos en México.

7.1.1 Definición de requisitos del sistema

Los requerimientos de las técnicas aplicadas en este estudio se basan exclusivamente en mantener las funcionalidades existentes previas al nuevo diseño que se desarrolla en este estudio aplicando las acciones que se consideren oportunas para optimizar el rendimiento y, de este modo, aumentar los beneficios. Así pues, los requisitos para el dominio a tratar son:

- Acceso a los mismos menús y opciones aplicando las consignas AMP como, por ejemplo, la supresión de todo código JavaScript.
- Desarrollo de plantilla HTML5-AMP que tenga la capacidad de obtener listados de preguntas almacenadas en bases de datos y mostrarlas a los usuarios con una interfaz amigable y cómoda.
- Desarrollo de plantilla HTML5-AMP que tenga la capacidad de mostrar el directorio médico existente en las bases de datos y mostrarlo a los usuarios con una interfaz amigable y cómoda.
- Integración absoluta de la publicidad empleando las opciones proporcionadas por HTML5-AMP.
- Habiendo sido completados los requisitos anteriores, el rendimiento de la web debe ser superior y la posición media de la misma en los buscadores debe prosperar.

7.1.2 Diseño del sistema

El funcionamiento del sistema a bajo nivel se basa en una arquitectura de tablas relacionales desplegadas previamente al estudio que se muestra en

este documento. Las tablas que afectan al desarrollo propuesto son las siguientes:

- Directorio: Es la tabla que contiene la información de los médicos que se muestran a través de la plantilla HTML5-AMP desarrollada para tal objeto.
- Preguntas: Es una tabla que contiene parejas de Preguntas-Respuestas obtenidas a lo largo de los años de las preguntas realizadas por los usuarios del dominio y las respuestas dadas por profesionales médicos que forman parte del directorio.

El acceso a estas tablas permite cargar los contenidos sobre las plantillas, desarrolladas de forma dinámica. La arquitectura a emplear es la siguiente:

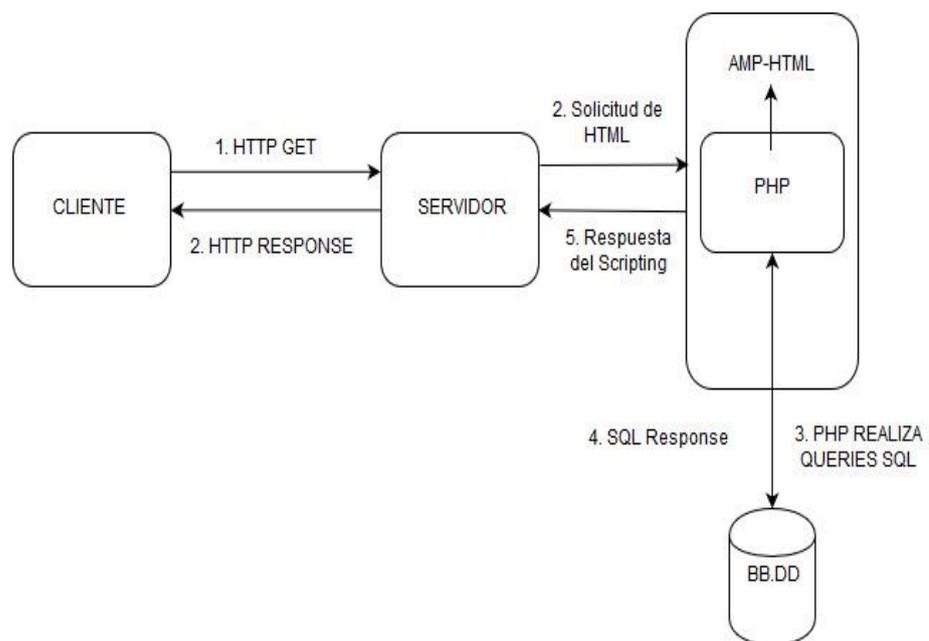


Ilustración 7—2: Diagrama SSC

Debido a las restricciones de AMP, no es posible emplear tecnologías como JQuery y actualizar el contenido de manera dinámica, por lo que es necesario recargar la página para la carga de cualquier contenido nuevo.

7.1.3 Diseño del software

En esta etapa del proyecto se pretende optimizar la calidad del código que se desarrolla en la siguiente fase. Para ello, es necesario llevar a cabo un correcto diseño que permita minimizar los posibles fallos del sistema y, en caso de suceder alguno, que pueda ser subsanado a la mayor brevedad posible, facilitando de esta manera la tarea de la etapa de programación. Las pautas a seguir son las siguientes:

- Al igual que en cualquier desarrollo software, es necesario pensar en la concurrencia del código, es decir, pensar en la posibilidad de que varios programadores trabajen sobre un mismo código. Los aspectos que se debe establecer en este apartado son:
 - Estructuración común de los apartados y funciones.
 - El código debe ser debidamente comentado.
 - Debe eliminarse todo código empleado para la realización de pruebas, es decir, el código debe mantenerse limpio.
- Reducir al mínimo la comunicación con el servidor. En este apartado se pueden diferenciar dos comunicaciones con el servidor:
 - Solicitud de información y publicidad: Las peticiones de información deben ser las menores posibles a fin de disminuir los tiempos de carga.
 - La comunicación con la base de datos desde el código, situado en el lado del servidor debe reducirse de manera que es necesario un equilibrio entre la cantidad de información necesaria y la solicitada. Para ello, las plantillas que muestran listas constan de páginas que muestran resultados sucesivos.
- La distribución del servicio se encuentra desplegada con anterioridad al proyecto aquí mostrado. El servicio sigue una arquitectura Server Side Scripting, es decir, el cliente posee una interfaz a través de la cual se envían solicitudes al servidor, el cual se encarga de realizar las tareas necesarias para obtener los datos y devolverlos al cliente cuyo navegador vuelva a mostrar mediante la interfaz de usuario (en el caso que atañe al proyecto, las plantillas), la información requerida.

A continuación, se muestra la parte del software en la cual se han de cumplir las mencionadas pautas. Esta parte se centra en la codificación HTML-AMP junto al código PHP que se encarga de relacionarse con las BB.DD.

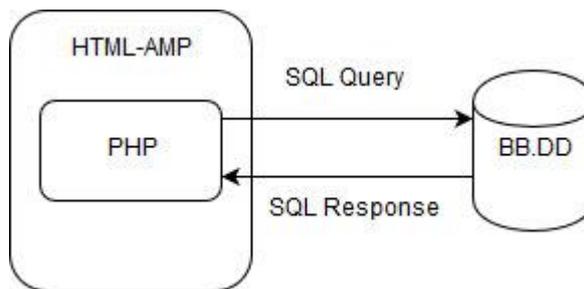


Ilustración 7—3: Comunicación BB.DD con PHP

7.1.4 Codificación

En este apartado se lleva a cabo la codificación de las plantillas que muestran contenidos de forma dinámica. En primer lugar, se llevan a cabo los menús y opciones que se encuentran incluidas en cualquiera de las páginas que siguen. A continuación, se desarrollan las plantillas para mostrar el resto de contenidos descritos en los requisitos, a saber, el directorio médico y las preguntas.

7.1.5 Verificación del software

Tanto durante el desarrollo como a la conclusión de cada una de las plantillas, se realizan pruebas unitarias que por un lado verifican su correcto funcionamiento a nivel software y que cada una de las plantillas cumple con el objetivo de mantener la funcionalidad previa a la implementación de HTML5-AMP. Por otro lado, se realizan pruebas de tiempos de carga con pequeñas muestras de contenido a fin de verificar que las modificaciones realizadas tienen un resultado positivo.

7.1.6 Verificación del sistema

En este punto, todo el código desarrollado y verificado en los anteriores puntos sustituye al previo HTML5+JavaScript. Se realizan pruebas que verifican lo que las pruebas unitarias muestran: una gran mejora en el rendimiento del dominio con los nuevos pragmas. Dichas pruebas, poseen un apartado propio a continuación, el 7.3.

7.1.7 Validación del sistema

En este apartado se verifican los siguientes puntos:

- Que el acceso a los menús y opciones aplicando las consignas AMP, funcionan de manera exactamente igual que con la anterior versión.
- Que la plantilla HTML5-AMP para mostrar preguntas tiene la capacidad de obtener listados de preguntas almacenadas en bases de datos y mostrarlas a los usuarios con una interfaz amigable y cómoda.
- Que la plantilla HTML5-AMP para mostrar el directorio médico existente en las bases de datos, es capaz de mostrarlo a los usuarios con una interfaz amigable y cómoda.
- Que la integración de la publicidad es absoluta empleando las opciones proporcionadas por HTML5-AMP.
- Que, habiendo sido completados los requisitos anteriores, el rendimiento de la web es superior. Que la posición media en los resultados de búsqueda no es algo verificable a corto plazo. No obstante, las pautas para la optimización del posicionamiento son aplicadas y se encuentran verificadas, lo cual hace pensar que la posición media mejorará.

Una vez dadas por cumplidas estas premisas se considera finalizado el apartado de desarrollo software de este proyecto. A continuación, el servidor debe minificar las plantillas con las que sirve los contenidos.

7.2 Procedimiento de minificación

El procedimiento de minificación para este proyecto se lleva a cabo en el servidor de forma manual. Para llevar a cabo este procedimiento, se hace uso de un script desarrollado por el usuario *fizker* de la comunidad GitHub. Para instalarlo se ejecuta el siguiente comando:

```
npm install -g minifier
```

El proceso de minificación puede llevarse a cabo manualmente, o bien mediante la generación de un script con el objeto de ser llamado por el demonio Crontab periódicamente. Dado que el proceso en cuestión solamente es necesario cuando existe alguna modificación o actualización del código fuente de las plantillas que muestran el contenido, se decide que su ejecución sea llevada a cabo manualmente de la siguiente forma, desde la línea de comandos:

```
minify [--output path/to/put/file] path/to/file
```

El resultado de este proceso se puede ver en la siguiente imagen:



Ilustración 7—4: Minificación [22]

7.3 Resultados de la verificación del sistema

A continuación, se procede al análisis de los resultados y su comparación con los datos del estado anterior del servicio ofrecido mediante otros frameworks y que, según las pruebas unitarias, es más lento que la nueva implementación.

Para llevar a cabo la correcta comparación de manera que ésta sea lo suficientemente clara como para hacer ver al lector, de una forma concisa, rápida y ordenada, cuáles son los resultados, se procede al empleo de herramientas previamente mencionadas en el documento.

7.3.1 Google Analytics

El análisis más sencillo se realiza mediante la herramienta de gestión y control de dominios de Google. En primera instancia, se comparan los resultados obtenidos en función de la tecnología y en segunda instancia, la proximidad al centro de distribución de datos. Los resultados obtenidos a continuación, comparan el tiempo medio de carga de una página desarrollada en AMP y el tiempo medio de carga de una página basada en HTML con el framework bootstrap.

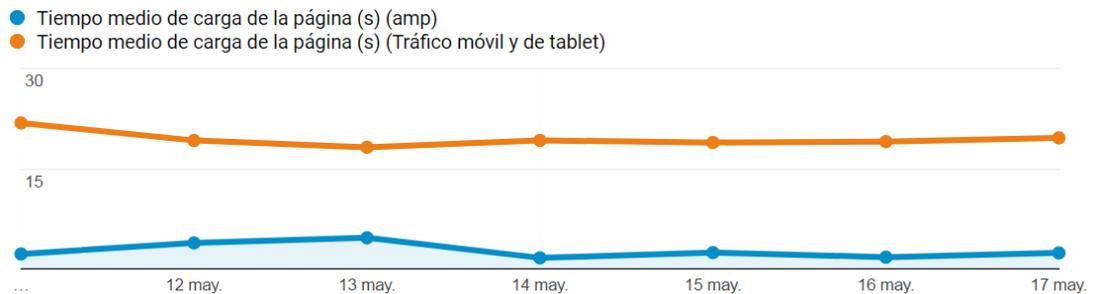


Ilustración 7—5: Tiempo de carga AMP VS No-AMP [1]

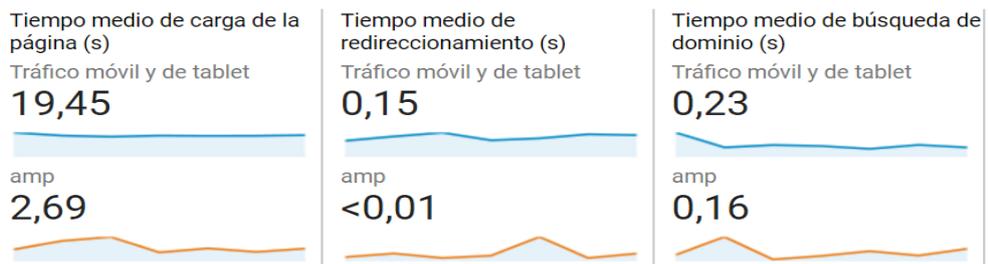


Ilustración 7—6: Resultados numéricos AMP V.S No-AMP [1]



Ilustración 7—7: Resultados numéricos AMP V.S No-AMP [1]

Como se puede observar en la ilustración anterior, existe una mejora altamente apreciable en cuanto al tiempo medio de carga de las páginas se refiere. De este modo el tiempo de redireccionamiento, los tiempos de búsqueda y los tiempos de respuesta son consecuentemente inferiores en AMP que en No-AMP. Por otra parte, el tiempo de descarga de la página se reduce considerablemente debido al menor tamaño de página y al menor número de peticiones necesarias para descargarla. Este último aspecto, se muestra más completo en los datos arrojados por las pruebas realizadas con la herramienta GTMetrix.

Por otra parte, en la siguiente gráfica se puede ver cómo el tiempo medio de descarga absoluto, es decir, de todas las páginas, incluidas tanto AMP, como No-AMP, para móviles y para escritorio, es muy dependiente de la proximidad al servidor de contenido.



Ilustración 7—8: Acceso próximo a NDC V.S Lejano a NDC [1]



Ilustración 7—9: Resultados relacionados proximidad al NDC [1]

Como se puede ver existen, de media, 5 segundos de diferencia entre los accesos a las páginas web desde México y los accesos desde otros países. Esta diferencia es muy elevada debido a la elevada cantidad de saltos de diferencia y distancia existentes entre el resto de países y el NDC situado en España. Es debido a esto último, que los tiempos medios de respuesta del servidor, los tiempos de descarga o tiempos de conexión son menores en el caso de los accesos de clientes desde España.

7.3.2 GTMetrix

A continuación, se muestran los resultados más relevantes obtenidos mediante GTMetrix (Resultados completos en Anexo I). Como se puede comprobar, estos resultados emplean un sistema de puntuaciones alfabeto-numéricas. Dichas puntuaciones clasifican con una letra cuyo máximo es F, y un número cuyo máximo es 100, las características analizadas. Siendo A y 100 las puntuaciones más altas y mejores, los resultados que se obtienen son los siguientes:

Tabla 7-1: Comparativa GTMetrix - I

	AMP	No-AMP
<i>Parseo de JavaScript</i>	A100	C77
<i>Compresión Gzip</i>	C79	D69
<i>Redirecciones externas</i>	A100	F0
<i>Tamaño de solicitud</i>	A100	C76
<i>Carga de fuentes asíncronas</i>	A100	B83

El parseo de JavaScript valora la velocidad con la que el navegador ejecuta la interpretación del código introducido mediante etiquetas <script> o inclusión de scripts en el código. Debido a que en AMP no existe dicho código, su puntuación es la máxima a la vez que considerablemente superior a la obtenida con la anterior metodología.

La compresión GZIP es una metodología que pretende ahorrar ancho de banda cuando se transmite una página web mediante la compresión de la misma. Debido a que ninguna de las metodologías implementadas emplea dicha compresión, las puntuaciones son relativamente bajas, siendo aún superior la puntuación obtenida por AMP debido al tamaño reducido de su código y su consecuente menor ancho de banda requerido.

En cuanto al número de redirecciones externas, está claro que debido a los principios de AMP acerca de la situación de los contenidos servidos, en su caso no existen este tipo de redirecciones en el momento de cargar la página. Por el contrario, la metodología No-AMP realiza las peticiones externas que

considera necesarias en el momento de obtener los banners de publicidad, lo cual supone una de las mayores ralentizaciones del servicio, de ahí que su puntuación sea la mínima posible.

El siguiente aspecto a analizar es el tamaño de la solicitud de información. El tamaño habitual para el tamaño de un segmento HTML ronda los 1500bytes. Entre su contenido se encuentran las cookies, cuyo tamaño máximo debe ser inferior a 1000bytes, mientras desde GTMetrix se recomienda no superar los 400bytes. Por tanto, comprimir el tamaño de la cabecera y reducir el uso de cookies permite reducir el tamaño requerido a la hora de hacer una solicitud. En AMP, debido a que las páginas validadas son servidas por Google desde su propio caché, estas cabeceras son de tamaño reducido y posibilita que las peticiones requieran de un único segmento. Sin embargo, en la metodología No-AMP, no siempre se respeta que la solicitud requiera de un único segmento, lo cual retarda más la respuesta por parte del servidor.

7.3.3 Waterfall

Esta prueba muestra la diferencia en cuanto al número de peticiones existentes al momento de cargar una página web. Como se puede observar, el número de peticiones de AMP es altamente inferior.

21 Request	495.8 KB	2.79s (onload: 1.52s)
201 Reques	3.2 MB	31.3s (onload: 13.77s)

Ilustración 7—10: Waterfall-4

Como se muestra en la Ilustración 7.12, para AMP se realizan 21 peticiones y para la página No-AMP se alcanzan las 201 peticiones, para cargar el mismo contenido.

7.4 Procedimiento de despliegue del NDC

El procedimiento para dar de alta un servicio consiste en poner al alcance de los usuarios la información que contienen las páginas web dinámicas. Así pues, es necesario crear las bases de datos con la información pertinente accesibles por al menos un usuario administrador. En el caso en que se sitúa el proyecto, el sistema de archivos se encuentra contenido inicialmente en el NDC central, que es el encargado de distribuirlo al otro NDC que se despliega en México. En ambos servidores, se dispone de las mismas bases de datos con los mismos o diferentes usuarios y contraseñas de acceso a la información. Por tanto, una vez se tiene la base de datos, en este caso particular basada en SQL, es necesaria la creación del usuario administrador de la siguiente manera:

```
CREATE USER usuario@localhost IDENTIFIED BY 'pwd'
```

A continuación, se procede a brindar al usuario creado privilegios de usuario root o superusuario. Para ello, se ejecuta el siguiente comando SQL:

```
GRANT ALL PRIVILEGES ON dominioDB.* TO 'usuario'@%
```

El siguiente paso a realizar es la adquisición del nombre de dominio ante un proveedor de dominios. De esta manera, los dominios en cuestión poseen un dominio web a través del cual ser accedidos.

En esta situación, el estado de los dominios a servir está prácticamente listo para ser visible al usuario final. Lo siguiente que se realiza es la comunicación al servidor DNS autoritativo contratado de los dominios adquiridos y las IPs de los equipos que se encargan de servir el contenido o NDCs. En este caso, dado que el servidor en España ya se encuentra dado de alta en el servidor de resolución de nombres autoritativo, únicamente es necesario publicar como novedad la IP del NDC situado en México

7.5 Conclusiones

En este apartado se muestra una serie de gráficas que permiten conocer, de una forma clara, las diferencias existentes en el rendimiento cuando se hace uso de AMP y cuando no. De los resultados obtenidos se concluye lo siguiente.

En primer lugar, en la Ilustración 7.13, se comparan dos gráficas resultantes del conjunto de resultados obtenidos en los subapartados anteriores. En la primera, se muestra la comparativa obtenida por el complemento PageSpeed de GTMetrix, y la segunda la obtenida con YSlow. En ambos casos, en verde se encuentra la puntuación de AMP y en azul la No-AMP. Como se puede ver en ambas pruebas, AMP supera a la misma página cargada en No-AMP.

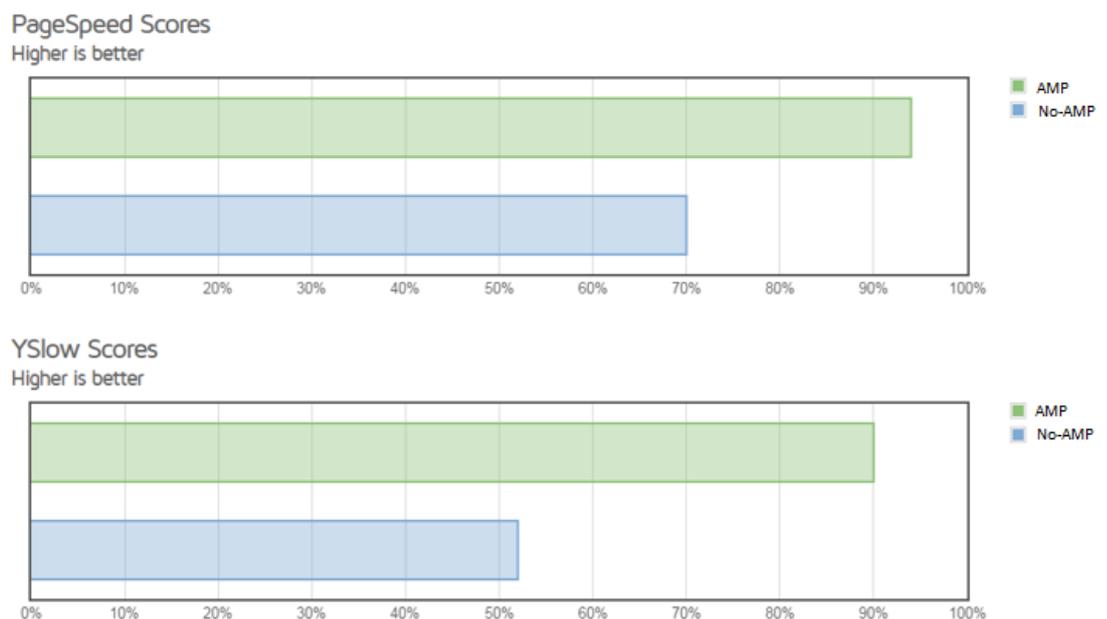


Ilustración 7—11: Gráficas comparativas – 1

Además, los análisis ejecutados por PageSpeed e YSlow, concluyen que la mejora del rendimiento que aporta AMP está por encima de la media de otros sitios analizados y cuyos datos también son recogidos por GTMetrix.

A continuación, se compara expresamente el tiempo de carga de ambas páginas AMP y No-AMP, así como el tamaño de página. Los resultados obtenidos previamente respecto a la velocidad de carga del sitio se muestran en la Ilustración 7.14. El tiempo medio de carga para la página AMP concreta es de aproximadamente 2.7s mientras que la No-AMP supera los 30s.

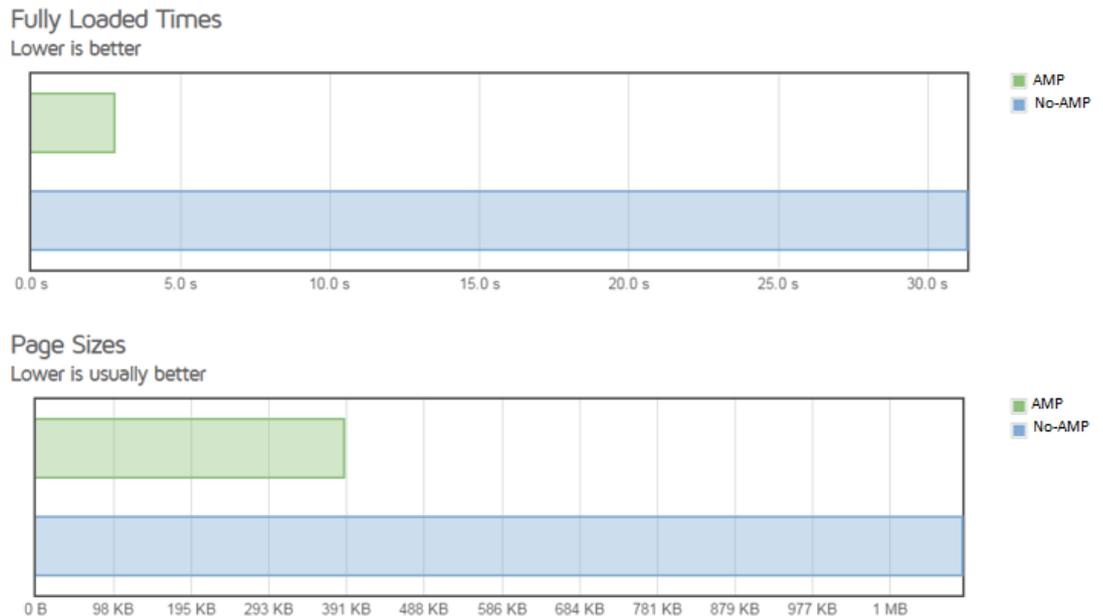


Ilustración 7—12: Gráficas comparativas - 2

Ello no se debe únicamente al empleo de una tecnología más eficiente como lo es AMP, sino a que el tamaño de página también es muy inferior. En la gráfica “Page Sizes”, se puede ver cómo el tamaño de la misma página en AMP es de apenas 390KB, mientras que en No-AMP es superior a 1MB. Es necesario recordar que, en AMP, tanto el tamaño de página como su velocidad de carga inicial son mejores debido a su estrategia de cargar determinados contenidos en tiempo real, es decir, mientras el usuario se mueve por la página. Esto difiere de lo que ocurre en No-AMP, donde el proceso de carga de todos los elementos se produce al principio, lo cual conlleva una peor experiencia de usuario y un potencial desaprovechamiento del ancho de banda al cargarse información no necesaria.

La Ilustración 7.15, que se muestra a continuación, indica que el número de peticiones realizadas por AMP es muy inferior al realizado en No-AMP.

Esto se debe a, entre otras cosas, la inclusión de código CSS en los ficheros HTML AMP y, fundamentalmente, a la prohibición de AMP respecto a obtener recursos externos al dominio de la página descargada.

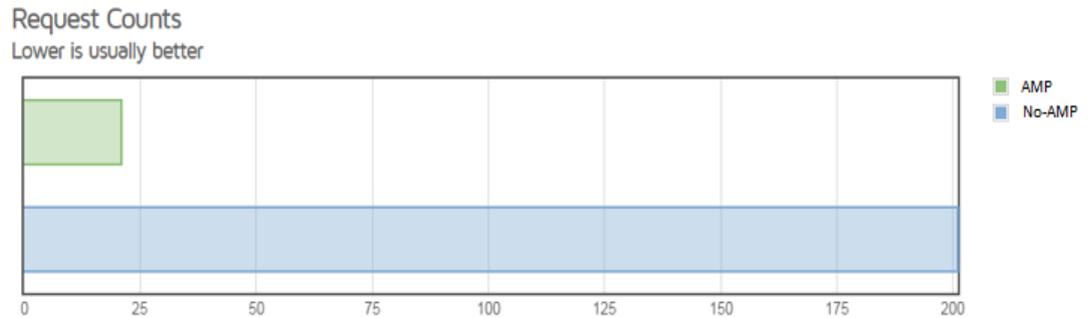


Ilustración 7—13: Graficas comparativas – 3

Por tanto, si se recogen los resultados como a continuación, se verifica el gran rendimiento de AMP.

Puntuación PageSpeed	A (94%)	C (70%) -24%
Puntuación YSLDW	A (90%)	E (52%) -38%
Tiempo de carga	2.8s	31.3s +28.5s
Tamaño de página	387KB	1.14MB +777KB
Peticiones Totales	21	201 +180

A comparison table showing performance metrics for AMP and No-AMP. The AMP column is highlighted with a green border and shows superior performance across all metrics. The No-AMP column is highlighted with a blue border and shows significantly worse performance. The metrics are: PageSpeed score (A vs C), YSLDW score (A vs E), load time (2.8s vs 31.3s), page size (387KB vs 1.14MB), and total requests (21 vs 201).

Ilustración 7—14: Resultados comparativos

8. PLAN DEL PROYECTO

8.1 Introducción

En el presente apartado se exponen los plazos de ejecución de las diferentes tareas que constituyen del proyecto. En cada una de ellas figuran su fecha de comienzo y finalización, así como las personas responsables de su ejecución y/o supervisión.

En primer lugar, se presenta el equipo de trabajo que interviene en el proyecto. Posteriormente, agrupadas en varios paquetes de trabajo, se muestran cuáles son las tareas y lo que se tiene que ejecutar en ellas a lo largo del proyecto. Cada una de éstas vendrá referenciada por un identificador de la forma T.x.x, que lo vincula al siguiente apartado, es decir, al diagrama de Gantt.

A continuación, se muestra el significado de la nomenclatura que se emplea:

- PT: Paquete de Trabajo.
- T: Tarea.
- UE: Unidad Entregable.
- CT: Carga de Trabajo (horas).

8.2 Equipo de trabajo

El personal directamente vinculado a la ejecución del proyecto se indica en la siguiente tabla.

Tabla 8-1: Personal encargado del proyecto

Id.	Responsabilidad	Horas
P1	Director del Proyecto	30
P2	Graduado en Telecomunicaciones	600
Total		630

8.3 Paquetes del proyecto

En este apartado se definen los paquetes de trabajo que se han realizado durante la ejecución del proyecto junto con las tareas que componen cada uno de ellos.

PT1: DIRECCIÓN Y GESTIÓN DEL PROYECTO.

Este paquete de trabajo consta de 7 tareas donde se define el proyecto que se va a realizar, como se va realizar, dónde, quien va participar, los pasos a seguir y el presupuesto que va a tener. En el participa tanto el director del proyecto, como el responsable de departamento y el graduado en ingeniería de telecomunicaciones.

Tabla 8-2: Paquete de trabajo 1. Horas de trabajo por persona según tarea

Id.	Definición Tarea	P1	P2
PT1	DIRECCIÓN Y GESTIÓN DEL PROYECTO.	19	75
T1.1	Definición del proyecto.	3	15
T1.2	Definición de los objetivos.	2	8
T1.3	Planificación del proyecto.	5	15
T1.4	Definición de medios y técnicas básicas.	5	20
T1.5	Definición de pruebas.	0	17
T1.6 (Hito)	<i>Presentación del proyecto.</i>	0	0

- **Recursos humanos**

Director: 19h

Graduado en ingeniería de telecomunicaciones: 75h

- **Recursos materiales**

-Ordenadores

- **Calendario**

[Comienzo: 10/10/2016 Fin: Semana 03/11/2016]

Duración: 2,5 semanas

PT2: Análisis de alternativas.

Este paquete consta de 4 tareas. En primer lugar, se definen las posibles alternativas para cada meta. En segundo lugar, se definen los criterios que permitan seleccionar la mejor alternativa para cada caso. En tercer lugar, se desarrollan pruebas piloto que permitan obtener los datos suficientes como para poder tomar una decisión sobre la alternativa a elegir. Finalmente, se realiza la evaluación y selección de las alternativas basándose en los resultados obtenidos en el anterior punto. En esta parte del proyecto participa únicamente el graduado en telecomunicaciones.

Tabla 8-3: Paquete de trabajo 2. Horas de trabajo por persona según tarea

Id	Definición Tarea	P2
PT2	Análisis de alternativas.	100
T2.1	Búsqueda de alternativas	50
T2.2	Definición de criterio de selección	5
T2.3	Desarrollo de alternativas sobre prueba piloto	40
T2.4	Selección de las alternativas óptimas	5

- **Recursos humanos**

Graduado en ingeniería de telecomunicaciones: 50 horas

- **Recursos materiales**

-Ordenadores

- **Calendario**

[Comienzo: 03/11/2016 Fin: 08/12/2016]

Duración: 4,5 semanas

PT3: Puesta en marcha del trabajo realizado y realización de pruebas de rendimiento.

Este paquete trata de desplegar y poner en marcha el conjunto de tecnologías aplicadas por las alternativas seleccionadas. Una vez se encuentran desplegadas, se valida y se testea que los resultados obtenidos coinciden con las pruebas realizadas en el apartado anterior. En este paquete participa el responsable del departamento, encargado de supervisar la veracidad de los resultados, y el graduado en ingeniería de telecomunicaciones.

Tabla 8-4: Paquete de trabajo 3. Horas de trabajo por persona según tarea

Id	Definición Tarea	P2
PT3	Despliegue y puesta en marcha.	280
T3.1	Despliegue de AMP.	200
T3.2	Despliegue de un Nodo Distribuidor de Contenidos (NDC).	40
T3.3	Aplicación de Web Performance Optimization.	20
T3.4	Contraste de resultados con datos previos al proyecto.	10
T3.5	Puesta en marcha del NDC.	5
T3.6	Realización de mediciones finales.	5

- **Recursos humanos**

Graduado en ingeniería de telecomunicaciones: 280 horas

- **Recursos materiales**

-Ordenadores

-Servicio de máquina remota para el NDC

-Conectividad a internet

-Servicio DNS autoritativo

- **Calendario**

[Comienzo: 08/12/2016 Fin: 16/03/2017]

Duración: 12 semanas

PT4: Fase de pruebas en funcionamiento real.

Este paquete consta de 3 tareas. En él se dispone del sistema desplegado y se procede a comprobar su correcto funcionamiento y eficacia. En primer lugar, se verifica que la disponibilidad del servicio sea estable y sin caídas o fallos. En segundo lugar, se verifica que el sistema es efectivo en todos los casos en que se está empleando. Por último, en tercer lugar, se comprueba que las modificaciones introducidas debido a las nuevas tecnologías no afectan al rendimiento de las fuentes de ingresos, como puede ser la publicidad. En él participan el responsable de departamento y el graduado en telecomunicaciones.

Tabla 8-5: Paquete de trabajo 4. Horas de trabajo por persona según tarea

Id	Definición Tarea	P2
PT4	VERIFICACIÓN DEL SISTEMA FINAL	30
T4.1	Verificación de disponibilidad de servicio.	15
T4.2	Verificación de la eficacia.	8
T4.3	Validación de los resultados.	7
T4.4 (Hito)	Comprobación de resultados	0

- **Recursos humanos**

Graduado en ingeniería de telecomunicaciones: 30 horas

- **Recursos materiales**

-Ordenadores

-Servicio de máquina remota para el NDC

-Conectividad a internet

-Servicio DNS autoritativo

- **Calendario**

[Comienzo: 16/03/2017 Fin: 28/03/2017]

Duración: 1,5 semanas

PT5: ELABORACIÓN DE LA DOCUMENTACIÓN DEL PROYECTO.

Este paquete consta de 2 tareas. En él se leen de nuevo los informes realizados a la conclusión de cada paquete y se desarrolla toda la documentación final. En este paquete participan el director del proyecto, el responsable de departamento, y el graduado en telecomunicaciones.

Tabla 8-6: Paquete de trabajo 5. Horas de trabajo por persona según tarea

Id	Definición Tarea	P1	P2
PT5	ELABORACIÓN DE LA DOCUMENTACIÓN DEL PROYECTO.	15	115
T5.1	Lectura de informes desarrollados y escritura de resultados.	15	15
T5.2	Descripción del proyecto (documentación).	0	100

- **Recursos humanos**

Director: 15 horas

Graduado en Ingeniería de Telecomunicaciones: 115 horas

- **Recursos materiales**

-Ordenadores

- **Calendario**

[Comienzo: 28/03/2017 Fin: 05/05/2017]

Duración: 6 semanas

8.4 Duración total

En la siguiente tabla se puede observar un resumen de la carga de trabajo en semanas de cada uno de los paquetes de trabajo planificados en la ejecución del proyecto.

Tabla 8-4: Duración total del proyecto

Id.	Semanas
PT1	2.5
PT2	4.5
PT3	12
PT4	1.5
PT5	6
TOTAL	26.5

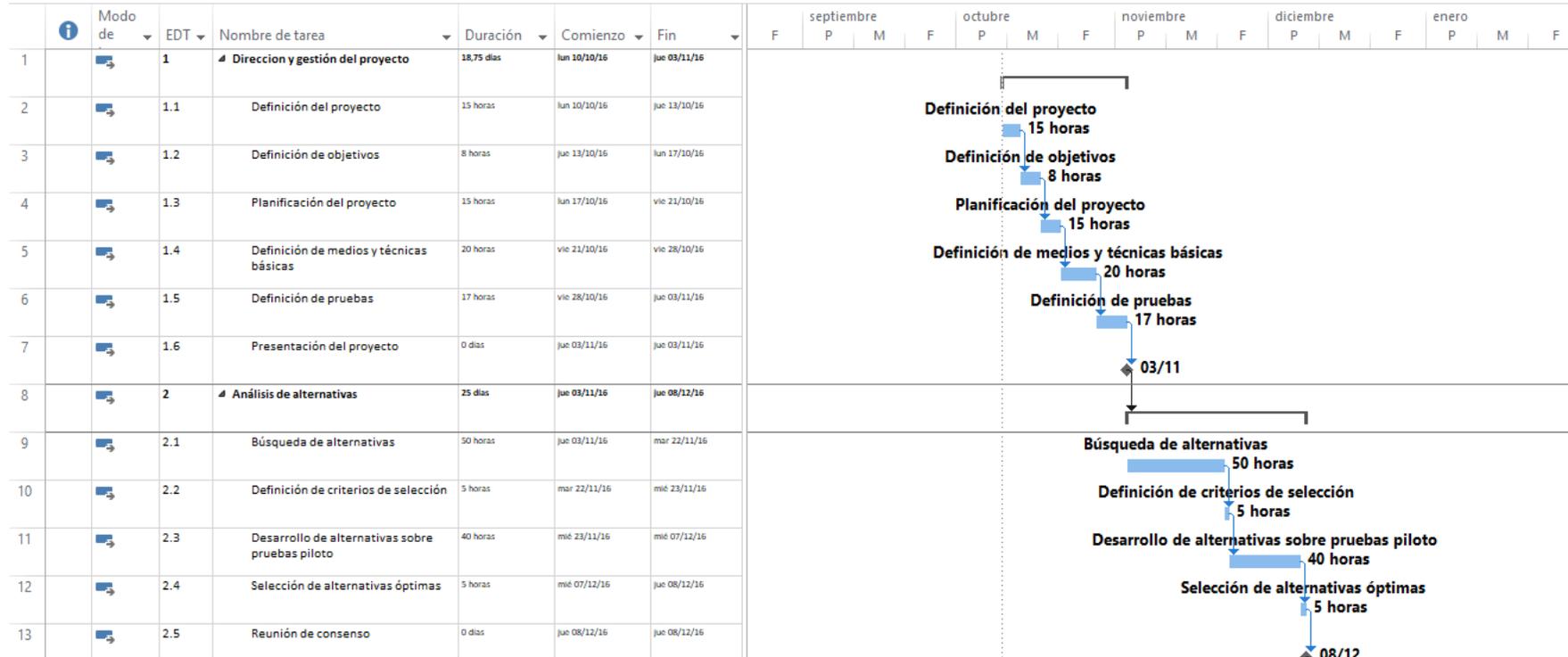
8.5 Hitos

En este apartado se reflejan aquellos puntos críticos que suponen un punto de inflexión en el proyecto. Consisten, básicamente, en puntos de control que dan idea del progreso del propio proyecto. A continuación, se muestran esos puntos críticos o hitos.

Tabla 8-5: Hitos

Hito	Fecha	Descripción
H1	03/11/2016	Presentación del proyecto
H2	08/12/2016	Reunión de consenso
H3	02/03/2017	Verificación de funcionamiento del NDC
H4	28/03/2017	Comprobación de resultados exitosos

8.6 Diagrama de Gantt



Optimización de la velocidad y rendimiento de Servicios Web

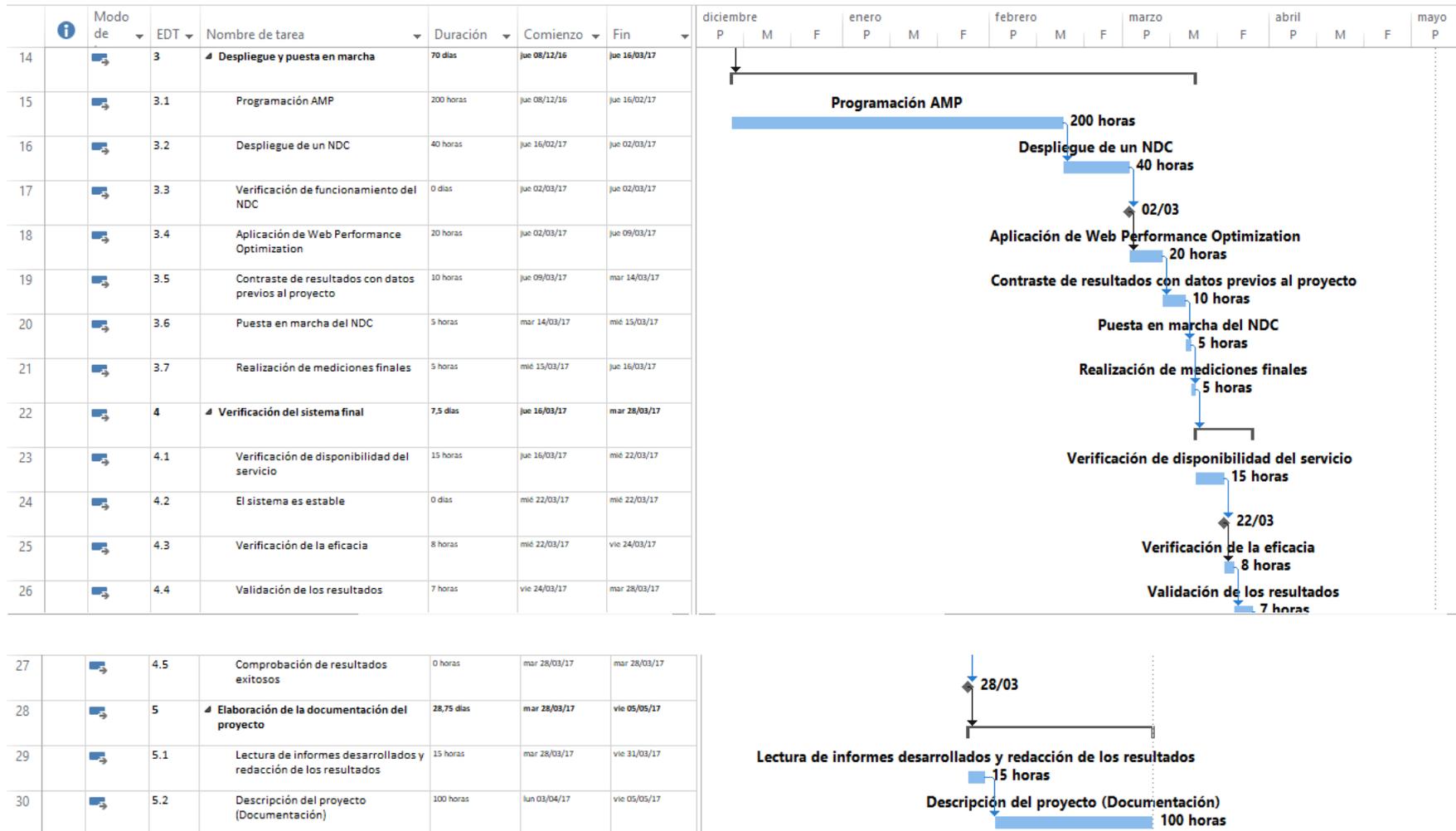


Ilustración 8—1: Diagrama de Gantt

9. RECURSOS HUMANOS Y MATERIALES

En este apartado se definen los recursos de los que dispone la empresa para llevar a cabo el proyecto propuesto.

9.1 Recursos humanos

Para la selección de los recursos humanos no se pretende atender a las cualificaciones de quienes son contratados sino sus capacidades, que tendrán que ser las necesarias para desempeñar su trabajo, pero si recibirán una remuneración adecuada al nivel de cualificaciones que se supone necesario para tener esos conocimientos. Por tanto, los recursos humanos requeridos son los siguientes:

- Se requiere una persona capaz de gestionar el proyecto y llevar a cabo las labores de dirección del mismo, como pueden ser la supervisión de los trabajos y la posible variación en los objetivos. (Director del proyecto).
- Se requiere de personal con conocimientos en programación web, capaz de aprender y adaptarse a las nuevas tecnologías que pudieran ser necesarias. Se exige además, cierta destreza en la redacción de documentación de carácter oficial. Por último, se deben poseer ciertos conocimientos de los siguientes lenguajes de programación y aspectos telemáticos:
 - PHP
 - JavaScript
 - HTML5
 - AMP
 - PERL
 - SQL
 - Manejo de Sistemas Linux (Gnome y CentOS)
 - Configuración de resolución de Nombres (DNS)
 - Redes de distribución de contenidos (CDNs)

Estas habilidades son recogidas por el Graduado en Ingeniería Técnica de Telecomunicaciones encargado de la realización de este proyecto.

9.2 Recursos materiales

En este apartado se especifican los recursos materiales requeridos para la realización del proyecto. Son los siguientes:

Tabla 10-9-1: Recursos materiales

Recurso	Cantidad
Ordenador	1
Nodo de distribución de contenidos	1
Servicio de internet	1
Contratación de servidor web	1

Los recursos deben cumplir con una serie de especificaciones que se detallan a continuación:

- El ordenador necesario debe permitir instalar software de desarrollo y manejarlo con fluidez. Se requieren al menos 8gb de memoria RAM, 250gb de almacenamiento, a ser posible SSD, y procesador no inferior a Intel Core i5-i4460.
- Como recursos software se requieren paquetes ofimáticos instalados sobre un sistema operativo Microsoft Windows. Además, se requiere de la instalación de las siguientes herramientas:
 - o Putty: para conexiones remotas SSH.
 - o Dreamweaver: SDK para el desarrollo web.
 - o NavyCat: para la gestión de bases de datos.
- Se requiere un servidor web sobre la plataforma Linux, ya sea CentOS o Ubuntu, preferiblemente el segundo.
- Se requiere un contrato con un operador cualquiera que ofrezca el servicio conexión a internet.

10. Presupuesto

En este apartado se muestran los detalles económicos que conciernen al presente proyecto: “**Optimización de la velocidad y rendimiento de Servicios Web**”. Además, se muestran los recursos económicos requeridos para lograr su consecución.

10.1 Costes unitarios

10.1.1 Mano de obra

Tabla 10–9–2: Costes en Recursos Humanos

Código	Nombre	Responsabilidad	Tasa Horaria
JIU	Juanjo Igarza Ugaldea	Director del proyector	60€/hora
AUE	Asier Unanue Esteban	Ingeniero Junior	30€/hora

10.1.2 Material amortizable

Tabla 10–9–3: Costes en material amortizable

Código	Descripción	Precio	Amortización	Precio Unitario
MA1	Ordenador sobremesa	600€	3 años	0,022€/hora
MA2	Licencia Windows 10 Pro	280€	1 año	0,031€/hora
MA3	Licencia Microsoft Office 2016	180€	1 año	0,021€/hora

MA4	Monitor	100€	3 años	0,004€/hora
MA5	Ratón y Teclado	20€	5 años	0,0004€/hora

10.1.3 Material fungible

Tabla 10-9-4: Costes en Material Fungible

Código	Descripción	Precio
MF1	Material de oficina	250€
MF2	Conexión a internet (5 meses)	150€
MF3	Gasto eléctrico (5 meses)	240€
MF4	Alquiler Host NDC (5 meses)	250€

10.2 Gastos en Recursos Humanos

Para realizar el cálculo del coste de las horas de cada integrante del proyecto, se hace uso de las horas totales dedicadas por cada una de las partes mostradas anteriormente. Por tanto, los costes en RR.HH son los siguientes:

Tabla 10–9–5: Costes en Recursos Humanos

Responsable	Horas Invertidas	Coste
JIU	30	1.800€
AUE	600	18.000€
Total		19.800€

10.3 Gastos en recursos materiales

En este apartado se recogen los gastos tanto de recursos materiales amortizables como los de los recursos fungibles.

Tabla 10–9–6: Costes en Recursos Materiales

Código del material	Cantidad	Horas de uso	Precio unitario	Coste Total
MA1	1	600	0,022€/hora	13,2€
MA2	1	600	0,031€/hora	18,6€
MA3	1	75	0,021€/hora	1,6€
MA4	1	600	0,004€/hora	2,4€
MA5	1	600	0,0004€/hora	0,25€
MF1	-	-	-	250€
MF2	-	-	-	150€
MF3	-	-	-	240€
MF4	-	-	-	250€
Total				924,05€

10.4 Resumen final de los gastos del proyecto

En la siguiente tabla se suman los gastos relativos al proyecto tomando en consideración los resultados obtenidos a lo largo de este apartado.

Tabla 10-9-7: Costes totales del proyecto

Concepto	Coste Total
Recursos Humanos	19.800€
Recursos Materiales	924,05€
Total (Antes de imprevistos)	20.726,05€
Imprevistos (4%)	829€
Total	21.555€

Según los datos obtenidos, el desarrollo del proyecto “**Optimización de la velocidad y rendimiento de Servicios Web**” es de 21.555€.

11. RIESGOS DEL PROYECTO Y PLAN DE CONTINGENCIA

En la ejecución del proyecto existen diversos riesgos que, en mayor o menor medida, Resumen y matriz de riesgos

Se va a mostrar en una matriz de una forma sencilla, la gravedad de cada riesgo y su probabilidad de ocurrencia. Con ésta se trata de determinar aquellos problemas que pueden dificultar en mayor medida no solo el diseño, sino el posterior funcionamiento del equipo, para así centrarse en mayor grado en la obtención de una posible solución ante los más perjudiciales.

Hay riesgos que, aun siendo poco probables pueden llegar a ser graves, mientras que, otros que tienen una mayor probabilidad de ocurrencia pueden solucionarse más fácilmente. En la matriz de riesgos que se presenta en la tabla X se muestran tres colores, los riesgos menos preocupantes son los de color verde, luego los amarillos y los más graves los rojos. Dentro de cada uno de estos colores existen varios grados, representados por la intensidad de color, la diferencia entre ellos estriba en la probabilidad de ocurrencia de cada contratiempo. En este caso, el más preocupante será aquel situado más a la derecha de la fila, esto es, el más probable dentro de un mismo grado de gravedad. A continuación, se citan y numeran los riesgos a fin de ser mostrados en la tabla:

1. Empresa competidora hace uso de AMP con mayor rapidez.
2. Los beneficios de rendimiento son inferiores a lo esperado.
3. El correcto funcionamiento del sistema no es compatible con la optimización.

Tabla 11–1: Evaluación de riesgos

Prob/Impacto	Muy bajo 0.1	Bajo 0.2	Medio 0.3	Alto 0.4	Muy Alto 0.5
<i>Muy baja 0.1</i>					
<i>Baja 0.2</i>		1			
<i>Media 0.3</i>		2		3	
<i>Alta 0.4</i>					
<i>Muy Alta 0.5</i>					

Los planes de contingencia valorados para los riesgos citados son los siguientes, siguiendo el mismo orden numérico:

1. No es un riesgo evitable. El objetivo es mantener el rendimiento optimizado al día aplicando las futuras novedades que aparezcan de forma continuada.
2. En caso de que el rendimiento sea inferior al actual puede deberse a dos causas: el mal uso de las nuevas tecnologías, o bien, éstas no son ideales para ser empleadas en el sistema. La solución radica en mantener el sistema en su estado más óptimo hasta lograr mejorar su rendimiento.
3. En caso de que algunas funcionalidades desplegadas en la actualizad no puedan ser añadidas debido a las restricciones de la optimización web, se debe sopesar otra manera de ofrecer dichas funcionalidades. En caso de tratarse de funciones en desuso cuya inclusión supone una demora significativa en la velocidad de carga o rendimiento de la página web, pueden ser suprimidas.

12. CONCLUSIONES

En este proyecto se ha llevado a cabo un análisis de las nuevas tecnologías normativas y tecnologías propuestas por algunos actores importantes en el entorno web como puede ser Google.

Actualmente, el problema se encuentra en que la mayor parte de las páginas que funcionan sobre terminales móviles no fueron diseñadas para ellos, sino que son fruto de la adaptación y, por tanto, suponen una carga que no cualquier terminal puede soportar de manera fluida. Este proyecto, demuestra que las nuevas tecnologías propuestas consiguen mejorar el rendimiento de las páginas web sin disminuir en los servicios ofertados en sus versiones complejas y sin afectar en absoluto a la codificación ya existente para otras plataformas. Para lograrlo, se han aplicado las normas propuestas por las siguientes tecnologías:

- Accelerated Mobile Pages o AMP, que permite cargar una página web con idénticos servicios, disminuyendo el tiempo de carga de la misma, reduciendo la cantidad de peticiones requeridas para visualizar la página y ocupando un menor ancho de banda.
- Redes de Distribución de Contenidos, que permiten situar nuevos servidores interconectados con el resto de la red del servicio, ofreciendo así una reducción del tiempo de carga de las páginas web de varios segundos debido a la reducción del RTT.
- Uso de scripts de minificación que, proporcionando versiones minificadas del código, facilitan el procesado del código a los clientes web o navegadores.
- Compresión de los recursos a transmitir, lo cual supone una reducción en el ancho de banda requerido y, consecuentemente, un menor tiempo de carga.

13. REFERENCIAS

- [1] Analytics, «Google Analytics,» [En línea]. Available: <https://analytics.google.com/>.

- [2] J. Pernas, «Pernas,» [En línea]. Available: <https://www.pernas.eu/wpo-web-performance-optimization/>. [Último acceso: 12 2016].

- [3] Wikipedia, «Wikipedia,» [En línea]. Available: <http://en.wikipedia.org>.

- [4] A. Mehta, «Slideshare,» [En línea]. Available: <https://www.slideshare.net/developeratworks/>. [Último acceso: 01 2017].

- [5] D. B.F, «Dariobf,» [En línea]. Available: <https://www.dariobf.com>. [Último acceso: 01 2017].

- [6] «ACENS,» [En línea]. Available: <https://www.acens.com/>.

- [7] Wikipedia, «Wikipedia,» [En línea]. Available: <http://en.wikipedia.org>. [Último acceso: 10 2016].

- [8] W3C, «w3schools,» [En línea]. Available: <https://www.w3schools.com/>. [Último acceso: 10 2016].

- [9] L. Kim, «kissmetrics,» [En línea]. Available: <https://blog.kissmetrics.com/get-google-to-index/>. [Último acceso: 12 2016].

- [10] MOZ, «moz,» [En línea]. Available: <https://moz.com/learn/seo/duplicate-content>. [Último acceso: 02 2017].
- [11] Qo2, [En línea]. Available: <https://www.quondos.com/que-es-el-link-juice/>. [Último acceso: 01 2017].
- [12] E. Torregosa, «webpositer,» [En línea]. Available: <http://www.webpositer.com/>. [Último acceso: 12 2016].
- [13] Zeokat, [En línea]. Available: <http://www.vozidea.com/>. [Último acceso: 01 2017].
- [14] «4cornermedia,» [En línea]. Available: <https://4cornermedia.com/>. [Último acceso: 12 2016].
- [15] «CtAmericas,» [En línea]. Available: <http://www.ctamericas.com>. [Último acceso: 01 2016].
- [16] «Google Cloud Flare,» [En línea]. Available: <https://cloud.google.com/about/locations/#regions-tab>. [Último acceso: 01 2017].
- [17] Amazon, «aws,» [En línea]. Available: <https://aws.amazon.com/es/cloudfront/details/>. [Último acceso: 01 2017].
- [18] «AmpProject,» [En línea]. Available: <https://www.ampproject.org/es/learn/case-studies/>. [Último acceso: 11 2016].
- [19] Bootstrap. [En línea]. Available: <https://v4-alpha.getbootstrap.com/>. [Último acceso: 11 2016].

- [20] [En línea]. Available: <https://stackoverflow.com/questions/9702643/mysql-vs-mongodb-1000-reads>.
- [21] «DigitalServer,» [En línea]. Available: <https://www.digitalserver.com.mx/servidores-vps-economicos.shtml>.
- [22] «DigitalServer,» [En línea]. Available: <https://www.digitalserver.com.mx/>. [Último acceso: 12 2016].
- [23] O. Ruiz, «rodrigoruiliver,» [En línea]. Available: <http://rodrigoruiliver.blogspot.com.es/>. [Último acceso: 12 2017].
- [24] Google Analytics, «Datos extraídos de Google Analytics,» 2017.
- [25] «Wikipedia,» [En línea]. Available: <http://en.wikipedia.org>. [Último acceso: 2016].