



UNIVERSIDAD DE VALLADOLID

GRADO EN INGENIERÍA DE TECNOLOGÍAS
ESPECÍFICAS DE TELECOMUNICACIÓN

MENCIÓN EN INGENIERÍA TELEMÁTICA

CURSO 2016/2017

TRABAJO DE FIN DE GRADO

***DESARROLLO DE UN PORTAL WEB EMPLEANDO TÉCNICAS
DE COMPUTACIÓN VERDE***

MARIO ALBERTO DURÁNTEZ STOLLE

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN

VALLADOLID, JULIO 2017

UNIVERSIDAD DE VALLADOLID

**GRADO EN INGENIERÍA DE TECNOLOGÍAS
ESPECÍFICAS DE TELECOMUNICACIÓN**

MENCIÓN EN INGENIERÍA TELEMÁTICA

CURSO 2016/2017

Trabajo presentado por:

Mario Alberto Duránte Stolle

Tutores:

Carlos Alonso Gómez y Patricia Fernández del Reguero

**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN**

VALLADOLID, JULIO DE 2017

AGRADECIMIENTOS

A mi familia por su apoyo incondicional durante toda la carrera.

A Carlos por su inestimable ayuda en la realización del Trabajo.

RESUMEN

El presente Trabajo Fin de Grado comprende todo el proceso de remodelación del portal web de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid. La remodelación se realiza bajo el contexto de la Computación Verde o Green ICT, "una colección de iniciativas estratégicas y tácticas que reducen directamente la huella de carbono de las operaciones informáticas de una empresa." (O'Neill, 2010)

Además de la remodelación y adecuación a los estándares de la programación verde, se adapta el portal web a la navegación desde diferentes dispositivos multimedia como smartphones o tablets mediante desarrollo web adaptativo (Responsive Web Development).

La realización de este trabajo consiste en las siguientes fases:

- Análisis del portal web de la E.T.S.I.T. previo a la realización del Trabajo Fin de Grado.
- Estudio del estado del arte en los ámbitos del diseño web, desarrollo web adaptativo y programación verde.
- Análisis de portales web de ámbito similar.
- Diseño y remodelación del portal de la E.T.S.I.T.
- Desarrollo Web Adaptativo.
- Implementación de medidas de computación verde.

PALABRAS CLAVE:

Desarrollo Web, Computación Verde, Diseño Web Adaptativo, Optimización.

ABSTRACT

This End-Of-Degree project comprises the complete renovation process for the web portal of the E.T.S.I.T. of the University of Valladolid. This renovation is carried out in the context of Green Computing or Green ICT, "a collection of strategic and tactical initiatives that directly reduces the carbon footprint of an organization's computing operations". (O'Neill, 2010)

On top of said renovation and adaptation to Green Computing's standards, the portal is designed and developed using Responsive Web Development techniques, in order for it to be easier and faster to navigate on any device, including smartphones and tablets.

The work done comprises the following phases:

- Analysis of the current web portal of the E.T.S.I.T., before the work of this End-Of-Degree Project.
- Study of the art in the areas of web design, responsive web development and green computing.
- Analysis of web portals of similar scope.
- Design and remodeling of the new E.T.S.I.T.'s web portal.
- Responsive Web Development.
- Implementation of Green Computing measures.

KEYWORDS:

Web development, Green Computing, Responsive Web Design, Optimization.

ÍNDICE

AGRADECIMIENTOS.....	5
RESUMEN	7
ABSTRACT	8
ÍNDICE.....	9
ÍNDICES DE CÓDIGO, TABLAS E ILUSTRACIONES.....	10
CÓDIGO	10
TABLAS	10
ILUSTRACIONES	11
1: INTRODUCCIÓN Y CONOCIMIENTOS PREVIOS.....	13
1.1 – INTRODUCCIÓN. MOTIVACIÓN. OBJETIVOS.....	13
1.2 – CONOCIMIENTOS PREVIOS.....	16
1.3 – ANÁLISIS DEL PROBLEMA Y ESTADO DEL ARTE	18
1.3.1 - DESCRIPCIÓN GENERAL DEL PORTAL WEB	18
1.3.2 - PRESENTACIÓN DE LA INFORMACIÓN	21
1.3.3 - ESTADO DEL ARTE: DISEÑO WEB ADAPTABLE	26
1.3.4 - ESTADO DEL ARTE: PROGRAMACIÓN VERDE.....	31
2: TRABAJO DESARROLLADO.....	38
2.1 - REDISEÑO VISUAL Y FUNCIONAL.....	39
2.1.1 – ANÁLISIS CUALITATIVO DE PORTALES WEB DE ÁMBITO SIMILAR	39
2.1.2 – ESPECIFICACIONES PREVIAS, REQUISITOS.....	44
2.1.3 – REDISEÑO DEL PORTAL.....	46
2.2 – RESPONSIVE WEB DESIGN.....	62
2.2.1 –HERRAMIENTAS PARA DISEÑO ADAPTATIVO (Bootstrap)	63
2.2.2 – CREACIÓN DEL PORTAL WEB ADAPTABLE.....	66
2.3 – PROGRAMACIÓN VERDE	70
2.3.1 – ANÁLISIS DEL CONSUMO DE RED	71
2.3.2 – MEDIDAS PARA REDUCIR EL CONSUMO RED	76
2.3.3 – CONSUMO ENERGÉTICO	83
3: CONCLUSIONES Y DISCUSIÓN	92
4: TRABAJO FUTURO	94
Bibliografía	98

ÍNDICES DE CÓDIGO, TABLAS E ILUSTRACIONES

CÓDIGO

CÓDIGO 1 – CÓDIGO HTML DE UN ELEMENTO VACÍO DEL CARRUSEL DE IMÁGENES.....	51
CÓDIGO 2 – CÓDIGO HTML DE UN ENLACE DE INTERÉS VACÍO.....	55
CÓDIGO 3 - CÓDIGO HTML DE UN ENLACE A RED SOCIAL VACÍO.....	56
CÓDIGO 4 - EJEMPLO DE USO EN CSS DE MEDIA QUERIES. ARRIBA: REGLA PARA PANTALLAS ENTRE 700PX Y 1000PX DE ANCHO. ABAJO: REGLA PARA PANTALLAS ENTRE 700PX Y 400PX DE ANCHO.....	62
CÓDIGO 5 - COMANDO UTILIZADO PARA LA COMPRESIÓN CON PÉRDIDAS DE ARCHIVOS DE IMAGEN.....	76
CÓDIGO 6 - CÓDIGO JAVASCRIPT PARA LA CARGA ASÍNCRONA DE SCRIPTS.....	78
CÓDIGO 7 - EJEMPLO DE MINIFICACIÓN DE CÓDIGO JAVASCRIPT. ARRIBA, CÓDIGO ORIGINAL, 77 CARACTERES. ABAJO, CÓDIGO MINIFICADO, 48 CARACTERES.....	80
CÓDIGO 8 - COMANDOS PARA MINIFICAR ARCHIVOS. ARRIBA: MINIFICACIÓN JAVASCRIPT CON <i>UGLIFY</i> . ABAJO, MINIFICACIÓN CSS CON <i>CLEANCSS</i>	80

TABLAS

TABLA 1 - DESGLOSE DE USUARIOS DIVIDIDOS ENTRE NUEVOS (UNA ÚNICA SESIÓN) Y HABITUALES (VARIAS SESIONES). GOOGLE ANALYTICS.....	24
TABLA 2 - PÁGINAS MÁS VISITADAS POR USUARIOS NUEVOS (IZQUIERDA) Y RECURRENTES (DERECHA). GOOGLE ANALYTICS..	24
TABLA 3 - DESGLOSE DE VISITAS AL PORTAL EN FUNCIÓN DEL DISPOSITIVO EMPLEADO. DESKTOP: ORDENADOR DE ESCRITORIO. MOBILE: TELÉFONO MÓVIL. TABLET: TABLETA. GOOGLE ANALYTICS.....	25
TABLA 4 - CANTIDAD DE DESECHO TECNOLÓGICO GENERADA EN EL MUNDO. LOS DATOS DESDE 2015 EN ADELANTE SON PREDICCIONES. (BALDÉ, 2014).....	31
TABLA 5 - CONSUMO ELÉCTRICO DE DIFERENTES REDES DE INTERNET. (VELA, 2011).....	33
TABLA 6 - CLASES Y MEDIDAS POR DEFECTO PARA RESPONSIVE WEB DESIGN EN BOOTSTRAP.....	64
TABLA 7 - RESULTADOS DE LA UTILIZACIÓN DE RED EN LA PÁGINA PRINCIPAL (TEL.UVA.ES).....	72
TABLA 8 - RESULTADOS DE LA UTILIZACIÓN DE RED EN UNA PÁGINA SECUNDARIA (HISTORIA).....	74
TABLA 9 - ANÁLISIS DEL CONSUMO DE RED Y TIEMPO DE CARGA DE OTROS PORTALES WEB DE CENTROS UNIVERSITARIOS ANALIZADOS.....	74
TABLA 10- DESGLOSE DE TIEMPO DE PROCESADOR POR TAREAS PARA TEL.UVA.ES (CARGA).....	85

ILUSTRACIONES

ILUSTRACIÓN 1 - DISTINTAS VERSIONES DEL PORTAL DE LA ESCUELA A LO LARGO DE LOS AÑOS.	13
ILUSTRACIÓN 2 - PÁGINA PRINCIPAL DEL PORTAL DE LA ESCUELA ANTES DE LA REALIZACIÓN DE ESTE PROYECTO.....	14
ILUSTRACIÓN 3 - PÁGINA PRINCIPAL DEL PORTAL DE LA ESCUELA ANTES DE LA REALIZACIÓN DE ESTE PROYECTO.....	19
ILUSTRACIÓN 4 - UNA PÁGINA SECUNDARIA DEL PORTAL DE LA ESCUELA ANTES DE LA REALIZACIÓN DE ESTE PROYECTO.....	20
ILUSTRACIÓN 5 - LUGARES DONDE LOS USUARIOS ESPERAN ENCONTRAR COMPONENTES TÍPICOS DE UNA PÁGINA WEB.	22
ILUSTRACIÓN 6 - PORCENTAJE DE TIEMPO EMPLEADO MIRANDO CONTENIDO EN FUNCIÓN DE LOS PÍXELES DESDE EL INICIO. EN ROJO, LÍMITE DE CONTENIDO VISIBLE SIN HACER SCROLLING. (NIELSEN, SCROLLING AND ATTENTION, 2010)	23
ILUSTRACIÓN 7 - TIEMPO EMPLEADO POR ADULTO Y DÍA EN CONTENIDOS DIGITALES, POR TIPO DE DISPOSITIVO. EN VERDE, MÓVILES. EN AZUL, ORDENADORES. EN AMARILLO, OTROS. (MEEKER, 2017)	26
ILUSTRACIÓN 8 - PÁGINA PRINCIPAL DEL PORTAL ANTIGUO DE LA ESCUELA VISTO EN PANTALLA DE MOVIL DE 360x640PX. IZQUIERDA: SIN ZOOM. DERECHA: CON ZOOM	28
ILUSTRACIÓN 9 - RESULTADOS DEL TEST DE OPTIMIZACIÓN PARA MÓVILES DE GOOGLE EN LA VERSIÓN ANTIGUA DEL PORTAL DE LA ESCUELA	29
ILUSTRACIÓN 10 - GASTO ENERGÉTICO TOTAL DE SERVIDORES DE INTERNET EN ESTADOS UNIDOS (MILES DE MILLONES DE KWH/AÑO). LOS DATOS DE 2014 EN ADELANTE SON PREDICIONES. LOS COLORES INDICAN LA GAMA DEL SERVIDOR, SIENDO LOS DE GAMA MÁS ALTA MORADOS, Y LA MÁS BAJA ROJOS. (SHEBABI, 2016)	32
ILUSTRACIÓN 11 - CANTIDAD SERVIDORES EN CENTROS DE DATOS INSTALADOS EN EL MUNDO, EN MILLONES POR AÑO. EN AZUL SERVIDORES PRIVADOS, EN VERDE PÚBLICOS. (CISCO, CISCO GLOBAL CLOUD INDEX: FORECAST AND METHODOLOGY, 2015–2020, 2016).....	32
ILUSTRACIÓN 12 - GASTO ENERGÉTICO TOTAL DE SERVIDORES DE INTERNET EN ESTADOS UNIDOS (MILES DE MILLONES DE KWH/AÑO). EN LÍNEA DE PUNTOS, PREDICIONES EN EL AÑO 2010. ÁREA EN GRIS: AHORRO TOTAL POR MEDIDAS DE OPTIMIZACIÓN ENERGÉTICA. (SHEBABI, 2016)	34
ILUSTRACIÓN 13 - MEDIA DEL TOTAL DE DATOS TRANSFERIDOS Y PETICIONES HTTP EN LAS 10.000 PÁGINAS DE INTERNET MÁS VISITADAS DEL MUNDO. (HTTPARCHIVE, 2017).....	36
ILUSTRACIÓN 14 - IZQUIERDA: ETSETB BARCELONA. DERECHA: ETSIT GRANADA	40
ILUSTRACIÓN 15 - IZQUIERDA: ETSIT UPM. DERECHA: INFORMÁTICA UVA.....	40
ILUSTRACIÓN 16 - FÍSICAS UCM	41
ILUSTRACIÓN 17 - NUEVO LOGO DE LA ESCUELA.....	45
ILUSTRACIÓN 18 - ESQUEMA DE LOS ELEMENTOS Y ESTRUCTURA EN LA PÁGINA PRINCIPAL DEL NUEVO PORTAL. CÓDIGO DE COLORES EXPLICADO EN EL CAPÍTULO 2.1.1.	49
ILUSTRACIÓN 19 - BARRA DE NAVEGACIÓN EN DIFERENTES ESTADOS. SUPERIOR: EN ESPERA. CENTRAL E INFERIOR: DESPLEGADAS SECCIONES "LA ESCUELA", Y "TITULACIONES" RESPECTIVAMENTE.....	50
ILUSTRACIÓN 20 - BANNER DE LA CABECERA DE LA PÁGINA PRINCIPAL. LOGO DE LA ESCUELA, NOMBRE DE LA ESCUELA Y LA UNIVERSIDAD, Y LOGO CONMEMORATIVO DEL 25º ANIVERSARIO DE LA ESCUELA.	50
ILUSTRACIÓN 21 - VISTA DEL CARRUSEL DE IMÁGENES. SE APRECIAN LOS BOTONES PARA AVANZAR Y RETROCEDER, Y EL RÓTULO EXPLICATIVO DE LA IMAGEN.	51
ILUSTRACIÓN 22 - VISTA PREVIA DEL TABLÓN DE ANUNCIOS EN LA PÁGINA PRINCIPAL DEL NUEVO PORTAL.	52
ILUSTRACIÓN 23 - LÍNEA TEMPORAL DEL PERFIL DE TWITTER DE LA ESCUELA.	53
ILUSTRACIÓN 24 - VISTA COMPLETA DE LA PRIMERA SECCIÓN HORIZONTAL.	53
ILUSTRACIÓN 25 - VISTA GENERAL DE LA SECCIÓN DE ENLACES DE INTERÉS.	54
ILUSTRACIÓN 26 - VISTA DE UN ENLACE DE INTERÉS. IZQUIERDA: INACTIVO, SÓLO TÍTULO VISIBLE. DERECHA: ACTIVO (PUNTERO POR ENCIMA), VISIBLE TÍTULO Y DESCRIPCIÓN.....	54
ILUSTRACIÓN 27 - ENLACE DE INTERÉS PERSONALIZADO, CON EL VÍDEO DE YOUTUBE INCRUSTADO.	55
ILUSTRACIÓN 28 - PÍE DE PÁGINA DEL NUEVO PORTAL.....	56
ILUSTRACIÓN 29 - VISTA GLOBAL DE LA NUEVA PÁGINA PRINCIPAL (COMPLETA) DE LA ESCUELA.	57

ILUSTRACIÓN 30 - ESQUEMA DE LOS ELEMENTOS Y ESTRUCTURA DE LAS PÁGINAS SECUNDARIAS DEL NUEVO PORTAL. CÓDIGO DE COLORES EXPLICADO EN EL CAPÍTULO 2.1.1. ESPACIO CENTRAL EN BLANCO DEDICADO A CONTENIDO ESPECÍFICO DE LA PÁGINA.	58
ILUSTRACIÓN 31 - EJEMPLO DE NAVEGACIÓN DE DOS NIVELES (IZQUIERDA) Y DE TRES NIVELES (DERECHA) EN LA BARRA DE NAVEGACIÓN LATERAL.	59
ILUSTRACIÓN 32 - EJEMPLOS DE PÁGINAS SECUNDARIAS DE PROPÓSITOS ESPECIALES. SUPERIOR: TABLÓN DE ANUNCIOS. INFERIOR: CALENDARIO DE EVENTOS.....	60
ILUSTRACIÓN 33 - PÁGINA SECUNDARIA DE PROPÓSITO GENERAL.....	61
ILUSTRACIÓN 34 - COMPARACIÓN DE FRAMEWORKS PARA DISEÑO WEB MÓVIL. FUENTE: HTTPS://TRENDS.GOOGLE.COM/TRENDS/EXPLORE?DATE=TODAY%2012-M&Q=BOOTSTRAP%203,FOUNDATION%206	63
ILUSTRACIÓN 35 - VISTA DE LA NUEVA PÁGINA PRINCIPAL EN DISPOSITIVOS MÓVILES. IZQUIERDA: SMARTPHONE (360x640px). DERECHA: TABLETA (768x1024).	67
ILUSTRACIÓN 36 - VISTA DE LA PÁGINA PRINCIPAL COMPLETA EN PANTALLAS MÓVILES. IZQUIERDA: SMARTPHONE (360x640px), DIVIDIDO EN DOS. DERECHA: TABLETA (768x1024px).	68
ILUSTRACIÓN 37 - RESULTADOS DEL TEST DE OPTIMIZACIÓN PARA MÓVILES DE GOOGLE EN LA NUEVA VERSIÓN DEL PORTAL DE LA ESCUELA	69
ILUSTRACIÓN 38 - IZQUIERDA: DETALLE DE LA BARRA DE NAVEGACIÓN SUPERIOR DESPLEGADA EN MÓVIL. DERECHA: EJEMPLO DE PÁGINA SECUNDARIA EN MÓVIL CON BARRA DE NAVEGACIÓN PRINCIPAL PLEGADA, Y BARRA DE NAVEGACIÓN SECUNDARIA VISIBLE.	69
ILUSTRACIÓN 39 - COMPARACIÓN DE VOLUMEN DE DESCARGAS MEDIO DE UNA PÁGINA WEB EN 2010 (IZQUIERDA) FRENTE A 2017 (DERECHA). FUENTE: HTTP://HTTPARCHIVE.ORG/INTERESTING.PHP	71
ILUSTRACIÓN 40- DISTRIBUCIÓN DE VOLUMEN DE DESCARGAS PARA TEL.UVA.ES, EN FUNCIÓN DEL DOMINIO DE PROCEDENCIA. VALORES EN MEGABYTES.	73
ILUSTRACIÓN 41 - EJEMPLO DE COMPRESIÓN CON PÉRDIDAS DE IMÁGENES. IZQUIERDA: SIN COMPRIMIR, 252KB. DERECHA, COMPRESIÓN AL 75% DE CALIDAD, 48KB.	77
ILUSTRACIÓN 42 - PÁGINAS QUE USAN LA API DE LIBRERÍAS DE GOOGLE. FUENTE: HTTP://MOBILE.HTTPARCHIVE.ORG/INTERESTING.PHP#GOOGLELIBS	79
ILUSTRACIÓN 43 - FRAGMENTO DEL OBJETO "CONNECTOR" DE LA CONFIGURACIÓN DE TOMCAT QUE CONTROLA LA COMPRESIÓN GZIP. LA LÍNEA "COMPRESSION="ON" ACTIVA LA COMPRESIÓN GZIP.....	81
ILUSTRACIÓN 44 - RESUMEN DE LOS RESULTADOS DE TRAS LA OPTIMIZACIÓN DEL ANCHO DE BANDA.	82
ILUSTRACIÓN 45 - RESULTADOS DEL TEST DE RENDIMIENTO PARA TEL.UVA.ES (CARGA)	85
ILUSTRACIÓN 46 - GRÁFICAS DE USO DE GPU Y CPU A LO LARGO DEL TIEMPO. SE HAN RECORTADO LOS RESULTADOS HASTA LA MITAD DEL INTERVALO (90s) POR FALTA DE RELEVANCIA.....	87
ILUSTRACIÓN 47 - ÁRBOL DE LLAMADAS DEL PRIMER PICO DE CADA PAR.....	88
ILUSTRACIÓN 48 - ÁRBOL DE LLAMADAS DEL SEGUNDO PICO DE CADA PAR.	88
ILUSTRACIÓN 49 - DISTRIBUCIÓN DEL USO DE CICLOS DE CPU Y GPU EN ESTADO DE REPOSO.	88
ILUSTRACIÓN 50 - DISTRIBUCIÓN DEL USO DE CICLOS DE CPU Y GPU DURANTE LA CARGA, TRAS LA OPTIMIZACIÓN.	89
ILUSTRACIÓN 51 - DISTRIBUCIÓN DEL USO DE CICLOS DE CPU Y GPU EN ESTADO DE REPOSO TRAS LA OPTIMIZACIÓN.....	90
ILUSTRACIÓN 52 - RESULTADOS DE EFICIENCIA ENERGÉTICA DE TEL.UVA.ES EN WEB ENERGY ARCHIVE.	91

1: INTRODUCCIÓN Y CONOCIMIENTOS PREVIOS

1.1 – INTRODUCCIÓN. MOTIVACIÓN. OBJETIVOS.

La idea de este proyecto es la de crear un nuevo portal web para la E.T.S.I. de Telecomunicación, con dos objetivos principales. En primer lugar, se desea que el portal mejore al anterior en funcionalidad, presentación y estética, facilitando la navegación al usuario y buscando ofrecer la información de manera más clara y agradable.

La escuela dispone de un portal web propio desde el año 1997, en el cual se han llevado a cabo multitud de remodelaciones. Sin embargo, desde la remodelación del año 2005 apenas se han llevado a cabo modificaciones significativas. Un vistazo a la página principal delata rápidamente este hecho, dejando en evidencia principalmente varios aspectos:

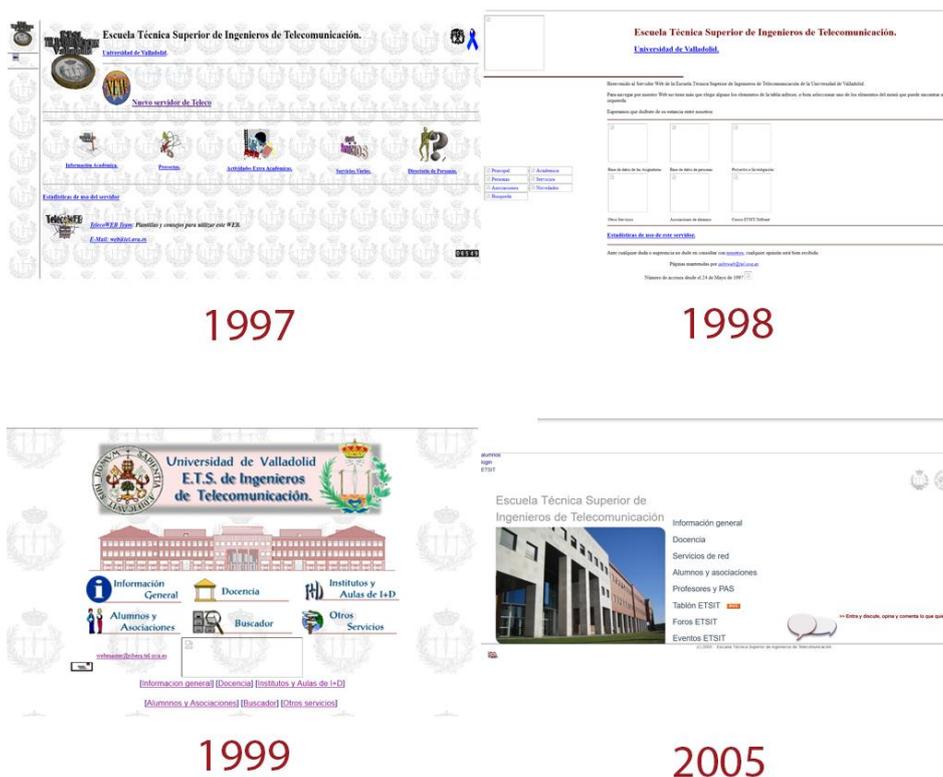


Ilustración 1 - Distintas versiones del portal de la Escuela a lo largo de los años.

Imágenes obtenidas de web.archive.org

Por un lado, la estética de la web ha quedado obsoleta con el paso de los años. Dispone de una presentación sobria y estática que no llama la atención al visitante ni invita a explorar los contenidos.

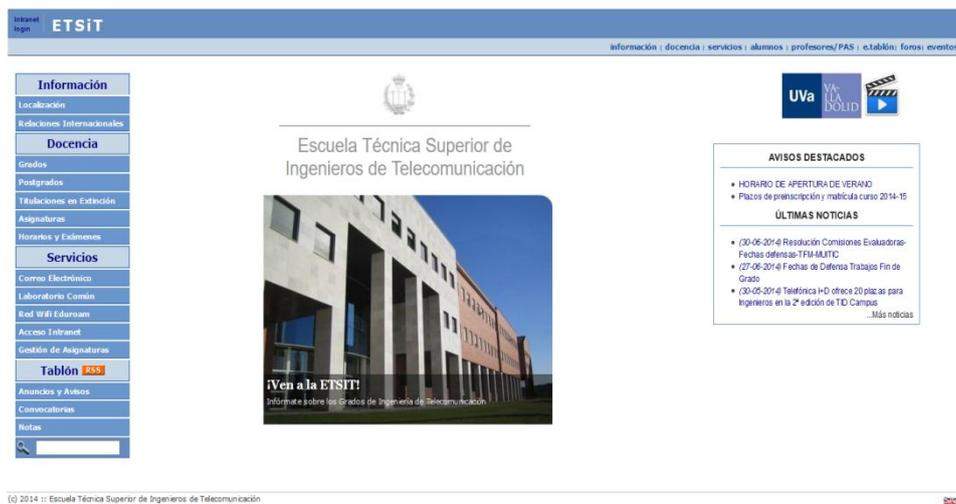


Ilustración 2 - Página principal del portal de la Escuela antes de la realización de este Proyecto

Por otra parte, los propios contenidos no están presentados de manera eficiente: no se identifican fácilmente las secciones de mayor interés para el visitante nuevo, y el visitante habitual no tiene muchas veces un acceso “directo” a las páginas con más tráfico, como son el área de asignaturas, o el tablón de anuncios

Se pretende dar un lavado de cara al portal, modernizando su aspecto y a su vez mejorando su usabilidad. Una de las funciones más importantes del portal web es la de servir como punto de acceso a nuevos visitantes que quieran conocer la Escuela, atrayendo incluso a potenciales alumnos o colaboradores. Una web con estética moderna y atractiva mantiene la atención del usuario, que puede ser entonces dirigida hacia los contenidos importantes para ellos gracias a una estructura más manejable y vistosa.

También se colaborará directamente con personal de la Escuela para implementar cambios y mejoras funcionales sugeridas, y adecuar el aspecto visual a la imagen pública que la Junta Directiva de la Escuela desea para la misma.

La segunda motivación principal surge de mi participación en el concurso internacional “Green Code Lab Challenge”, en el que varios compañeros y yo representamos a la E.T.S.I.T. de la Universidad de Valladolid. El objetivo del concurso es fomentar la programación “verde”, es decir, energéticamente eficiente. Dentro de este contexto, surge la idea de crear un portal “verde” para la Escuela. Se quiere lograr, renunciando en la menor medida posible a las mejoras en el rediseño estético y funcional del portal, reducir el consumo energético derivado del servicio y uso del mismo. Con esto pretendo no sólo conseguir un resultado positivo en cuanto al gasto económico que supone el mantenimiento del portal web, sino también lograr un menor impacto en el medio ambiente.

Como se verá en el capítulo dedicado a la presentación del problema, el ingente crecimiento en la cantidad de páginas y servidores web suponen una importante fuente de contaminación medioambiental debido, en gran parte, al alto consumo energético derivado de su funcionamiento, si bien también producen una huella a través de otros sistemas: La

refrigeración de las salas de servidores, o el consumo de batería y datos en dispositivos móviles que visualizan sus páginas web son algunos ejemplos del impacto medioambiental de la red.

Es necesario destacar que la modernización del portal web concebida como se ha descrito en la primera parte de esta sección, inevitablemente va a redundar en el aspecto energético. Añadir funcionalidades supone casi siempre una mayor cantidad de código a ejecutar tanto en el servidor como en el cliente, que a su vez produce un mayor uso de ciclos de procesador y de la memoria RAM y por tanto mayor consumo energético. Además, la extrema sencillez del portal anterior significaba un uso muy reducido de ancho de banda así como de recursos en el servidor y en el cliente, por lo que el objetivo de mejorar sus cifras de forma absoluta sería poco, o nada realista.

El foco del trabajo por tanto no puede dividirse entre ambos aspectos, sino que deberá buscar siempre un compromiso entre uno y otro, mejorando funcionalidad sin perder de vista el aspecto “verde”, y viceversa.

Con todo lo anterior en cuenta, el objetivo final se puede resumir en crear un portal web más moderno, más usable, responsable y ético con el medio ambiente, que potencie la presencia en internet de la Escuela y facilite la navegación a usuarios tanto nuevos como habituales.

1.2 – CONOCIMIENTOS PREVIOS

Durante mis estudios en la carrera de Ingeniería de Telecomunicación y el Grado en Ingeniería de Tecnologías de Telecomunicación, ambas cursadas en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid, he adquirido una serie de conocimientos y habilidades relacionados con el tema de este Trabajo de Fin de Grado que pondré en práctica.

En la asignatura Tecnologías para Aplicaciones Web se imparten conocimientos sobre desarrollo y diseño de páginas web, particularmente con el uso de HTML, hojas de estilo en cascada y Javascript. También diseño e implementación bases de datos relacionales con SQL, y desarrollo de aplicaciones web que interactúen con dichas bases de datos (en esta asignatura mediante PHP, si bien el portal web de la Escuela emplea Java para este propósito).

Cursé esta asignatura durante mi etapa en Ingeniería de Telecomunicación antes de empezar a estudiar el Grado de Ingeniería de Tecnologías de Telecomunicación, y he continuado desarrollando y ampliando estos conocimientos en proyectos personales desde entonces.

Otras asignaturas en las que aprendí conocimientos que han sido útiles en la realización de este Trabajo fueron Sistemas de Comunicación I y Teletráfico y Gestión, de la Ingeniería de Telecomunicación, e Ingeniería de Sistemas Software del Grado en Ingeniería de Tecnologías de Telecomunicación.

En Sistemas de Comunicación I se impartieron conocimientos sobre aplicaciones web, la Web 2.0, programación orientada a objetos y desarrollo en Java, incluido desarrollo de interfaces gráficas.

Teletráfico y Gestión incluía temario sobre servidores de internet, además de emplear, siendo la primera asignatura de la carrera en introducirlo, un sistema de control de versiones (CVS).

Ingeniería de Sistemas Software basaba su temario en el diseño y planeamiento de proyectos de software y programación en Java, así como nociones básicas de la metodología ágil.

Si bien estas fueron las asignaturas en las que aprendí los conocimientos más directamente aplicables a este Trabajo, hubo un gran número de asignaturas en las que adquirí habilidades transversales y conocimientos aplicables de una u otra manera, demasiadas como para enumerar aquí.

La competición de programación internacional Green Code Lab Challenge, en la que participan alumnos de la E.T.S.I.T., me aportó una base en cuanto a la computación energéticamente eficiente, o computación verde, con la que no había tenido ningún contacto anterior. Mi participación durante dos años me permitió explorar y desarrollar un conocimiento de las técnicas empleadas en esta corriente de programación, llegando a alcanzar junto con mis compañeros el tercer puesto de la competición por delante de estudiantes de universidades de toda Europa. Particularmente, el objetivo de la competición ese año consistió en aplicar las técnicas de programación verdes para optimizar una página web, evento del cual surgió la idea para el presente Trabajo.

Realicé unas prácticas en empresa en el verano de 2014, en la empresa de ingeniería acústica Audiotec. En ellas trabajé desarrollando una aplicación de escritorio en Visual Basic para automatizar la creación de informes para clientes. Estas prácticas me aportaron experiencia trabajando con software real diseñado para usuarios reales, en contraposición a los proyectos de la carrera creados con fines académicos y que, difícilmente, hubieran tenido algún uso práctico fuera de las clases. También aprendí nociones básicas de diseño de interfaces de usuario.

1.3 – ANÁLISIS DEL PROBLEMA Y ESTADO DEL ARTE

1.3.1 - DESCRIPCIÓN GENERAL DEL PORTAL WEB

1.3.1.1 – ARQUITECTURA

El portal web de la E.T.S.I.T. se aloja en un servidor perteneciente a la Escuela, situado dentro de la misma, y es accesible a través de la URL *www.tel.uva.es*. El servidor emplea un sistema operativo basado en Linux y usa como servidor web el programa Apache Tomcat, un contenedor de servlets de código abierto desarrollado por Apache Software Foundation, que hace las veces de Servidor HTTP y compilador de código Java.

Los servlets son clases de Java que se ejecutan en servidores web extendiendo sus funcionalidades, en lugar de en un navegador (*applets*), funcionando como el sustituto de otros lenguajes de contenido dinámico Web como ASP.NET o PHP. Tomcat, implementando las especificaciones de Oracle Corporation, compila archivos JavaServer Pages (JSP) para convertirlos en servlets, y a continuación puede ejecutarlos de manera local para ampliar las funcionalidades de su servidor HTTP. Los datos del portal, incluido el contenido de algunas páginas, se almacenan en una base de datos relacional gestionada con PostgreSQL, un servidor de bases de datos de código abierto.

El portal está construido usando el *framework* Struts, de Apache, diseñado para facilitar la creación de aplicaciones web que usan la tecnología de JavaServer Pages (JSP). Struts proporciona un marco con el que integrar la arquitectura MVC (Model, vista y controlador) en la aplicación web para separar el código relativo al control de la base de datos, el diseño de la página, y el control de flujo.

Las tareas de mantenimiento y actualización del portal son llevadas a cabo, en gran medida, por Carlos Alonso Gómez, secretario académico de la Escuela y Analista de Red de la misma. La construcción del portal tal como existe previamente a la realización de este Trabajo de Fin de Grado fue realizada en 2005 por el alumno Ignacio Clérigo, como Proyecto Fin de Carrera. La arquitectura del servidor y de sus páginas permanece inalterada desde entonces, si bien el contenido sí ha sufrido modificaciones.

El contenido de la página principal está incrustado totalmente en el código, salvo las secciones de las barras de navegación y el tablón de anuncios. Estas secciones tienen un espacio definido, inicialmente vacío de contenido. El contenido se genera durante la compilación de los archivos .jsp, extrayendo la información necesaria de la base de datos y generando el código HTML que ocupará la sección.

Las páginas secundarias, salvo secciones excepcionales como el calendario de eventos, se generan a través de una plantilla común que genera su contenido durante la compilación. Dicha plantilla contiene un esqueleto de la página, concretamente secciones comunes a todas ellas como la cabecera y el pie de la página y la barra de navegación. El contenido específico de cada sección, generado al compilar, puede obtenerse de dos fuentes diferentes:

- Escrito en un archivo .jsp específico para dicha sección. Para crear estas páginas es necesario al menos un mínimo conocimiento de HTML/CSS y, por tanto, suelen ser creadas y mantenidas por el analista de red de la Escuela.

- Leído desde la base de datos. El portal dispone de una sección para personal interno, desde la que se pueden crear secciones a través de un Sistema Gestor de Contenidos (CMS). La creación de una página mediante el CMS no requiere de conocimientos de HTML y por tanto estas páginas suelen ser creadas por personal de la Escuela.

Con el fin de no interferir en el buen funcionamiento de todas las páginas del portal, será necesario que el rediseño sea compatible con el CMS interno, de manera que los nuevos estilos y funcionalidades no interfieran con el código generado por el mismo en un futuro.

1.3.1.2 - DISEÑO



Ilustración 3 - Página principal del portal de la Escuela antes de la realización de este Proyecto

La página principal se compone de una barra superior y tres columnas que organizan el contenido. En la barra superior se encuentra una barra de navegación con enlaces a las secciones principales del portal. En la columna izquierda se encuentra una lista de enlaces a algunas secciones de la web que no aparecen en la barra de navegación haciendo las veces de una barra de navegación secundaria. En la columna central, la de mayor anchura, hay una imagen en formato JPG de la fachada principal de la Escuela en el centro de la página. Sobre la imagen, el nombre de la Escuela y su escudo. En la columna derecha hay un resumen de los últimos anuncios del tablón, así como un enlace al portal de la Universidad de Valladolid (www.uva.es) y otro al vídeo corporativo de la Escuela alojado en el portal de videos online Youtube.com.

El contenido de las páginas ocupa todo el ancho del navegador, extendiéndose desde el extremo izquierdo hasta el derecho, y dejando vacíos (en blanco) los espacios entre los elementos descritos.

Los colores predominantes en los elementos de la página son:

- Azul claro, empleado como fondo en la barra de navegación y en la barra lateral izquierda.
- Azul oscuro, empleado como color de texto en zonas con colores de fondo claros.
- Blanco, empleado como color de fondo en el resto de la página, y como color de texto sobre colores de fondo oscuros.
- Gris claro, empleado como color de texto para el nombre y escudo de la Escuela.

Las páginas secundarias presentan, a grandes rasgos, una estructura similar a la principal, con una cabecera en la parte superior (que incluye la barra de navegación principal), y una barra de navegación vertical secundaria en la parte izquierda de la pantalla. El contenido individual de la página se sitúa bajo la cabecera, ocupando todo el espacio libre a la derecha de la barra de navegación secundaria.

ETSIT | Información | docencia | servicios | alumnos | profesores/PAS | e.tablón | foros | eventos

Historia

Presentación

La Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid, se creó el 27 de septiembre 1991. Tiene como misión básica la formación científica y técnica y la preparación para el ejercicio profesional de Ingenieros que han de realizar las funciones que la sociedad actual demanda. Además está comprometida con la creación y desarrollo de la ciencia y de la técnica mediante la investigación y el estudio, colaborando con las empresas en el desarrollo tecnológico del país y de la Comunidad de Castilla y León, contribuyendo de esta forma a la consolidación de Sociedad de la Información.

La Escuela se encarga asimismo, de la gestión administrativa y de la organización de las enseñanzas para la obtención de títulos de Grado y Máster en el ámbito de la Ingeniería de Telecomunicación, coordinando la docencia que imparten los Departamentos de acuerdo con los Planes de Estudio y los objetivos de las titulaciones. Cuenta para ello con unos recursos humanos que, junto con unas instalaciones nuevas y actualizadas, permiten cumplir sus objetivos al nivel de las Escuelas más prestigiosas del país.

La ETSIT, cumpliendo su principio fundacional y a pesar de su juventud, ha estado integrada en todo momento en la Universidad de Valladolid y en la sociedad Castellano-Leonesa, contribuyendo a su desarrollo como lo demuestra su pertenencia a foros de investigación, desarrollo y potenciación de las telecomunicaciones y de la electrónica, tanto nacionales como internacionales, implicándose en particular en la potenciación del mundo empresarial regional, estando presente allí donde las necesidades de estos sectores lo demandan.

- Ven a la ETSIT (presentación en Vídeo)

Titulaciones impartidas

- Grado en Ingeniería de Tecnologías de Telecomunicación (50 plazas de nuevo acceso)
- Grado en Ingeniería de Tecnologías Específicas de Telecomunicación (130 plazas de nuevo acceso)
 - Mención en Ingeniería de Sistemas de Telecomunicación
 - Mención en Ingeniería de Sistemas Electrónicos
 - Mención en Ingeniería Telemática
- Máster en Ingeniería de Telecomunicación
- Ingeniero en Electrónica (2º ciclo)(en proceso de extinción)
- Ingeniero de Telecomunicación (en proceso de extinción)
- Máster en Investigación en Tecnologías de la Información y las Comunicaciones (40 plazas de nuevo acceso)
- Doctorado en Tecnología de la Información y las Telecomunicaciones

(c) 2014 :: Escuela Técnica Superior de Ingenieros de Telecomunicación

Ilustración 4 - Una página secundaria del portal de la escuela antes de la realización de este Proyecto

1.3.2 - PRESENTACIÓN DE LA INFORMACIÓN

El primer paso a la hora de analizar el portal web original consistió en determinar qué información está disponible en las páginas (tanto en la principal como en todas las secundarias), que función cumple y cómo está situada.

En la cabecera de la página principal se encuentran enlaces de navegación a las secciones de nivel superior o principales: información, docencia, servicios, alumnos, profesores/PAS, e.tablón, foros, y eventos. Para acceder a las secciones de nivel secundario (las que están anidadas dentro de las de nivel superior) es necesario hacer clic en alguna de las secciones de nivel superior, visitar la página correspondiente, y en ella emplear una barra de navegación situada en la parte izquierda de la pantalla, que ofrece los enlaces a las secciones de niveles secundarios.

Desde la página principal se puede acceder al área de login. Existen enlaces externos al portal de la Universidad de Valladolid, al vídeo de la Escuela en Youtube, y a la versión en inglés del portal de la Escuela. En la parte izquierda de esta página existe una zona con enlaces internos a algunas áreas secundarias del portal, localizadas dentro de las secciones principales de información, docencia, servicios y tablón. La zona derecha de la página la ocupa una vista previa del tablón de anuncios.

Además de la disposición de la información, es necesario analizar también el comportamiento de los visitantes. Un lector no actúa como una variable independiente dentro de nuestro portal, su comportamiento depende de factores internos y externos que podemos dividir para estudiar en dos ámbitos principales: sus hábitos de navegación, y su familiarización con el portal de la Escuela.

¿Usuario novel, o habitual?: Un lector habitual no visita las mismas páginas que uno primerizo. El lector educado conoce a priori las páginas que le interesan y donde localizarlas. A menudo visita directamente esas páginas a través de los marcadores de su navegador, y cuando lo hace a través de la página principal, suele conocer de antemano a dónde dirigirse. Concretamente, un usuario habitual de nuestro portal pertenece en la mayoría de los casos a uno de dos grandes grupos: Personal de la Escuela y alumnos.

Para estos usuarios es importante mantener, fácilmente alcanzables, enlaces a las páginas internas más visitadas, ya que estadísticamente serán más propensos a entrar al portal buscando esos enlaces. Además, al estar involucrados en el día a día de la Escuela son los mayores interesados en las noticias y novedades relacionadas con la misma y con los eventos en los que toma parte.

Los visitantes noveles, o que han visitado el portal pocas veces, no conocen la estructura de la página, y por norma general no acceden directamente a sus páginas objetivo, sino que necesitan encontrarlas siguiendo enlaces desde la página principal. Los visitantes de estas características suelen formar un conjunto más heterogéneo, si bien se pueden aislar grupos importantes dentro del conjunto, como puede ser el de potenciales nuevos alumnos.

Para este grupo de usuarios existe una preocupación adicional, y es que el portal web es uno de los puntos de contacto más importantes entre la Escuela y el posible alumno, por ser el más fácilmente accesible. Cobra necesidad identificar las páginas con información más relevante para dichas personas y situarla de manera accesible para ellos.

Hábitos de navegación: El usuario medio de internet visita alrededor de 60 portales web diferentes al mes, y más de 2500 páginas únicas durante ese periodo (OFcom, 2015). Durante este proceso se somete a una continua exposición de estructuras, diseños y funcionalidades a las que acaba adaptándose por costumbre. De la misma manera, los portales que visita se adaptan progresivamente a los hábitos de navegación de sus usuarios. Este proceso converge en un Internet en el que los hábitos de los usuarios se han homogeneizado a medida que lo hacían las páginas que visitaban. El resultado actual es que los usuarios están *entrenados* para buscar el mismo tipo de información en las mismas áreas de las páginas que visita, y los portales se adaptan colocando la información donde saben que sus usuarios la buscarán.

Varios estudios, entre ellos los realizados por el NielsenNorman Group de consultoría, y la universidad de Wichita en Kansas, han determinado los principales caminos que sigue el ojo del lector visitando una página (Nielsen, F-Shaped Pattern For Reading Web Content, 2006), así como dónde busca, instintivamente determinados pedazos de información en su pantalla (Bernard, 2010) cuando visita una página nueva. Este comportamiento es fruto y causante a la vez, de la tendencia mencionada a que las páginas web coloquen los elementos con propósitos similares en lugares similares dentro de su estructura. El desarrollador dispone, gracias a estos estudios, de una donde colocar los elementos más comunes de su página, de manera que cualquier usuario los encuentre de manera rápida e intuitiva.

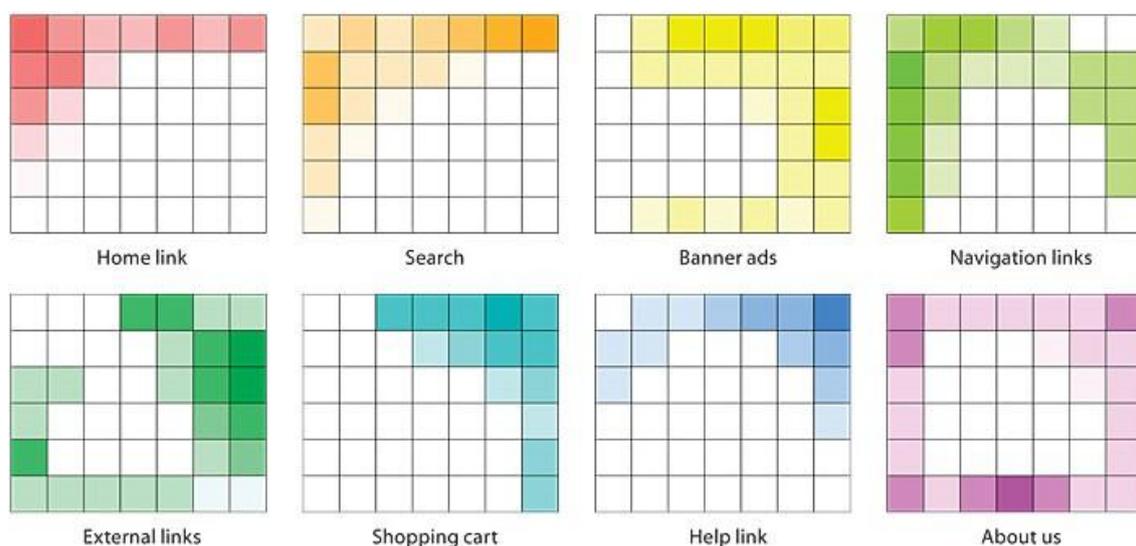


Ilustración 5 - Lugares donde los usuarios esperan encontrar componentes típicos de una página web.

Fuente: (Lynch, 2016), basado en (Bernard, 2010)

Estos estudios también han cuantificado la importancia de la correcta colocación de los objetos de contenido en el flujo de la página, en cuanto a su situación por encima o por debajo de los límites visibles al cargar dicha página, el llamado *page fold*. Una investigación llevada a

cabo por Jakob Nielsen y el NN Group llegó a la conclusión de que “los usuarios de internet pasan el 80% del tiempo mirando la información situada sobre el *page fold*. Aunque hacen *scroll* hacia abajo, solamente dedican un 20% de su atención a lo que se encuentra por debajo de dicho límite” (Nielsen, Scrolling and Attention, 2010).

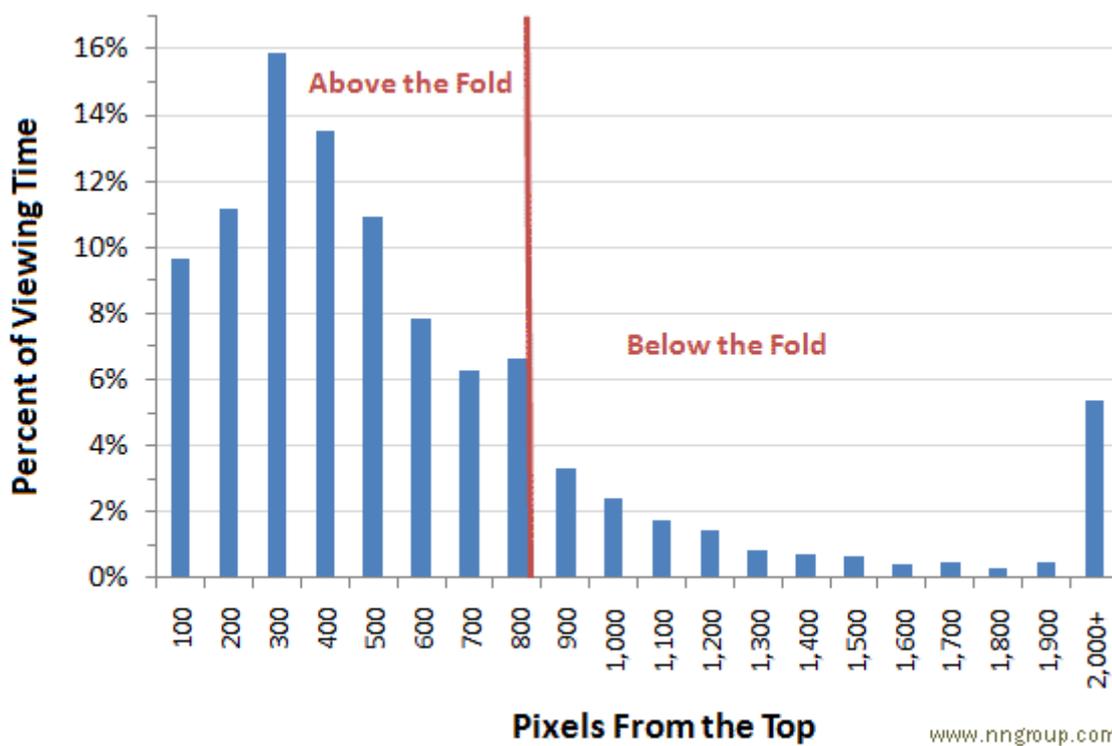


Ilustración 6 - Porcentaje de tiempo empleado mirando contenido en función de los píxeles desde el inicio. En rojo, límite de contenido visible sin hacer scrolling. (Nielsen, Scrolling and Attention, 2010)

Conociendo toda esta información acerca de nuestros visitantes, pretendemos que el nuevo portal disponga la información en una estructura clara e intuitiva que permita a los usuarios, tanto nuevos como habituales, navegar de forma natural y sencilla, e invite a visitarla de nuevo. Para ello es necesario conocer qué información interesa a los diversos grupos de usuarios y de qué manera se puede presentar esta información para conseguir llegar al público. Se ha procedido a investigar cuales son las páginas más visitadas por los grupos de usuarios, para realizar un diseño que destaque dichas páginas a la vez que aprovecha los hábitos de navegación generales para presentar la información estructurada como se ha descrito anteriormente.

Para obtener los datos acerca de las páginas más visitadas dentro del portal, se ha empleado la herramienta GoogleAnalytics de Google. Tras suscribirse en el programa de GoogleAnalytics, el desarrollador debe integrar un script de Javascript dentro de todas las páginas de su portal que desee incluir dentro de las estadísticas. Automáticamente, el sistema comenzará a recopilar todo tipo de información acerca del tráfico y los visitantes recibidos. La herramienta es gratuita en su versión básica, y permite añadir funcionalidades mediante una suscripción de pago. El portal de la Escuela lleva recopilando información desde antes de la realización de este Trabajo, por lo que se disponen de datos de un amplio espacio temporal.

Los datos más importantes entre los analizados para este Trabajo son los relacionados con las páginas más visitadas y la distribución de los usuarios entre habituales y nuevos. Con estos datos se puede obtener una idea general de qué clase de usuarios visitan las páginas, y que información es la más buscada. También se analizará el uso de dispositivos móviles para visualizar el portal.

	Usuarios activos en un día	Usuarios activos durante 7 días	Usuarios activos durante 14 días	Usuarios activos durante 30 días
Todos los usuarios	104 % del total: 100,00 % (104)	672 % del total: 100,00 % (672)	1.308 % del total: 100,00 % (1.308)	2.531 % del total: 100,00 % (2.531)
Usuarios de una única sesión	91 % del total: 87,50 % (104)	526 % del total: 78,27 % (672)	943 % del total: 72,09 % (1.308)	1.827 % del total: 72,18 % (2.531)
Usuarios de varias sesiones	13 % del total: 12,50 % (104)	146 % del total: 21,73 % (672)	365 % del total: 27,91 % (1.308)	704 % del total: 27,82 % (2.531)

Tabla 1 - Desglose de usuarios divididos entre nuevos (una única sesión) y habituales (varias sesiones). Google Analytics

Se puede observar en la tabla sobre estas líneas cómo los visitantes habituales son mucho más numerosos que los nuevos, con un 72,17% del total de visitantes frente al 27,82% del segundo grupo.

Página ?	Número de visitas a páginas ?	Página ?	Número de visitas a páginas ?
Usuarios nuevos	63.318 % del total: 26,48 % (239.110)	Usuarios recurrentes	175.792 % del total: 73,52 % (239.110)
1. /	13.211 (20,86 %)	1. /	31.638 (18,00 %)
2. /tablon/avisos.htm	3.158 (4,99 %)	2. /docencia/asignaturas.htm	11.386 (6,48 %)
3. /docencia/horarios.htm	2.942 (4,65 %)	3. /docencia/horarios.htm	10.752 (6,12 %)
4. /docencia/asignaturas.htm	2.865 (4,52 %)	4. /tablon/avisos.htm	7.229 (4,11 %)
5. /docencia/planes/masterbds.htm	2.092 (3,30 %)	5. /docencia/asignaturas.htm?controlador(titulacion)=Pcomun	7.174 (4,08 %)
6. /docencia/asignaturas.htm?controlador(titulacion)=Pcomun	1.725 (2,72 %)	6. /docencia/asignaturas.htm?controlador(titulacion)=null&controlador(asignatura)=null	3.557 (2,02 %)
7. /docencia/grados.htm	1.636 (2,58 %)	7. /docencia/asignaturas.htm?controlador(titulacion)=P460b	3.350 (1,91 %)
8. /informacion/localizacion.htm	1.550 (2,45 %)	8. /docencia/asignaturas/notas.htm?controlador(titulacion)=Pcomun&controlador(asignatura)=A45002	2.669 (1,52 %)
9. /informacion/historia.htm	1.028 (1,62 %)	9. /tablon/notas.htm	2.626 (1,49 %)
10. /docencia/asignaturas.htm?controlador(titulacion)=null&controlador(asignatura)=null	954 (1,51 %)	10. /docencia/asignaturas/notas.htm?controlador(titulacion)=Pcomun&controlador(asignatura)=A45000	2.427 (1,38 %)

Tabla 2 - Páginas más visitadas por usuarios nuevos (izquierda) y recurrentes (derecha). Google Analytics.

De entre los usuarios nuevos, las páginas más buscadas son las relacionadas con noticias (el tablón de avisos), con información acerca de la Escuela (Localización e Historia), y con información sobre los planes impartidos (Máster, Asignaturas y Grados).

La información que más interesa a los usuarios habituales es la relacionada con las asignaturas que cursan: sus horarios, sus notas, y sus recursos. Existe un gran número de visitas a las diferentes páginas derivadas de /docencia/asignaturas, que se corresponden con las listas de asignaturas de los diferentes estudios de Grado Y Máster impartidos. También prestan especial atención al tablón de avisos.

Categoría de dispositivo [?]	Adquisición			Comportamiento		
	Sesiones [?] ↓	% de nuevas sesiones [?]	Usuarios nuevos [?]	Porcentaje de rebote [?]	Páginas/sesión [?]	Duración media de la sesión [?]
	56.367 % del total: 100,00 % (56.367)	34,09 % Media de la vista: 34,02 % (0,19 %)	19.214 % del total: 100,19 % (19.178)	38,34 % Media de la vista: 38,34 % (0,00 %)	4,24 Media de la vista: 4,24 (0,00 %)	00:04:56 Media de la vista: 00:04:56 (0,00 %)
1. desktop	33.899 (60,14 %)	37,88 %	12.842 (66,84 %)	34,35 %	4,56	00:05:02
2. mobile	20.738 (36,79 %)	26,87 %	5.573 (29,00 %)	44,29 %	3,74	00:04:52
3. tablet	1.730 (3,07 %)	46,18 %	799 (4,16 %)	45,32 %	3,94	00:03:46

Tabla 3 - Desglose de visitas al portal en función del dispositivo empleado. Desktop: Ordenador de escritorio. Mobile: Teléfono móvil. Tablet: tableta. Google Analytics.

En cuanto a los dispositivos utilizados para navegar dentro del portal, se observa como los dispositivos móviles (teléfonos y tabletas) suponen cerca de un 40% del tráfico total. El porcentaje de rebote (la cantidad de usuarios que visitan una página única y a continuación abandonan el portal) es cerca de un 10% superior para los usuarios de dispositivos móviles. Este dato, unido a la ausencia de una versión del portal adaptada para móviles, hace pensar que cierto porcentaje de usuarios pueden estar abandonando el portal por el hecho de no contar con una página adaptada.

Los datos recopilados por GoogleAnalytics servirán para valorar las necesidades del nuevo diseño del portal en futuras secciones.

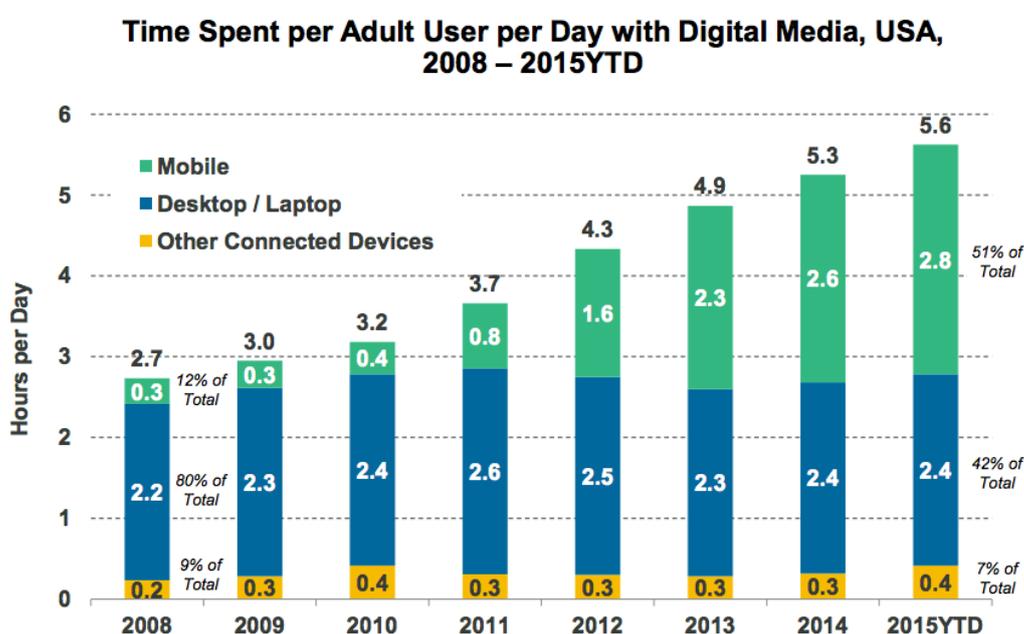
1.3.3 - ESTADO DEL ARTE: DISEÑO WEB ADAPTABLE

La estructura y apariencia de la página de la Escuela es la misma independientemente del dispositivo desde el que se acceda (PC, *Tableta*, *Smartphone*). Al visualizar la página principal desde una pantalla pequeña (*Tableta* o *Smartphone*), los contenidos se adaptan al ancho de la pantalla, de manera que para poder leer textos o pulsar en enlaces es necesario hacer *zoom* sobre la zona en cuestión. Previamente a la popularización de los smartphones, este era el comportamiento usual de las páginas web en pantallas pequeñas, como las de los móviles con sistemas operativos basados en Java.

Durante la década de los 2000, el desarrollador web se debió enfrentar al problema de diseño que suponía la diversidad de navegadores web existentes, y la disparidad en el comportamiento de algunos de ellos (especialmente Internet Explorer) a la hora de *renderizar* una página web e implementar nuevas funcionalidades de CSS y Javascript.

La solución a este problema generalmente consistía en crear y mantener varias versiones de la página web en cuestión, adaptadas específicamente para cada uno de los navegadores objetivo.

Internet Usage (Engagement) Growth Solid
+11% Y/Y = Mobile @ 3 Hours / Day per User vs. <1 Five Years Ago, USA



@KPCB Source: eMarketer 9/14 (2008-2010), eMarketer 4/15 (2011-2015). Note: Other connected devices include OTT and game consoles. Mobile includes smartphone and tablet. Usage includes both home and work. Ages 18+; time spent with each medium includes all time spent with that medium, regardless of multitasking. 14

Ilustración 7 - Tiempo empleado por adulto y día en contenidos digitales, por tipo de dispositivo. En verde, móviles. En azul, ordenadores. En amarillo, otros. (Meeker, 2017)

A finales de la década de los 2000 se había extendido el uso de smartphones hasta tal punto, que las compañías empezaron a prestar atención a la apariencia de sus páginas web en estos dispositivos. Durante los años siguientes el crecimiento del uso de móviles para navegar por internet ha tenido un gran crecimiento, llegando incluso a desbancar al ordenador como el medio más utilizado, con un 51% del total de horas dedicadas a navegar. (Meeker, 2017)

La práctica más extendida entonces era similar a la que se ponía en práctica con el problema de los navegadores: crear una versión de la página web para cada tipo de dispositivo al que se quisiera adaptar. El servidor web detectaría el tipo de dispositivo desde el que se realizó la petición del documento y consecuentemente serviría la versión del documento diseñada para el tipo de dispositivo en cuestión.

Esta solución permitía al usuario navegar cómodamente desde su Smartphone en páginas adaptadas para pantallas pequeñas, pero implicaba que el desarrollador debía mantener varias versiones de la misma página simultáneamente. Los cambios debían ser introducidos y probados en cada una de las versiones y el trabajo se multiplicaba. Además, la práctica de servir un documento en función del dispositivo y navegador que lo solicitara dejó de ser práctica a medida que crecía la gama de dispositivos desde los que se accedía a internet: el usuario ya no solo navega con su Smartphone, su portátil o su PC. Surgen nuevos dispositivos como las tabletas, y otros como los televisores o los lectores de libros electrónicos comienzan a implementar acceso a internet y un navegador (al que también hay que dar soporte).

Rápidamente deja de ser factible mantener y actualizar semejante cantidad de código y empiezan a surgir soluciones inteligentes basadas en la detección de características, que cambian la manera en la que se crean las páginas web. En cuanto a la disparidad entre navegadores, aparecen librerías Javascript conocidas como *polyfill*, que permiten emular funcionalidades que no implementa el navegador que está ejecutando su código. Si la librería *polyfill* detecta que cierta característica de HTML, CSS o Javascript no es implementada por el navegador, se pone en marcha un sistema para emular dicha funcionalidad en caso de ser necesario, o se ejecuta un código alternativo de respaldo (*fallback*) escrito por el desarrollador, que sustituye al que no ha podido ser ejecutado.

Estas librerías permiten al desarrollador mantener una única versión de su página web, común para todos los navegadores, con un esfuerzo mínimo en comparación al que suponían técnicas anteriores. En esa versión única debería, igualmente, ocuparse de establecer el comportamiento de las secciones que puedan tener defectos de implementación. Sin embargo, mientras que anteriormente debía escribir una solución para cada navegador, ahora la librería automatiza gran parte del trabajo y la parte del desarrollador es mínima.

Paralelamente a esto, se empieza a adaptar la misma filosofía en cuanto a la adaptación a los diferentes tipos de dispositivos. De igual manera que se pasa de detectar navegadores a detectar las características que implementan los navegadores, en esta área se pasa de responder a diferentes tipos de dispositivos, a detectar las características de dichos dispositivos (principalmente la resolución de su pantalla).

En 2010, la publicación online sobre diseño web *A List Apart* divulga el reporte Responsive Web Design (Marcotte, 2010) de Ethan Marcotte. En su artículo, Marcotte enumera los

problemas existentes ya mencionados en el área del desarrollo web para móviles. A continuación presenta una nueva iteración de la filosofía de *detectar* características y responder a ellas. A través del módulo Media Queries de CSS, se crea un diseño fluido que adapta la estructura de la página a las dimensiones (en píxeles) de la pantalla. Este reporte se convertiría rápidamente en la piedra angular del desarrollo web para dispositivos móviles.

Media Queries permite especificar reglas CSS que se activarán dependiendo de si el medio de visualización cumple unas características u otras. En Responsive Web Design, Marcotte muestra cómo, usando Media Queries que dependen del ancho en píxeles de la pantalla, varía la estructura de la misma, los elementos se colocan de manera diferente y permiten al lector navegar de forma natural al no necesitar hacer zoom sobre textos o enlaces.

En la actualidad los usuarios de internet navegan más horas desde sus teléfonos móviles y tabletas que desde sus ordenadores. La corriente de diseño RWD prevalece sobre las otras técnicas de diseño web para múltiples dispositivos, y está presente en gran cantidad de páginas web. Tanto es así que, en abril de 2015, el gigante tecnológico Google presenta su nuevo algoritmo de clasificación de páginas web en su buscador, con una fuerte tendencia a favorecer las páginas mobile-friendly. El nuevo algoritmo introduce un parámetro en el análisis de una página web, que indica si la misma está adaptada para dispositivos móviles. El parámetro es booleano de tal manera que una página solo puede entrar en la categoría de mobile-friendly si cumple con todos los requisitos. En caso de que falle en alguno, será considerada no adaptada a móviles y sufrirá una fuerte penalización en el ranking de búsqueda, haciendo que aparezca por detrás de otras que sí que han pasado el test favorablemente. Todo esto merma la exposición al público de dicha página web.

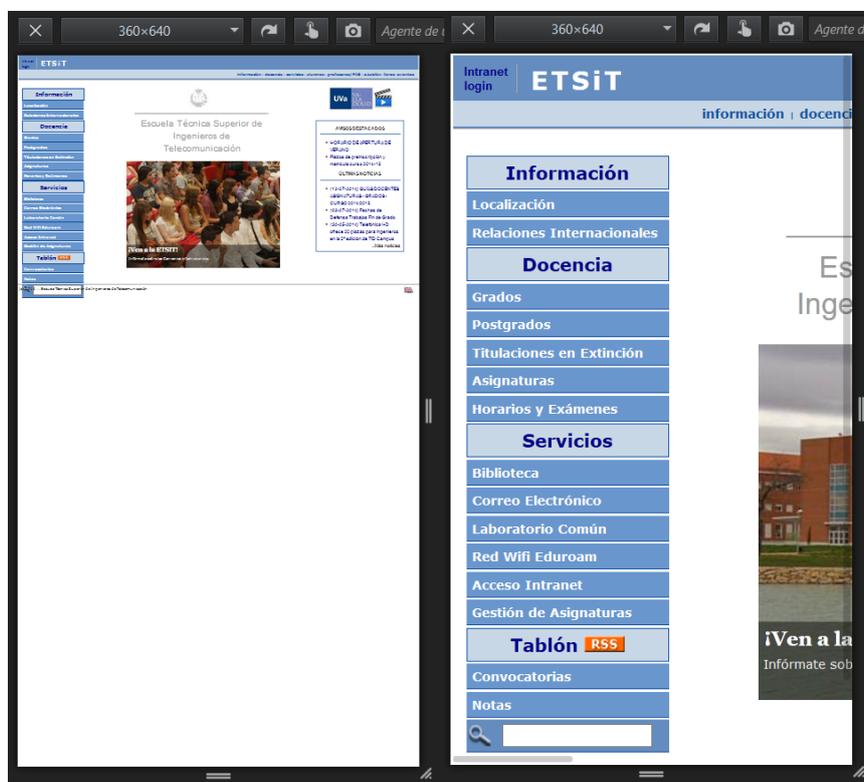


Ilustración 8 - Página principal del portal antiguo de la Escuela visto en pantalla de móvil de 360x640px. Izquierda: sin zoom. Derecha: con zoom

El portal web de la Escuela, como se describió anteriormente, no dispone de ninguna de las técnicas mencionadas para facilitar la navegación en pantallas pequeñas. La página se deforma para adaptarse al ancho de la pantalla sin cambiar su estructura, diseñada en un principio para las anchas pantallas de los ordenadores de sobremesa. El contenido acaba por tanto con un tamaño extremadamente reducido, lo que hace que sea difícilmente legible sin acercar y alejar el zoom constantemente.

Google pone a disposición de los desarrolladores una serie de herramientas que permiten valorar el desempeño de sus páginas web en diversos campos, incluida una para valorar si una página está correctamente adaptada a dispositivos móviles. En su estado actual (anterior a los cambios fruto de este Trabajo de Fin de Grado), el portal de la Escuela no cumple con los requisitos del algoritmo y está siendo penalizada bajando posiciones en las búsquedas realizadas a través de dispositivos móviles (esto no tiene efecto en búsquedas en ordenadores de escritorio).

La herramienta se encuentra disponible al público en <https://search.google.com/test/mobile-friendly>, y permite introducir la URL de la página web a analizar. Tras unos minutos, la herramienta mostrará al desarrollador los resultados del análisis, detallando los puntos en los que su página ha fallado el test y ofreciendo sugerencias para su optimización.



The screenshot shows the Google Mobile-Friendly Test results for a website. At the top, a warning icon and the text "Se han producido problemas al cargar la página VER DETALLES" are visible. The main message states: "Última prueba: 28 jun. 2017 a las 14:02" and "La página no está optimizada para móviles". Below this, it says "Puede que sea difícil usar esta página en un dispositivo móvil" and provides a link to "OBTÉN MÁS INFORMACIÓN SOBRE EL DISEÑO PARA MÓVILES". A red icon of a smartphone with a diagonal slash through it is also present. The section "Soluciona los 4 problemas siguientes" lists the following issues:

- ✘ El contenido es más ancho que la pantalla
- ✘ El texto es demasiado pequeño para leerlo
- ✘ La ventana gráfica no se ha definido
- ✘ Los elementos en los que se puede hacer clic están demasiado cerca unos de otros

Ilustración 9 - Resultados del test de optimización para móviles de Google en la versión antigua del portal de la Escuela

Será muy importante cumplir con las especificaciones de Google para diseño web adaptado a móviles y, así, conseguir que el portal escale posiciones en el buscador respecto a otras páginas web del mismo ámbito que no cumplan con los estándares. Concretamente se observa en las indicaciones de Google que los problemas de la página radican en la falta de adaptación en el tamaño de los objetos a diferentes anchos de pantalla, con la subsecuente dificultad para leer e interactuar con sus contenidos.

Desde la misma página en la que se encuentra el test se puede acceder a la lista completa de criterios empleados por el mismo, que se corresponden con los que emplea el buscador de Google a la hora de valorar las páginas *mobile-friendly*:

- Evitar el uso de *Flash*: La mayor parte de navegadores para móviles no soportan contenido basado en Flash, por la información encontrada en dicho contenido no estaría disponible para los usuarios que navegan desde el *Smartphone*.
- Adaptación automática al tamaño de la pantalla: Los usuarios de móviles navegan desde pantallas de multitud de tamaños diferentes. Es recomendable emplear la etiqueta de HTML “meta viewport” para dar instrucciones al navegador sobre cómo actuar de forma automática ante esta variedad de resoluciones.
- Contenido de tamaño adaptable: Es un error común especificar el tamaño de los elementos de una página con un número concreto de píxeles. Esto provoca que dichos elementos no se adapten a diferentes tamaños de pantalla, desbordando el ancho disponible y obligando a hacer un *scroll* horizontal para visualizar todo el contenido. En su lugar es recomendable expresar los tamaños en porcentajes del tamaño total disponible.
- Tamaños de letra adaptables: Muchos desarrolladores especifican un tamaño de letra fijo para su contenido, de manera que en pantallas de diferentes tamaños el texto puede aparecer excesivamente grande o pequeño, obligando a hacer *zoom* para leerlo correctamente. Se recomienda emplear diferentes tamaños de letra en función del tamaño de la pantalla.
- Evitar enlaces demasiado cercanos entre sí: Cuando se navega desde una pantalla táctil, como la de un móvil o un *Smartphone*, la precisión a la hora de hacer *click* en un objeto de la pantalla es menor. Por esta razón es importante dejar un espacio entre los elementos susceptibles de ser activados, como los enlaces, para evitar que el usuario haga *click* por error en el objeto incorrecto.

Será necesario cumplir con todas estas recomendaciones para “aprobar” el test de Google y, con ello, aparecer en los primeros puestos de las búsquedas realizadas desde dispositivos móviles.

Por todo lo expuesto anteriormente, decidimos que sería necesario emplear alguna de estas técnicas de diseño web para móviles. Concretamente, y debido a la mayor sencillez a la hora de mantener una única página para todos los navegadores y dispositivos, decidimos emplear Responsive Web Design en la creación del nuevo portal. Se espera que tras el rediseño de todo el portal, este pase satisfactoriamente todas las pruebas y gane posiciones en los rankings de búsquedas en Google, atrayendo de esta manera más tráfico de visitantes.

1.3.4 - ESTADO DEL ARTE: PROGRAMACIÓN VERDE

1.3.4.1 – El problema medioambiental de la era de las comunicaciones y la necesidad de la computación verde

Los sistemas software en general, incluyendo los portales y las aplicaciones web, están aumentando continuamente en funcionalidad y complejidad, lo que trae la necesidad de hardware más potente tanto para desarrollar y mantener las piezas de software como para consumirlas. A su vez las empresas aplican las nuevas tecnologías a sus productos (internet de las cosas), los usuarios cada vez pasan más tiempo usando estos productos, y el número de objetos electrónicos en los hogares y oficinas crece a un ritmo incansable. El impacto de todos estos aparatos, sistemas y servicios en el medio ambiente se multiplica a velocidad alarmante, debido principalmente a la cantidad de desecho electrónico producido y al consumo de energía eléctrica por parte de los aparatos.

GLOBAL QUANTITY OF E-WASTE GENERATED			
Year	E-waste generated (Mt)	Population (billion)	E-waste generated (kg/inh.)
2010	33.8	6.8	5.0
2011	35.8	6.9	5.2
2012	37.8	6.9	5.4
2013	39.8	7.0	5.7
2014	41.8	7.1	5.9
2015	43.8	7.2	6.1
2016	45.7	7.3	6.3
2017	47.8	7.4	6.5
2018	49.8	7.4	6.7

Data 2015 onwards are forecasts

Tabla 4 - Cantidad de desecho tecnológico generada en el mundo. Los datos desde 2015 en adelante son predicciones. (Baldé, 2014)

En 2014, la cantidad de desecho electrónico generado en todo el planeta superó los 40 millones de toneladas, cerca de 6kg por habitante de la Tierra. En muchas ocasiones los componentes electrónicos desechados son altamente tóxicos y no existe una vía fácil y barata de contención. Los mecanismos de reciclaje y reutilización se ven desbordados por semejantes cantidades (solamente 6,5 millones de esas toneladas fueron recogidas por organismos de reciclaje oficiales), y estudios de las Naciones Unidas prevén que este problema siga empeorando en los años venideros (Baldé, 2014).

El consumo de energía eléctrica derivado de la fabricación y el funcionamiento de sistemas informáticos también dispara las alarmas cuando se analiza la progresión de los últimos años. Particularmente, los centros de procesamiento de datos están multiplicándose rápidamente desde la popularización de los servicios en la nube, con una previsión de crecimiento cercana al 200% hasta 2020 (CISCO, Cisco Global Cloud Index: Forecast and Methodology, 2015–2020, 2016). Las previsiones calculan que para esa fecha, el conjunto de centros de datos solamente en los Estados Unidos consumirá cerca de 40.000 millones de kWh al año (Shebabi, 2016).

Las grandes compañías de datos ya están tomando medidas para frenar el impacto de estas instalaciones, desde trasladarlas a áreas con climas fríos para facilitar su refrigeración -como hizo Google al construir su mayor centro de datos en Finlandia (Ando, 2013)-, hasta desarrollar sistemas de inteligencia artificial que administran automáticamente los centros de datos para maximizar su eficiencia.

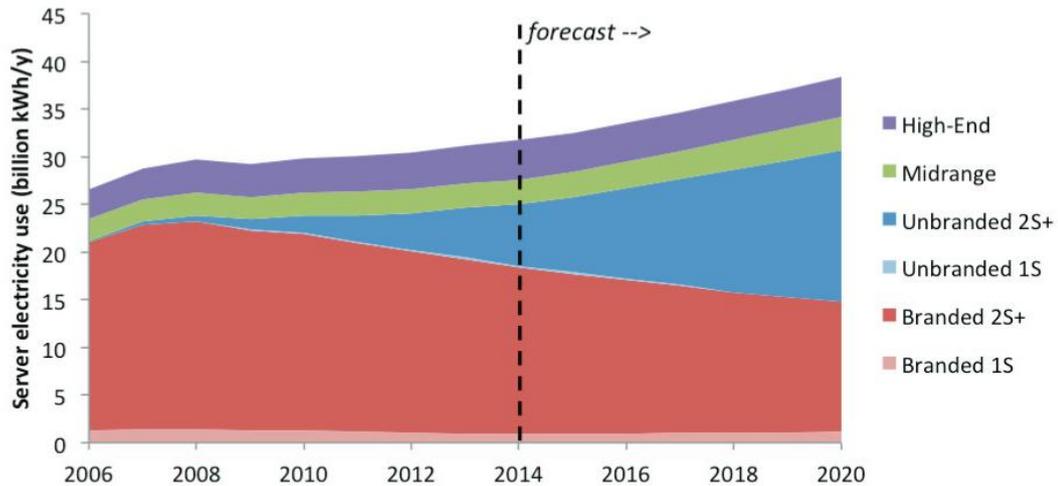


Ilustración 10 - Gasto energético total de servidores de internet en estados unidos (miles de millones de kWh/año). Los datos de 2014 en adelante son predicciones. Los colores indican la gama del servidor, siendo los de gama más alta morados, y la más baja rojos. (Shebabi, 2016)

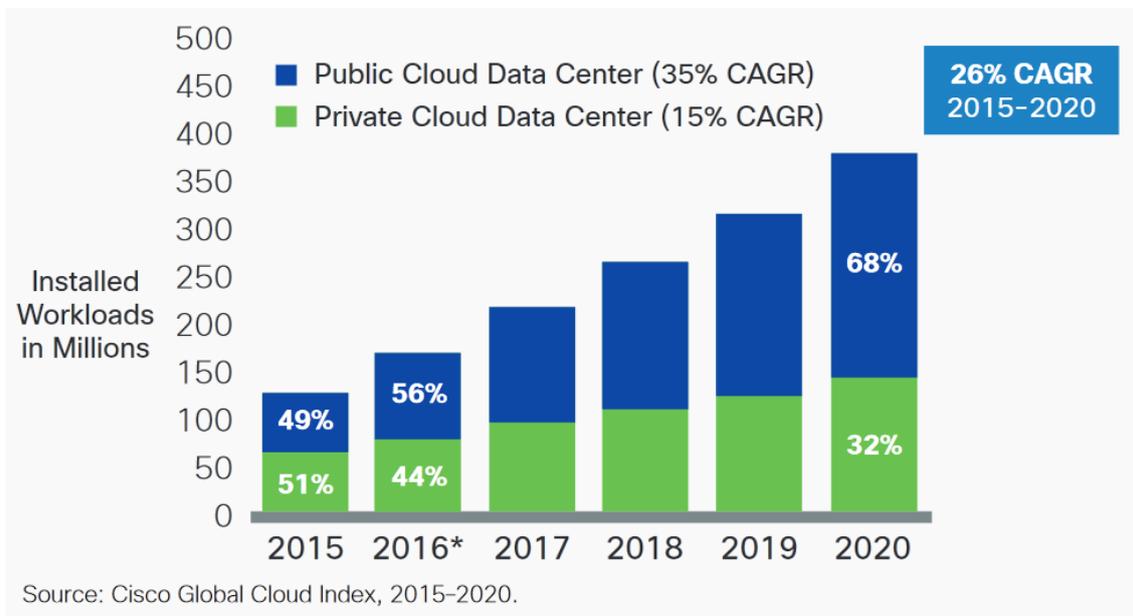


Ilustración 11 - Cantidad servidores en centros de datos instalados en el mundo, en millones por año. En azul servidores privados, en verde públicos. (CISCO, Cisco Global Cloud Index: Forecast and Methodology, 2015-2020, 2016)

El consumo de datos a través de internet también supone un fuerte impacto para el medio ambiente. La infraestructura necesaria para operar y mantener las redes de banda ancha que transportan los datos a todo el mundo también consume energía eléctrica. Además, el fuerte crecimiento del uso de ancho de banda en internet, agudizado por el boom en países en vías de desarrollo, está disparando las alarmas sobre la sostenibilidad a largo plazo (hablando del impacto sobre el medio ambiente) de estas infraestructuras.

Technology	Bandwidth (Mbps)	Watts per Circuit	Watts per Mbps
ADSL2+	24	6	0.25
VDSL2	60	9.7	0.16
GPON	75	12.25	0.16
Active Ethernet	1,000	10	0.01

Tabla 5 - Consumo eléctrico de diferentes redes de internet. (Vela, 2011)

El crecimiento explosivo del mercado de la telefonía móvil durante la última década ha acelerado el alza de estos dilemas: en el mundo se fabrican 2.000 millones de teléfonos móviles al año (ETBC, 2016), y los usuarios cada vez están más conectados a internet (según Cisco, en 2016 el consumo de datos en móviles creció un 63% solamente en 2016 (CISCO, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper, 2017)).

Del intento de escuchar y buscar una solución a todas estas preocupaciones surge la corriente de “computación verde”, o *green IT*. Cuando se habla de la computación “verde”, o energéticamente eficiente, se está haciendo referencia al conjunto de técnicas que buscan una reducción del impacto medioambiental producido por sistemas informáticos, no necesariamente obtenida mediante programación. Las técnicas pueden abarcar desde la modificación del software para mejorar la eficiencia de sus algoritmos, hasta la sustitución de hardware en un servidor por alternativas con menor gasto eléctrico, pasando por el diseño de centros de datos optimizados con este fin. Prácticamente cualquier elemento que interviene en un sistema informático es objeto de estudio en el ámbito de la computación verde.

Desde que comenzaron los esfuerzos de concienciación de la computación verde en el ámbito de los servidores y centros de datos, al inicio de la década, los datos actuales y las previsiones en cuanto al consumo energético de estas instalaciones han mejorado sustancialmente respecto a las proyecciones realizadas cerca del año 2010. Se ha logrado estabilizar la progresión exponencial que se preveía anteriormente, ahorrando más de 600.000 millones de kWh en una década (Shebabi, 2016).

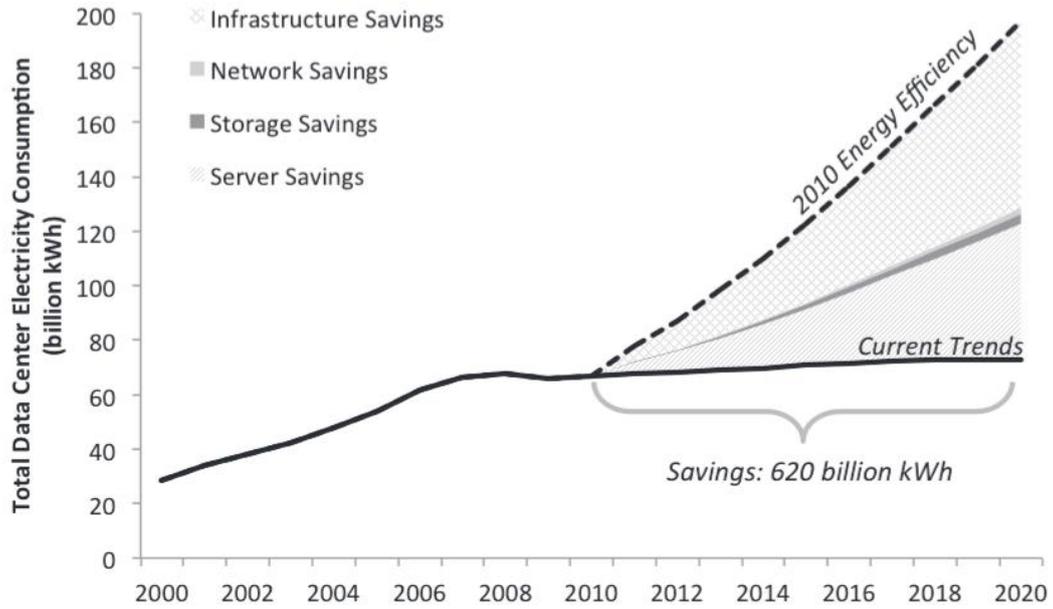


Ilustración 12 - Gasto energético total de servidores de internet en estados unidos (miles de millones de kWh/año). En línea de puntos, predicciones en el año 2010. Área en gris: ahorro total por medidas de optimización energética. (Shebabi, 2016)

1.3.4.2 – Computación verde en el desarrollo web

La programación verde aplicada en el campo del desarrollo web se encuentra con actores y escenarios que no aparecen en otras áreas, al ser la web un sistema distribuido en el que servidores y clientes interactúan entre sí.

Por una parte está el servidor web: Un potente ordenador que permanece encendido prácticamente 24 horas al día, 7 días a la semana, durante todo el año, supone un gasto energético considerable simplemente por tenerlo encendido. La revista de computación verde Vertatique publicó en 2015 situaba el consumo de los servidores de gama media en torno a los 350W de potencia (Vertatique, 2015). La carga de trabajo que soporta al servir una web a sus usuarios, compilar código, comunicarse con otras entidades del sistema como la base de datos... etc.; eleva el consumo en gran medida. Además de elementos directamente relacionados con la web que sirve, como los servidores HTTP, o la compilación de código del *back-end*, también son susceptibles de ser mejorados elementos menos obvios como el sistema operativo que corre en el servidor y los servicios en segundo plano que no son necesarios para el funcionamiento del ordenador. El hardware del que se compone el servidor también juega un papel importante, existiendo pérdidas importantes de potencia en ventiladores, tomas de alimentación, memorias y chips.

Reducir ciclos de trabajo de su procesador, memoria RAM empleada, o ancho de banda consumido, o sustituir hardware por piezas eléctricamente más eficientes, pasando por medidas para reducir la temperatura de su electrónica y por tanto disminuir la necesidad de refrigeración, son medidas que pueden crear un servidor web más “verde”.

Por otro lado está el cliente, el dispositivo que efectúa la petición de la página web, la carga y la muestra por la pantalla. El desarrollador web tiene la capacidad de influir en el consumo energético del cliente por varias vías: Modificar el código para disminuir el número de ciclos de procesador y memoria RAM que emplea una aplicación web; o reducir la cantidad de datos descargados que se requieren para visualizar la página web, son las vías a través de las que se puede conseguir un mayor impacto energéticamente.

Con la popularización de ordenadores portátiles, smartphones y tabletas, cobra enorme importancia la optimización de la página web en el lado del cliente con el fin de ahorrar batería en el dispositivo, un factor que apenas existía en décadas anteriores, cuando el usuario de internet navegaba eminentemente desde su ordenador de sobremesa.

La evolución de las tendencias en el desarrollo web también ha traído consigo un crecimiento sustancial en el “peso” de las páginas: los documentos cada vez son más grandes, se emplean más librerías Javascript y CSS, las páginas web se sirven de más imágenes y videos para atrapar la atención del usuario...

Durante los últimos cinco años, el peso medio de las 1000 páginas web más visitadas en el mundo creció más de un 100% (HTTPArchive, 2017). Crear un portal web moderno, que se adapte a las tendencias actuales, sin que a la vez aumente de manera notable el tamaño de las descargas supone un reto para el desarrollador

Total Transfer Size & Total Requests

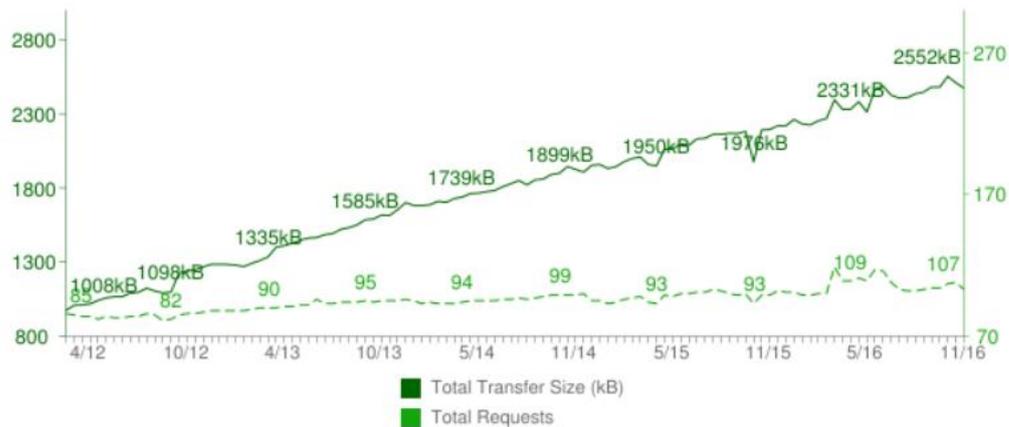


Ilustración 13 - Media del total de datos transferidos y peticiones HTTP en las 10.000 páginas de internet más visitadas del mundo. (HTTPArchive, 2017)

Uno de los objetivos de este Trabajo de Fin de Grado es que el nuevo portal de la Escuela implemente medidas “verdes” para reducir el gasto energético en el servidor, el consumo de datos a través de internet, tanto en servidor como en cliente, y el gasto de batería en dispositivos móviles clientes. La diversidad de aspectos en los que se puede trabajar a la hora de optimizar el portal web de la Escuela suponía una cantidad de trabajo que rebasaría por mucho el objetivo de un Trabajo de Fin de Grado, por lo que se vuelve necesario acotar el foco del Trabajo para no perder perspectiva y diluir los esfuerzos en demasiados frentes. Al comenzar a trabajar en el TFG, establecimos que los esfuerzos se centrarían principalmente en el *front-end* del portal, combinando el trabajo de remodelación con el de optimización energética.

El ámbito de *front-end* se conforma por todos los aspectos del desarrollo web que entran en contacto con el usuario, en contraposición del *back-end*, que engloba los aspectos que quedan ocultos al mismo. Los documentos HTML, CSS y Javascript llegan al navegador a través de internet, y este se encarga de crear la visualización de la página web que aparece en la pantalla, y son los componentes más importantes del *front-end*, por lo que serán el área de trabajo principal sobre el que se centrarán los esfuerzos.

Existen infinidad de áreas y conceptos en el ecosistema de un portal web sobre los que se podría trabajar para incrementar la eficiencia energética del mismo, pero debido a la acotación mencionada, no serán objeto de trabajo. Se tratará muy brevemente el aspecto *back-end*, y concretamente la arquitectura del servidor y su hardware caen totalmente fuera del estudio del Trabajo.

Cabe resaltar que la remodelación de un portal web supone un abanico de cambios muy amplio, especialmente cuando se trata de la modernización de una web que ha recibido tan pocas modificaciones durante los últimos años. Modernizar un portal y adaptarlo a las corrientes de diseño web actuales, mejorando y añadiendo características, incurre en un

código más complejo operativamente hablando. La inclusión de *frameworks* y librerías que agregan funcionalidad, y el diseño de estilos acordes a las corrientes actuales, con uso exhaustivo de contenido multimedia, pueden aumentar el ancho de banda consumido. Es por ello que el objetivo fundamental del aspecto “verde” de la remodelación no debe ser mejorar los datos del portal anterior, sino conseguir que el nuevo portal web (visto de manera independiente al anterior), implemente técnicas energéticamente eficientes que mejoren sustancialmente el consumo de un hipotético nuevo portal que hubiera sido desarrollado sin tener en cuenta la programación “verde”.

2: TRABAJO DESARROLLADO

La fase de remodelación se dividió en dos bloques principales: el rediseño visual y funcional por una parte, y la optimización energética por la otra. Los bloques no son independientes entre sí, dado que no tendría sentido optimizar energéticamente el portal antiguo y rediseñarlo posteriormente ya que muchas de las mejoras de eficiencia se perderían en el proceso. Por esta razón los primeros esfuerzos se concentraron en el bloque de rediseño visual y funcional, hasta obtener un portal web prácticamente definitivo y listo para abrir al público. Posteriormente se trabajará en la optimización energética del nuevo portal, a la vez que se realizarían actualizaciones menores en el diseño, en función de la realimentación que se obtenga tras publicar el nuevo diseño de la página.

Finalizado el análisis del problema, y tras determinar cuáles serían los focos de trabajo en el capítulo anterior, realicé una serie de remodelaciones en todo el aspecto *front-end* del portal. Las modificaciones fueron consensuadas con la Junta Directiva de la Escuela, actuando como intermediario, Carlos Alonso Gómez, secretario académico y miembro de dicha directiva.

Carlos Alonso actuó, además de cómo enlace con la Junta Directiva, como orientador, ya que en calidad de Analista de Red de la Escuela lleva varios años siendo el principal administrador del portal web y el servidor, y conoce de forma rigurosa su estructura y funcionamiento.

Los primeros pasos de esta etapa consistieron en crear una copia del servidor actual para disponer de un entorno de pruebas. El repositorio de código fuente del portal de la Escuela se viene administrando mediante *Subversion* (SVN), un Sistema de Control de Versiones de código abierto desarrollado por Apache. Mediante *Subversion*, los cambios llevados a cabo en el entorno de pruebas serán llevados al entorno de producción una vez esté en un estado satisfactorio, momento en el cual el nuevo portal estará accesible al público. *Subversion* también permitirá actualizar el entorno de pruebas con los cambios que se realicen en producción en tiempo real.

Para crear el servidor del entorno de pruebas, se replicó la arquitectura del servidor de producción en la medida de lo posible: Servidor HTTP Apache Tomcat, funcionando sobre Linux. Replicar la Base de Datos no sería necesario ya que todas las operaciones necesarias para las pruebas serían en modo lectura, por lo que se conectó el entorno de pruebas con la base de datos de producción.

Establecido el entorno de pruebas, comenzó la fase de rediseño visual y funcional.

2.1 - REDISEÑO VISUAL Y FUNCIONAL

2.1.1 – ANÁLISIS CUALITATIVO DE PORTALES WEB DE ÁMBITO SIMILAR

Con el fin de obtener un entendimiento de las tendencias en el ámbito del diseño web para portales similares al del presente Trabajo, se analizaron cualitativamente los sitios web de otras escuelas de Telecomunicación y carreras afines en España.

El proceso consistió en identificar las páginas web más relevantes, (en referencia a la similitud en cuanto a las carreras impartidas y la modernidad del portal) buscar patrones comunes en su diseño y estructura, y analizar dichos patrones para determinar si son aplicables al portal de la escuela.

Es necesario apuntar que, mientras que el CMS del portal de la ETSIT gestiona únicamente contenidos, la mayoría de CMS utilizados por las páginas visitadas gestionan además estilos CSS. Esto hace que algunas webs hayan sido descartadas por ser muy similares estéticamente y funcionalmente entre sí al usar el mismo CMS, o uno muy parecido.

Las páginas web analizadas finalmente fueron:

- ETSETB – Universitat Politècnica de Catalunya (<https://etsetb.upc.edu/ca>)
- ETSIT – Universidad de Granada (<http://etsit.ugr.es/>)
- ETSIT – Universidad Politécnica de Madrid (<http://www.etsit.upm.es/>)

Por ser tres escuelas técnicas en las que se imparten carreras de ingenierías de Telecomunicación.

- Escuela de Ingeniería Informática – Universidad de Valladolid (<https://www.inf.uva.es/>)

Por ser una Escuela de Ingeniería de un perfil muy similar al de la ETSIT y pertenecer, además, a la misma Universidad.

- Físicas – Universidad Complutense de Madrid (<http://fisicas.ucm.es/>)

Con el objetivo de incluir en el análisis, al menos, una página con un estilo menos moderno y más similar al del portal antiguo de la ETSIT.

A continuación se muestran capturas de las páginas principales de los portales mencionados. En las capturas de las páginas web se ha identificado, mediante un código de colores, las secciones en las que se dividen las mismas. Se tratará de ver que secciones son compartidas entre estas páginas y si, en general, las secciones están estructuradas de manera similar.



Ilustración 14 - Izquierda: ETSETB Barcelona. Derecha: ETSIT Granada

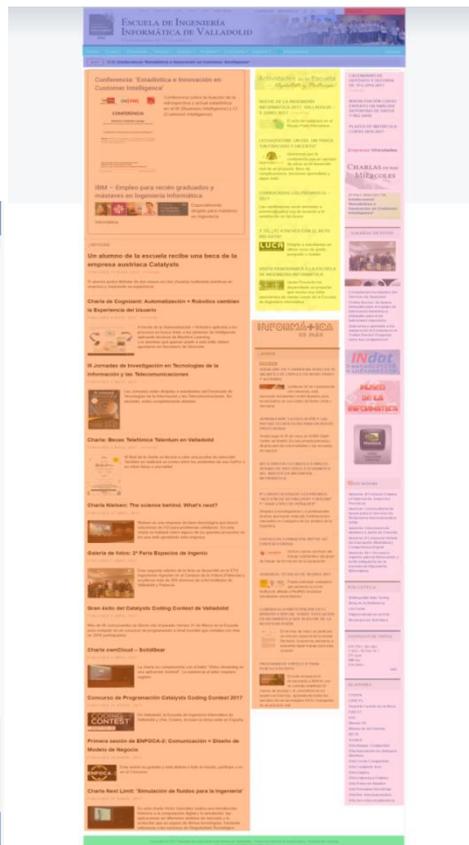


Ilustración 15 - Izquierda: ETSIT UPM. Derecha: Informática UVA

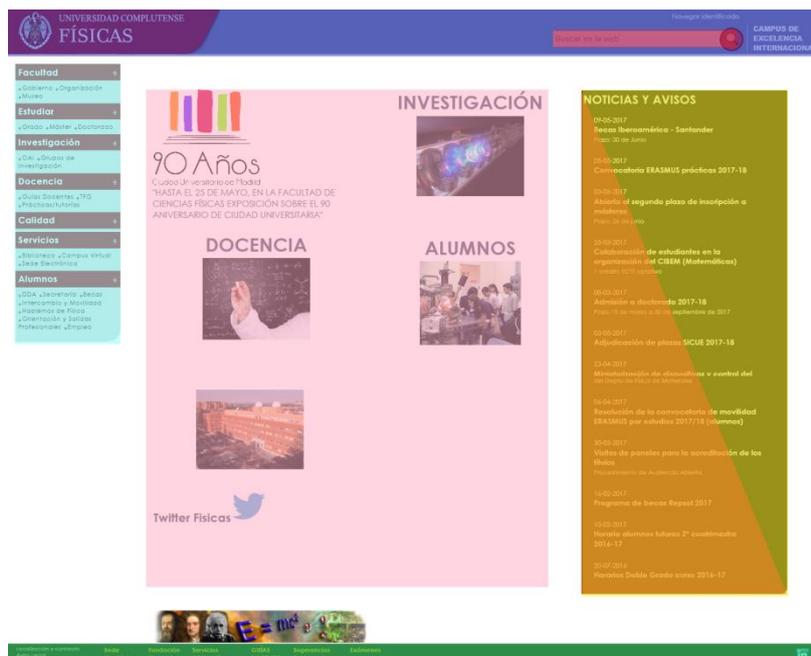


Ilustración 16 - Físicas UCM

A continuación se detallan las secciones más significativas, por aparecer con frecuencia en las páginas analizadas. Se analizará su estructura, colocación e importancia, y se identifican en las capturas de las páginas con un código de colores mencionado en cada sección.

- Cabecera o *header* (azul oscuro): En la parte superior de todas las páginas existe una sección de encabezado con funciones similares. Se muestra el nombre y el logo de la escuela a la que pertenece la página web, y en la mayoría existe un cuadro de búsqueda y una barra de navegación. Además, en los portales en los que aparece la opción de identificarse como miembro de la escuela, la opción siempre está presente en la sección de cabecera.

Al ser lo primero con lo que se encuentra alguien al visitar una página web, es convención colocar la navegación principal en esta sección facilitando al visitante encontrar los enlaces a las secciones deseadas sin tener que buscar por toda la página. Es especialmente importante en portales grandes con muchas páginas y secciones.

- Búsqueda (rojo): El cuadro de búsqueda interna aparece en todas las páginas analizadas, siempre dentro de la cabecera. Ninguna de las páginas implementa su propio motor de búsqueda, sino que hacen uso de servicios gratuitos como el buscador *Google*.

El nuevo portal de la Escuela hará también uso del motor de búsqueda de *Google*, que permite personalizarlo para realizar búsquedas internas.

- Barras de navegación o *navbar* (azul claro): aunque por norma general se observan las barras de navegación en la cabecera y en disposición horizontal, existen variaciones: la ETSIT de la Universidad de Granada dispone también de una barra vertical más completa en un lateral de la página, y la página de la Facultad de Físicas de la Universidad Complutense de Madrid únicamente tiene una barra lateral vertical.

La barra vertical lateral permite mostrar más cantidad de información a cambio de espacio, mientras que la barra horizontal superior es un método más compacto de ofrecer unos pocos enlaces. La solución en algunas páginas es emplear categorías desplegadas que solo aparecen cuando se mantiene el puntero sobre dicha categoría, consiguiendo más espacio para enlaces.

- Pie de página o *footer* (verde oscuro): En el borde inferior de cada página existe una sección de pie de página en la que se incluye información legal y enlaces a páginas de ayuda y contacto. También suele aparecer la dirección de la escuela y, de existir, las opciones de accesibilidad. En un caso se colocan enlaces de navegación secundaria en esta sección, lo que ayuda a liberar espacio en otras secciones de la página.

Es especialmente importante la colocación de la información de contacto en esta sección ya que, por convención dentro del diseño web, un gran número de sitios web colocan aquí esta información. El usuario acostumbrado a navegar por internet sabe que, para encontrar la información de contacto, puede viajar hasta el extremo inferior de la página. Esto se vio en el capítulo sobre 1.3.2 – Presentación de la información, cuando se hablaba de los hábitos de navegación de los usuarios de internet. Por esta razón consideramos el *footer* como el sitio ideal para colocar enlaces a los perfiles en las redes sociales de la Escuela al ser estos una forma de contacto diferente.

- Galería de imágenes o *carrusel de imágenes* (verde claro): Las páginas de la ETSIT de la UPM y de la ETSETB de Cataluña presentan una galería rotatoria en la que se muestran imágenes con enlaces a diferentes secciones relevantes. La imagen mostrada en la galería cambia cada varios segundos, y cada imagen enlaza con una sección o noticia. La galería suele estar situada en el centro de la página y ocupar una buena parte de la misma para mostrar imágenes de gran tamaño.

Las imágenes de gran tamaño son una gran herramienta para atraer la atención del público, y por tanto es costumbre colocar en esta sección enlaces a los que se quiera dar visibilidad, como eventos especiales, noticias de gran importancia...

Cabe destacar que al emplear archivos de gran tamaño, el carrusel de imágenes suele aumentar considerablemente el *peso* de una página web. El código Javascript necesario para rotar continuamente las imágenes causa, además, un consumo de energía muy elevado comparado con el de los elementos estáticos de las páginas.

- Tablón de anuncios, cartelera, eventos, agenda, calendario (amarillo): Es común que aparezca una sección en la que se enuncien eventos próximos relacionados con la escuela y su actividad, ordenados por fecha, y a veces acompañado de un calendario.

Suele ser una de las partes más dinámicas de la página web en cuanto a la frecuencia con la que cambia su contenido y por ello atrae la atención de los visitantes habituales. Ya que su función es anunciar eventos próximos, conviene colocar esta sección en una zona visible de la página e identificarla claramente. También es importante que el usuario pueda conocer, de un rápido vistazo, las próximas citas anunciadas.

- Noticias (naranja): De forma similar a la función realizada por el tablón de anuncios para eventos futuros, se incluyen también secciones en las que se anuncian eventos pasados y

noticias relacionadas con el centro, la actividad de sus estudiantes, el ámbito académico o el mundo de las telecomunicaciones. Las secciones de noticias y de tablón de anuncios suelen cumplir tareas similares y, en ocasiones, comparten área.

- Enlaces destacados (rosa): Todos los portales analizados dedican áreas para enlaces a secciones internas, que por su función acumulan grandes cantidades de tráfico o enlazan a páginas de gran importancia. Estas áreas de enlaces destacados son muy dispares entre los portales analizados en todos los aspectos: posición dentro de la página, estructura, número de secciones dedicadas a estos enlaces, cantidad de detalle ofrecido sobre el destino del enlace...

Estas secciones se suelen emplear para facilitar el acceso a zonas de un portal web que no aparecen en las barras de navegación por no ser secciones principales, pero que atraen muchas visitas por cumplir funciones de gran interés. Su objetivo es mejorar la usabilidad de la página haciendo que el usuario encuentre estas áreas rápidamente.

- Redes sociales (morado): A pesar de que, en general, las universidades españolas aún no han diseñado estrategias claras para el uso de las redes sociales (Brito, 2016), muchas están comenzando a adquirir presencia on-line, creando sus perfiles en las redes y estableciendo una comunicación con sus alumnos y sus potenciales alumnos. Se pueden ver secciones en algunas de las páginas web analizadas, dedicadas exclusivamente a enlazar a los perfiles de su escuela en las redes, si bien otras páginas no tienen enlaces, o se les da poca importancia colocándolos en áreas compartidas con todo tipo de utilidades.

En cuanto a la estructura de las páginas, se puede apreciar como la mayoría de estas dispone su contenido en varias columnas, generalmente tres. Una de las columnas ocupa mayor ancho que las demás, y en ella se suelen encontrar las secciones de noticias y enlaces destacados. La otra disposición observada consiste en la colocación de bloques horizontales, separados por espacios en blanco. Estos bloques agrupan contenidos similares entre sí y, a su vez, pueden estar divididos en columnas. Esta segunda estructura se aprecia principalmente en la página web de la ETSIT-UPM.

2.1.2 – ESPECIFICACIONES PREVIAS, REQUISITOS

En esta etapa no se parte de total libertad creativa para remodelar el portal, al existir ciertas especificaciones establecidas de antemano por los miembros de la Junta. Las especificaciones tienen como objetivos mantener una continuidad respecto al anterior portal, añadir funcionalidades específicas deseadas para el nuevo portal, y dotar al mismo de “imagen de marca”, identificándolo con la Escuela y la Universidad de Valladolid.

En sucesivas reuniones de la Junta de Escuela se concretaron los puntos fundamentales que el portal debería satisfacer. Fueron necesarias varias iteraciones para llegar a un acuerdo en cuanto a la complejidad de algunos puntos y, finalmente, se establecieron los siguientes compromisos:

- Mantener las barras de navegación superior y lateral. Los enlaces que figuran en dichas barras no están escritos en el código fuente de portal, sino que se obtienen de la base de datos al compilar la página. De esta manera, al añadir o eliminar categorías en el árbol de navegación del portal a través de la base de datos, las barras de navegación se actualizarán automáticamente.

Esta estructura elimina la necesidad de modificar el código perteneciente a los elementos de navegación cuando se altere la estructura de páginas del portal.

- Mantener una vista previa del tablón de anuncios en la página principal. De igual manera que las barras de navegación, el tablón de anuncios se genera automáticamente al compilar la página principal del portal, obteniendo los anuncios y avisos de la base de datos. Esto permite añadir, modificar o eliminar anuncios sin tener conocimientos de programación web.

La estructura del tablón de anuncios debe estar compuesta por tres secciones: Avisos destacados, Conferencias y Talleres, y Empleo y Prácticas. Dentro de cada sección se mostrarán el titular de hasta cuatro elementos, con un enlace a la correspondiente sección de la página del tablón de anuncios.

- Carrusel de fotografías: En la página principal debe existir una sección con fotografías que enlacen a diferentes secciones destacadas del portal, pensada para enlaces de interés temporal. Solo una imagen será visible en un momento determinado y esta ocupará el 100% del espacio del carrusel. La fotografía mostrada rotará con el tiempo. También habrá botones para avanzar y retroceder entre las diferentes imágenes.
- Sección de enlaces preferentes. Si bien no se especifica detalladamente en cuanto a apariencia ni funcionalidad, se pide que exista en la página principal una sección de enlaces destacados. Será de utilidad similar al carrusel, pero ideada para enlaces más duraderos como secciones más visitadas por los usuarios.
- Colores de marca: Los colores principales del portal pasarán de ser azul, blanco y gris, a azul y naranja. El color azul es el empleado tradicionalmente como color principal en el portal de la Escuela durante los años, y será el tono principal de la página. El color naranja

es el que tradicionalmente se ha utilizado para encuadernar los Proyectos de Fin de Carrera y Trabajos Fin de Grado, y aparecerá como color secundario. Además, estos colores son complementarios entre sí, formando según la teoría del color, el tipo de armonía más básico entre dos colores.

El rediseño del portal web coincide temporalmente con el lanzamiento, por parte de la escuela, de un concurso público para que alumnos y trabajadores presenten sus propuestas para un nuevo logotipo corporativo oficial para la E.T.S.I.T, conmemorando los 25 años de vida de la Escuela.

El nuevo logotipo aparecerá en la página web sustituyendo al antiguo, y contará con la misma paleta de colores: azul y naranja.



Ilustración 17 - Nuevo logo de la Escuela

- Enlaces al portal web de la Universidad de Valladolid, a la Biblioteca de la Uva (el cual fue retirado posteriormente), y a la versión en inglés del portal de la Escuela, visibles en la página principal de la web.

2.1.3 – REDISEÑO DEL PORTAL

2.1.3.1 – Metodología de trabajo

Tras la fase de planificación, comenzamos a trabajar en la escritura del código fuente del nuevo portal. Desde el comienzo se pretendió poner en práctica elementos de los paradigmas de *programación ágil*, que pretende establecer unas guías de conducta para el desarrollo de productos de software de forma iterativa e incremental, en el que las especificaciones, los objetivos y las soluciones varían y evolucionan con el tiempo (Beck, 2001).

- Reuniones “cara a cara” y frecuentes con un representante del cliente (en este caso el representante es Carlos Alonso que, como ya se ha explicado anteriormente, ejerce de enlace con la junta directiva del centro, a la cual pertenece en calidad de Secretario).

En estas reuniones se da lugar a un intercambio de información y pareceres, en el que el representante me transmite las preguntas, opiniones y sugerencias del cuerpo directivo. Asimismo, yo presento al representante las evoluciones en el portal desde la última reunión, para que la Junta pueda evaluarlas y devolver a su vez sus impresiones.

- Iteraciones frecuentes, en pequeños ciclos de trabajo, de manera que el resultado de la retroalimentación fruto de las reuniones sea visible en un plazo corto de tiempo. Esto permite que la evolución del proyecto sea más ágil y dinámica, y se malgaste menos tiempo en características que están destinadas a ser descartadas o modificadas.

En contraposición al desarrollo en cascada típico, en el desarrollo iterativo característico de *Agile* pretende ofrecer *software* funcional en cada una de las cortas iteraciones. Esto se traduce, en cuanto a impacto en el presente proyecto, en que cada una de las características añadidas es implementada y testeada individualmente, en lugar de en grandes bloques. Este proceso redundante en ciclos más cortos en los que la retroalimentación llega mucho antes, lo que a su vez proporciona una gran flexibilidad a la hora de realizar modificaciones en respuesta a las reacciones del cliente.

- El trabajo realizado se encuentra siempre, desde la primera iteración, disponible para el cliente. Para esto, se habilitó en los servidores de la Escuela un subdominio de desarrollo: *desa.tel.uva.es*. Después de realizar cambios en el entorno de desarrollo explicado anteriormente, los cambios se enviaban al repositorio de control de versiones vinculado al subdominio de desarrollo. De esta manera, la junta podía visualizar los cambios en cualquier momento visitando la dirección *desa.tel.uva.es*, sin necesidad de disponer de ningún software especial. Este concepto en metodología ágil se denomina *irradiador de información*.

Para que se iniciara el intercambio de pareceres lo antes posible, centré mis primeros esfuerzos en desarrollar una versión usable del nuevo portal en poco tiempo. Esto supuso realizar un rediseño inicial completo a la página principal, en cuanto a apariencia y estructura, y un lavado de cara menor a las páginas secundarias, con la intención de que se pudiera apreciar la dirección del rediseño en el futuro sin consumir excesivo tiempo durante esa fase.

Tras las primeras reuniones con el representante de la Junta de Escuela, habiendo conseguido un resultado satisfactorio en la página principal, retomé el trabajo en el resto de páginas y en

el desarrollo de características de menor importancia, además de continuar refinando la página principal en base a la realimentación obtenida en las reuniones.

La tercera etapa en la que se dividieron los esfuerzos fue en la creación de la versión móvil de la página, una vez alcanzada la iteración en la que el rediseño general de todas las páginas obtuvo una respuesta positiva por parte del cliente.

El objetivo en esta fase fue escribir el código necesario para que la navegación desde dispositivos móviles y tabletas fuera lo más natural posible, a la vez que se mantuviera constante el estilo de las páginas. Durante los pasos previos se tomaron varias decisiones, tanto de diseño como de programación, pensando en facilitar la integración de la versión móvil. En el caso ideal, estos esfuerzos minimizarían la necesidad de trabajar sobre el código HTML y Javascript de la página, dejando la mayor parte del trabajo para el código CSS. Durante los capítulos siguientes se explicarán estas decisiones mientras se presenta el trabajo realizado.

La metodología de trabajo expuesta se mantuvo constante durante todo el ciclo de rediseño. El objetivo final se fijó en la creación del portal web planeado, adaptado para móviles y listo para ser lanzado al servidor de producción (es decir, listo para presentar al público). Sólo tras el lanzamiento comenzarán las labores de optimización y programación verde, de manera que ésta primera versión no será todo lo eficiente que pudiera ser.

2.1.3.2 – Creación de la página principal

La primera iteración de la página principal (la visible en www.tel.uva.es) consistió en una versión simplificada del prototipo planeado, con la mayor cantidad posible de características implementadas en una forma básica, de forma que el cliente tuviera con prontitud una idea del resultado final. Para ello, basándonos en el análisis de otras páginas web similares y en las especificaciones previas, ambas explicadas anteriormente, decidimos los elementos y la estructura que tendría la página.

En cuanto a la disposición de los elementos dentro de la página, se decidió optar por la estructura de bloques horizontales observada en la página de la ETSIT-UPM. Esta estructura presenta la ventaja, frente a la de columnas verticales, de ser más modular, permitiendo alterar la composición y añadir, modificar o eliminar elementos produciendo el menor efecto posible en el resto de elementos.

Esta modularidad también será ventajosa a la hora de realizar la versión para móviles y tabletas, ya que programáticamente será más sencillo y práctico recolocar los elementos de los bloques horizontales de manera que sean legibles en pantallas pequeñas, frente a la complejidad de recolocar los elementos dispuestos en largas columnas.

Otra decisión sobre la estructura tomada a priori pensada con la versión móvil en mente fue sobre el ancho ocupado por el contenido frente al ancho disponible en la pantalla. Hasta ahora, la página web de la Escuela ha empleado una estructura de “pantalla completa”, en la que el contenido se expandía para ocupar el 100% de la pantalla independientemente de su resolución. Con la explosión del diseño de páginas web para móviles, se ha generalizado el diseño de “ancho fijo”, en el que el contenido se expande hasta ocupar un ancho en píxeles fijado por el diseñador. Si la pantalla tuviera una resolución menor de ese ancho fijo, el contenido se re-escalaría hasta ocupar otro ancho fijo, menor, fijado de antemano.

La ventaja de este diseño de “ancho fijo” reside en que el diseñador fijará varios “escalones”, de manera que solo tendrá que preocuparse del aspecto de la página web para esas resoluciones conocidas de antemano. La alternativa de “pantalla completa” aprovecha un mayor espacio de la pantalla, pero presenta el inconveniente de no saber cuál será el ancho de pantalla en el que se visualizará el contenido: la página se ensanchará o estrechará hasta ocupar un ancho diferente en función de la pantalla. Esto hace necesario un diseño que se adapte a todas las pantallas en lugar de a unos pocos tamaños conocidos.

Durante los años en los que los monitores de ordenador eran de tamaños reducidos y no existían los smartphones ni las tabletas, este era un problema menor, pero a medida que los monitores han aumentado su tamaño y su resolución, y los smartphones y tabletas han aparecido con pequeñas pantallas, el rango de resoluciones a “vigilar” ha crecido fuera de control. Por las razones anteriores se decidió abandonar el aspecto de “pantalla completa”, para emplear anchos fijos.

Tomadas las decisiones anteriores, y decididos los elementos que aparecerían en la página principal, se realizó un esquema con la estructura prevista, y posteriormente se escribió el código HTML, CSS y Javascript para plasmar el diseño.

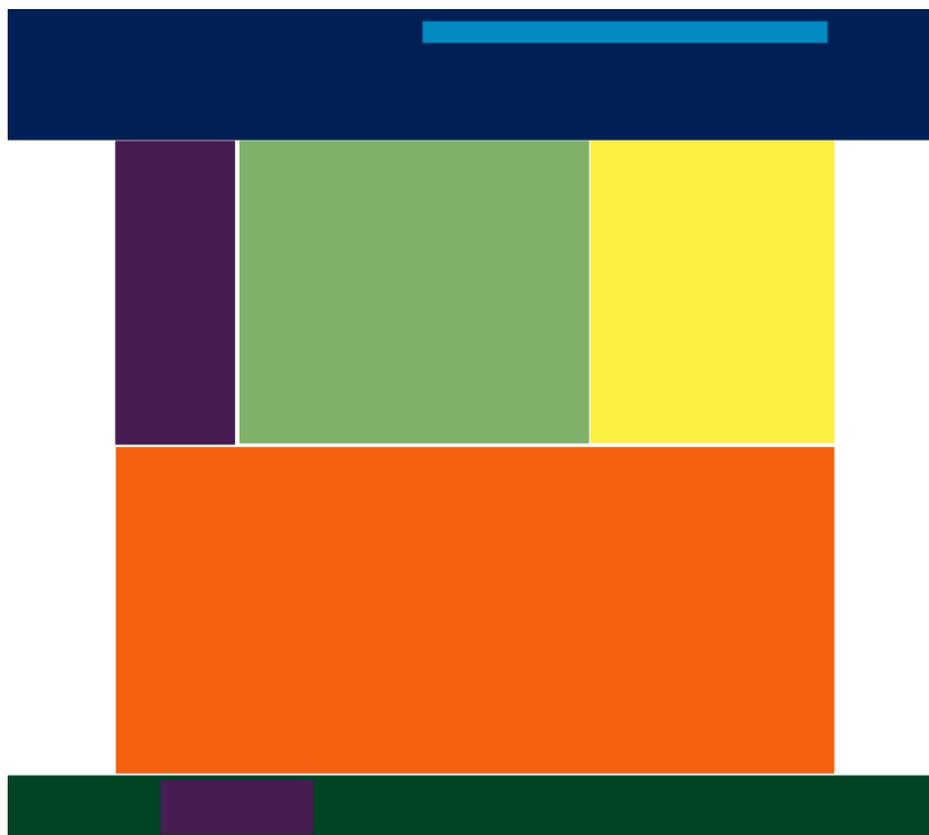


Ilustración 18 - Esquema de los elementos y estructura en la página principal del nuevo portal. Código de colores explicado en el capítulo 2.1.1.

En el esquema se indican, con el código de colores del capítulo 2.1.1 – Análisis de portales web de ámbito similar, las diferentes secciones de la página principal y como se colocarán dentro de la misma. A continuación se ofrece una explicación de cada una de las secciones:

- Cabecera o *header*: Ocupando todo ancho de la parte superior de la página, contendrá la barra de navegación, acceso al área de login, y un enlace al portal de la Universidad de Valladolid.

En la página principal del portal anterior existía, además de la barra de navegación en la cabecera, una barra lateral con enlaces a zonas de interés que no quedan directamente enlazadas por la barra superior. Con el fin de mantener el acceso a estas zonas sin requerir una segunda barra de navegación en un lateral, se diseñó una barra de navegación de dos niveles, de manera que al pasar el puntero del ratón por encima de un enlace de primer nivel, se hagan visibles enlaces a las secciones de segundo nivel vinculadas.

Por petición de la Junta, se incluyó en la cabecera, bajo la barra de navegación, un banner con el nuevo logo y marca de la ETSIT, creado por el ganador del concurso con motivo del 25 aniversario de la Escuela.

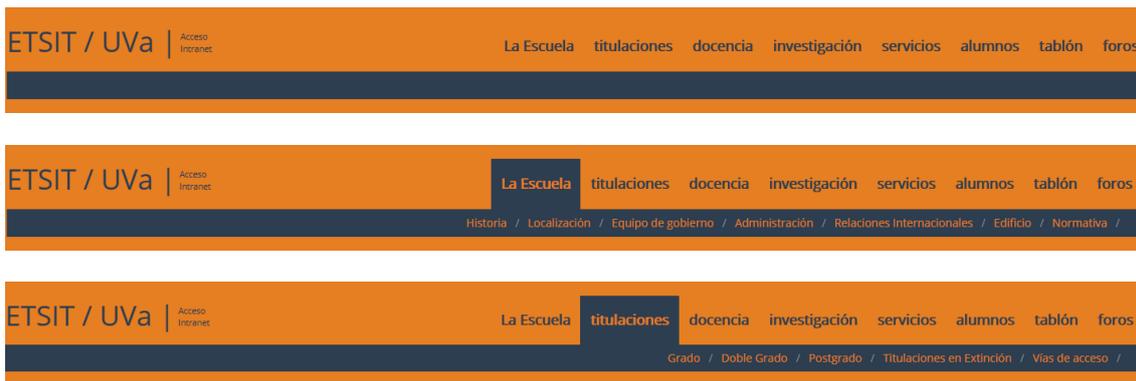


Ilustración 19 - Barra de navegación en diferentes estados. Superior: en espera. Central e inferior: desplegadas secciones "La Escuela", y "titulaciones" respectivamente.



Ilustración 20 - Banner de la cabecera de la página principal. Logo de la Escuela, Nombre de la Escuela y la Universidad, y logo conmemorativo del 25º aniversario de la Escuela.

Para añadir elementos a la barra de navegación no será necesario escribir sobre el código fuente de la página, ya que el contenido de dicha barra se genera en el momento de la compilación de las plantillas de JSP, como ya se explicó anteriormente. En su lugar, los elementos se modificarán en la base de datos de la página web. El personal de la Escuela tiene acceso a un gestor de contenidos tras identificarse, a través del que pueden realizar cambios en ciertas páginas, añadir contenidos y, si dispone de los permisos necesarios, modificar esa estructura de navegación. Dicho gestor de contenidos no forma parte del presente proyecto, sino que fue desarrollado con anterioridad y lleva en funcionamiento desde entonces.

A continuación de la cabecera, un primer bloque horizontal, dedicado a las secciones de mayor importancia. Este bloque ocupará aproximadamente todo el alto desde el fin de la cabecera hasta el extremo superior de la pantalla, de forma que no sea necesario hacer *scroll* hacia abajo para visualizar todo su contenido. El bloque contiene tres secciones que se reparten todo el ancho disponible:

- Carrusel de imágenes o *slideshow*: Se situará en el centro de la página y siendo el elemento de mayor tamaño, atrayendo por tanto la atención del visitante. Se empleará este espacio para enlazar a áreas del portal de interés global, tanto para personas relacionadas con la Escuela (personal, estudiantes), como visitantes externos.

Como se especificó anteriormente, el carrusel consistirá en una imagen con pie de foto que enlaza a una página de interés. Existirán múltiples imágenes, que rotarán automáticamente cada cierto número de segundos, llevando la atención del visitante a los distintos enlaces. La sucesión de imágenes se detiene mientras el cursor del ratón está situado sobre el carrusel, y mediante dos botones se podrá navegar a voluntad entre todas las imágenes que contiene.

Para añadir, modificar y eliminar contenido del carrusel, será necesario escribir sobre el código de la página. Sin embargo, se ha escrito el código de manera que este proceso sea lo más sencillo posible para que no sean necesarios más que unos básicos conocimientos de *HTML*.

El formato de cada imagen del carrusel será el siguiente:

```
<li class='noselected'><a href=""><img src='' border='0' /></a></li>
```

Código 1 – Código HTML de un elemento vacío del carrusel de imágenes.

Tan solo será necesario escribir la dirección del enlace entre las comillas que aparecen tras la propiedad “href” en el elemento *anchor*, y la ruta de la imagen entre las comillas tras la propiedad “src” del elemento *image*, y añadir la línea resultante a la lista que se encuentra dentro del código HTML. El código CSS y Javascript existente se encargará de manejar todas las imágenes añadidas sin necesidad de más intervenciones.



Ilustración 21 - Vista del carrusel de imágenes. Se aprecian los botones para avanzar y retroceder, y el rótulo explicativo de la imagen.

- Tablón de anuncios y avisos: Situado a la derecha del carrusel, ocupando la mayor parte del ancho restante. Se pretende que, al estar situado directamente junto al carrusel, reciba atención por parte del visitante. Contiene los enlaces generados automáticamente al compilar la página principal, de manera que su actualización no requiere escribir sobre el código de la página. El mecanismo es similar al de la barra de navegación, y el personal de la Escuela puede añadir nuevos eventos y avisos a través del gestor de contenidos.



Ilustración 22 - Vista previa del tablón de anuncios en la página principal del nuevo portal.

- Línea temporal de Twitter, *feed* de Twitter, o *timeline*: Con el objetivo de fomentar la interacción de los visitantes con las redes sociales de la Escuela, se dispone de una sección en el espacio restante a la izquierda del carrusel para mostrar un resumen de las últimas publicaciones del perfil oficial de la Escuela en la red social Twitter. La lista de publicaciones se actualizará en tiempo real a través de la API (Interfaz de Programación de Aplicaciones) proporcionada por la propia red social.

Tweets por @ETSIT_UVa



Ilustración 23 - Línea temporal del perfil de Twitter de la Escuela.



Ilustración 24 - Vista completa de la primera sección horizontal.

- Enlaces de interés: Debajo de este bloque horizontal existe un segundo bloque dedicado a los enlaces de interés. En esta sección se dará visibilidad a enlaces a contenidos diversos y que, en algunos casos, estarán dirigidos a visitantes frecuentes. Estos contenidos serán páginas con alto volumen de tráfico a las que se quiere facilitar el acceso.



Ilustración 25 - Vista general de la sección de enlaces de interés.

El ancho completo del bloque estará dedicado a los enlaces de interés, que se compondrán de una imagen, un título, y un breve texto explicativo. A simple vista sólo la imagen y el título serán visibles, y en el momento en que el usuario coloque el cursor del ratón encima del enlace se desplegará el texto explicativo.

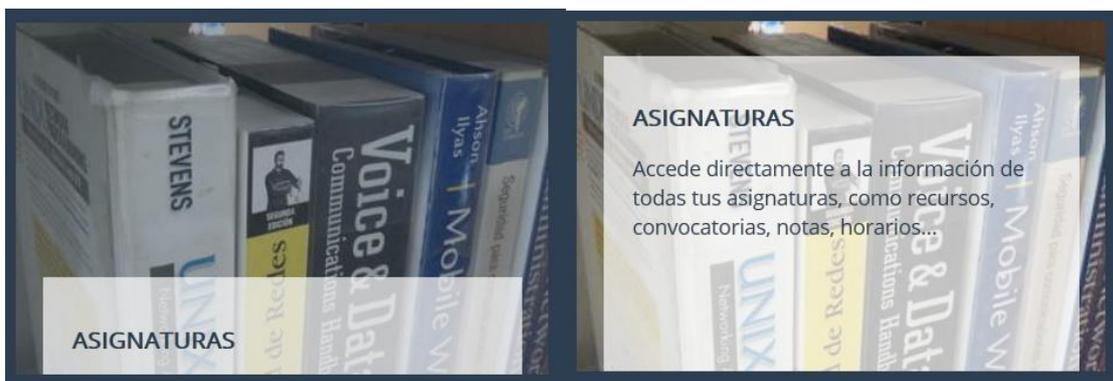


Ilustración 26 - Vista de un enlace de interés. Izquierda: Inactivo, sólo título visible. Derecha: Activo (puntero por encima), visible título y descripción.

Al igual que los elementos del carrusel, el código de los enlaces de interés ha sido escrito de tal manera que no se requieran grandes conocimientos de HTML ni CSS.

```

<a class="col-xs-12 col-sm-6 col-md-4" href=" " target="_blank">
  <div class="interes">
    
    <div class="interesContenido">
      <h1> </h1>
      <p> </p>
    </div>
  </div>
</a>

```

Código 2 – Código HTML de un enlace de interés vacío.

Crear un nuevo enlace de interés consiste en añadir a la plantilla mostrada los tres elementos mencionados más el enlace: la ruta de la imagen de fondo se coloca en el atributo "src" del elemento *image*, el enlace dentro del atributo "href" del elemento *anchor*, el título dentro del elemento "h1", y el texto explicativo dentro del elemento "p". Una vez seguidos estos pasos, basta con añadir el bloque completo a la lista de enlaces de interés, y gracias al código CSS existente, los enlaces se colocarán automáticamente.

Una característica a destacar de esta sección es que una persona con conocimientos más avanzados de desarrollo web puede añadir fácilmente enlaces de interés personalizados con otros formatos. Un ejemplo es la inclusión del video de presentación de la Escuela alojado en el portal de *streaming* de video online Youtube. En lugar de una imagen de fondo, título y texto, el elemento *anchor* contiene un elemento *iframe* que cargará el reproductor embebido de Youtube en su lugar.



Ilustración 27 - Enlace de interés personalizado, con el vídeo de Youtube incrustado.

Inicialmente se han creado 6 espacios para enlaces de interés divididos en filas de tres, pero el código CSS de la sección está diseñado para poder modificar esta estructura de forma relativamente sencilla, añadiendo o eliminando filas, y cambiando el número de elementos en cada fila. Este proceso se explicará más adelante, ya que está estrechamente relacionado con el desarrollo de la versión para móviles y tabletas.

Para poder visualizar esta sección, el visitante tendrá que realizar *scroll* hacia abajo, razón por la cual se prioriza el área para contenido dirigido a los visitantes frecuentes, que probablemente conozcan de antemano el contenido de la misma.

En esta área se localiza, además de enlaces a páginas internas del portal de la Escuela, el enlace al portal de la biblioteca de la Universidad de Valladolid (retirado posteriormente).

- Pie de página o *footer*: Inmediatamente debajo de los enlaces de interés se encuentra el *footer* que cierra la página principal. Esta sección se emplea, como se ha visto también en otras páginas analizadas, a modo de área de contacto, y alberga tres secciones:
 - o Enlaces a las redes sociales oficiales de la Escuela. Al igual que otras secciones explicadas, se ha intentado que esta sección fuese fácilmente modificable sin mayores conocimientos de desarrollo web. Siguiendo la plantilla para el enlace, basta con añadir la dirección del hipervínculo dentro de las comillas en el atributo "href", y la imagen a mostrar en el atributo "src". La sección está diseñada para poder albergar múltiples enlaces además de los iniciales, en caso de que la Escuela amplíe su presencia a otras redes sociales.

Se pretende con esto dar a conocer los perfiles de la Escuela en las redes,

```
<a target="_blank" href=""><img src="" /></a>
```

Código 3 - Código HTML de un enlace a red social vacío.

- o Botones para cambiar entre la versión por defecto (en castellano) y la versión en inglés de la página. La página en inglés cae fuera del foco del proyecto, pero se ha mantenido un enlace en el footer para que los visitantes puedan acceder a ella. El funcionamiento es similar al de los enlaces a las redes sociales, en caso de que en un futuro se quisiera adaptar el portal a nuevos idiomas.
- o Contacto legal: Dirección postal y teléfono de la Escuela.

LA ESCUELA EN LAS REDES



SITIO EN ESPAÑOL / ENGLISH SITE



(c) 2016 :: Escuela Técnica Superior de Ingenieros de Telecomunicación
Campus Miguel Delibes
Paseo Belén 15
47011 Valladolid, España
983423660

Ilustración 28 - Pie de página del nuevo portal

Como se puede apreciar, la mayoría de peticiones y especificaciones impuestas por el cliente han sido incluidas satisfactoriamente, incluida la paleta de colores azul y naranja, que está presente tanto en el diseño de la página web como en el logotipo conmemorativo por el 25º aniversario de la Escuela. La única petición que quedó desatendida fue la de mantener la barra de navegación secundaria en la parte izquierda de la página, si bien la nueva funcionalidad de la barra superior, así como la inclusión del área de enlaces de interés, cumplen sus funciones.

Dicha barra sí estará presente en las páginas secundarias, al no estar presente en estas la zona de enlaces de interés.



Tweets por @ETSIT_UVA

ETSIT_UVA @ETSIT_UVA
La detección de ondas gravitacionales gana el premio Princesa de Asturias de Investigación [#ipasa.com/ipasa/2017/06... via @el_pais](#)

La investigación sob...
39 candidaturas de 17... [#ipasa.com](#)

ETSIT_UVA @ETSIT_UVA
La nota de corte de la

Grado en Administración y Dirección de Empresas

Grado en Ingeniería de Tecnologías de Telecomunicación

PROGRAMA DE ESTUDIOS CONJUNTO TELECO - ADE

ANUNCIOS Y AVISOS

AVISOS DESTACADOS

TFM-MUITIC Comisiones evaluadores y fechas de defensa
TFG adjudicados, tribunales y fechas de defensa
TFM (IT) adjudicados, tribunales y fechas de defensa curso 2016-17
[Avisos y Oferta de TFG/TFM](#) ...Más avisos

CONFERENCIAS Y TALLERES

(29 Junio) Jornada de ORIENTACIÓN Y SALIDAS PROFESIONALES

EMPLEO Y PRÁCTICAS

Open positions in Big Data at Vestlandsforskning, Norway
Telefónica abre una nueva convocatoria de becas Talentum
...Más ofertas

ASIGNATURAS

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN Y GRADO EN ADMINISTRACIÓN Y DIRECCIÓN DE EMPRESAS

DOBLE GRADO TELECO-ADE

25 aniversario ETSIT UVA

XV ANIVERSARIO ETSIT

ORIENTATELECOS

Aprendiendo Telecomunicaciones en

BLOG ORIENTATELECOS ABRIL DE 2015

INVESTIGACIÓN

ETS Ingenieros de Telecomuni...

Universidad de Valladolid Centros

Escuela de Ingeniería de Telecomunicación

UVA

LA ESCUELA EN LAS REDES



SITIO EN ESPAÑOL / ENGLISH SITE



(c) 2016 :: Escuela Técnica Superior de Ingenieros de Telecomunicación
Campus Miguel Delibes
Paseo Belén 15
47011 Valladolid, España
983423660

Ilustración 29 - Vista global de la nueva página principal (completa) de la Escuela.

2.1.3.3 – Páginas secundarias

El diseño de las páginas secundarias mantiene, por lo general, la estructura de la página principal en cuanto a la disposición de los elementos. La mayoría de dichas páginas comparten una plantilla (un archivo .JSP con código HTML y Java listo para ser compilado y convertido en un archivo .HTML) común, compilada desde un archivo .JSP, en la que se vuelca el contenido correspondiente a de dicha página. Ciertas páginas con funciones específicas, como el calendario, no emplean dicha plantilla sino que disponen de su propio archivo .JSP.

En el momento de compilar cada una de las páginas individuales, el servidor de Java importa las plantillas incluidas en dichas páginas, de modo que una página puede contener varias plantillas diferentes y, a la vez, estas plantillas pueden estar compartidas con otras páginas.

Concretamente en el portal de la Escuela, prácticamente el 100% de las páginas se componen de 3 plantillas; una que contiene la cabecera o *header*, otra que contiene el cuerpo, y otra que contiene el pie de página o *footer*. En el caso de la página principal, y de algunas páginas secundarias con funciones muy específicas, la plantilla del cuerpo recoge directamente el contenido de dicha página. Para la mayoría de las páginas secundarias esto es diferente: la plantilla del cuerpo contiene elementos comunes, como la navegación que se verá más adelante, y una plantilla dentro de esta con el contenido específico de dicha página.

Este funcionamiento permite crear páginas modulares en las que parte de los elementos son repetidos entre muchas páginas, sin tener que reescribir y, especialmente, mantener una copia de ese código para cada página. Con el uso de las plantillas, solamente será necesario mantener una única copia de los elementos comunes como la cabecera o el pie.



Ilustración 30 - Esquema de los elementos y estructura de las páginas secundarias del nuevo portal. Código de colores explicado en el capítulo 2.1.1. Espacio central en blanco dedicado a contenido específico de la página.

Sobre estas líneas se muestra el esquema en el que se detalla la estructura y composición de las páginas secundarias. El código de colores es el mismo que en ocasiones anteriores, dejando en blanco el espacio central dedicado al contenido específico de cada página:

- En la parte superior se encuentra la sección de *header*. Dicha sección es igual al *header* de la página principal, con la diferencia de que no aparece el banner conmemorativo del 25º aniversario de la Escuela, con el logotipo y el nombre. Los enlaces al portal web de la Universidad de Valladolid y al área de login, así como la barra de navegación, permanecen iguales a la página principal, ya que la plantilla de *header* es común para todas las páginas, pero no contiene el banner, que es añadido *a posteriori* en la página principal.
- En la parte inferior se encuentra el mismo *footer* que en la página principal.
- El cuerpo de la página contiene los elementos específicos de cada página, y está dividido en dos columnas:
 - o La columna izquierda, ocupando en torno a un 25% del ancho de la sección, contiene un área de navegación secundaria. Esté área guarda similitud con la barra de navegación auxiliar que se encontraba en la página principal de la versión anterior del portal, y se corresponde con la petición del cliente de mantener una segunda barra de navegación con mayor profundidad.

La barra de navegación superior, como se describió anteriormente, contiene enlaces de primer y segundo nivel. Esto junto con la sección de enlaces de interés cubre todas las necesidades de navegación desde la página principal, pero es necesario permitir el acceso a páginas de niveles más profundos de alguna manera. Para esto la barra lateral incluida en las páginas secundarias muestra los enlaces a páginas de segundo nivel dependientes de la sección de primer nivel en la que se encuentra la página, además de los enlaces a páginas de tercer nivel dependientes de la página en la que se encuentra, de haberlos.

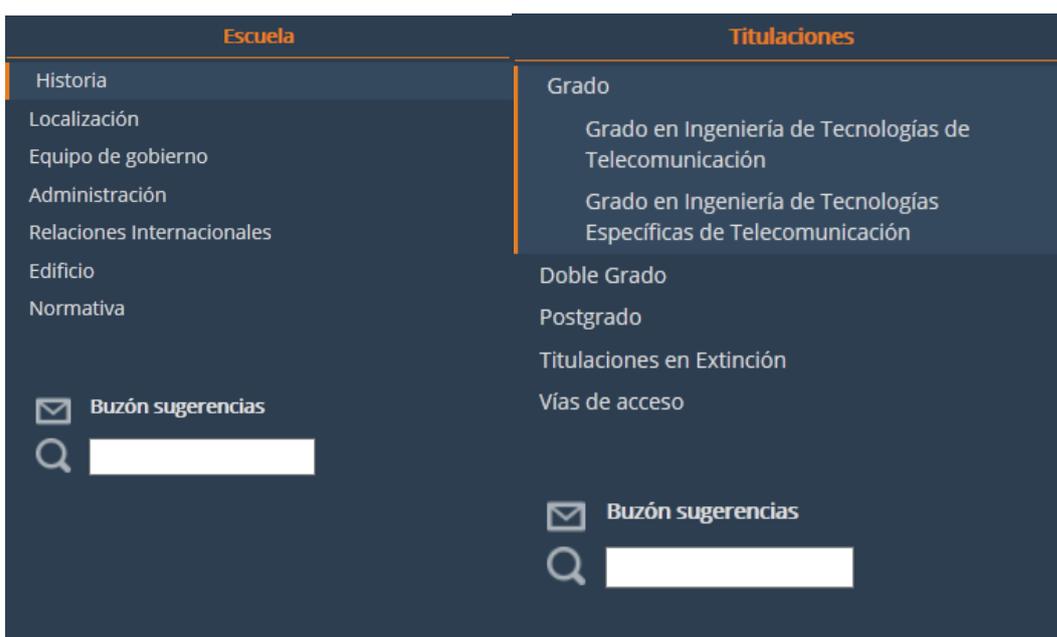


Ilustración 31 - Ejemplo de navegación de dos niveles (izquierda) y de tres niveles (derecha) en la barra de navegación lateral.

Además de la navegación, en esta barra lateral también se encuentra un enlace al buzón de sugerencias abierto a todos los visitantes, así como un cuadro de búsqueda interna. La búsqueda interna emplea el motor de búsqueda de Google para buscar coincidencias en las páginas web dentro del dominio tel.uva.es.

- La segunda columna, ocupando el 75% del ancho de la sección, es la que presenta todo el contenido específico de la página. Este contenido es el que conforma la página secundaria y por tanto es único y varía para cada una.

Si bien la mayoría de páginas secundarias presentan una estructura muy homogénea, variando únicamente en el contenido, existen ciertas páginas dedicadas a funciones muy dispares, como es el caso del foro o el tablón de anuncios. Se ha puesto mucho esfuerzo en que cada una de estas páginas mantenga sus funciones características sin romper la estética y el diseño del conjunto del portal.

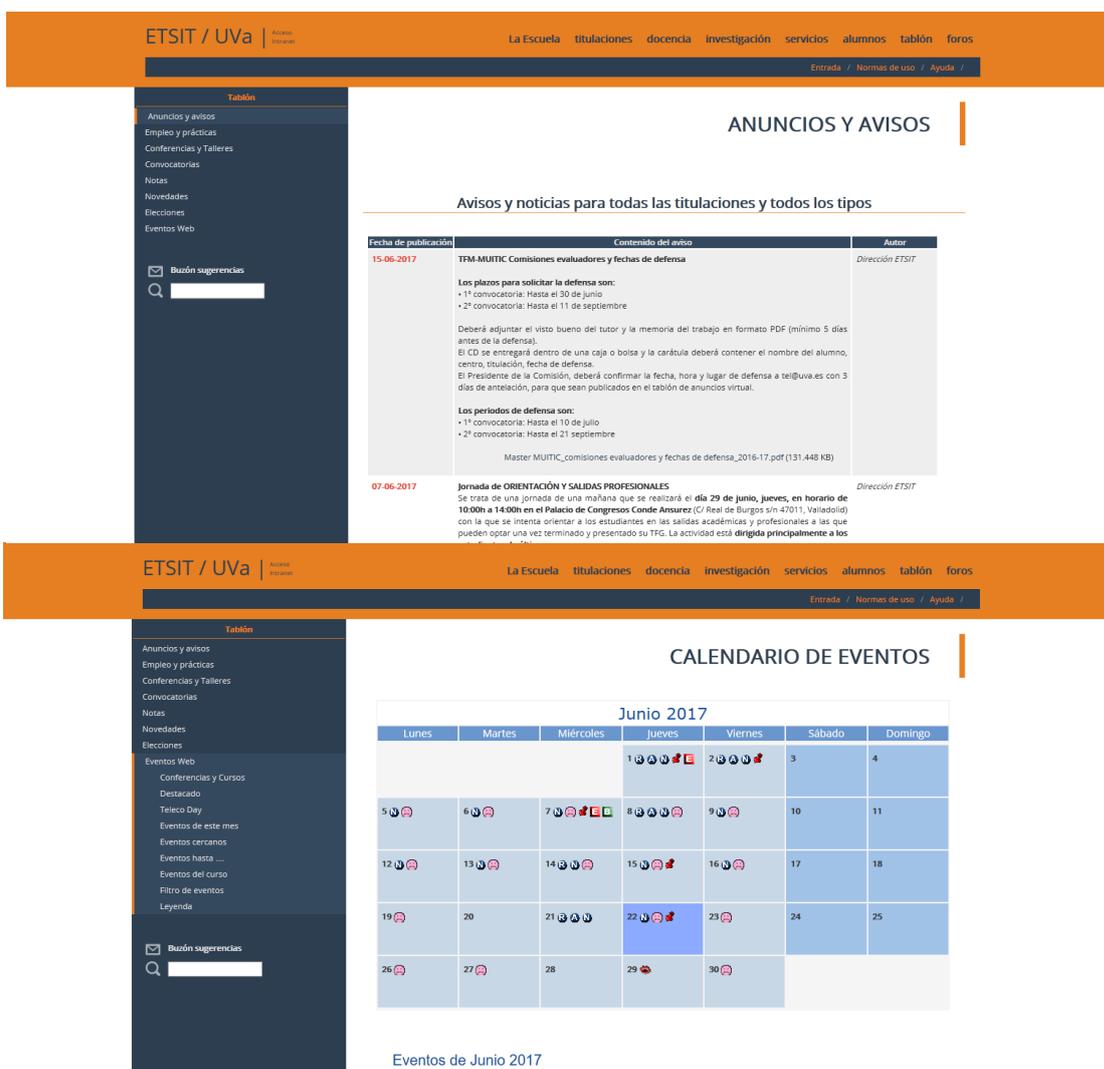


Ilustración 32 - Ejemplos de páginas secundarias de propósitos especiales. Superior: tablón de anuncios. Inferior: Calendario de eventos.

ETSIT / Uva ANEXO 100 años La Escuela titulaciones docencia investigación servicios alumnos tablón foros

Historia / Localización / Equipo de gobierno / Administración / Relaciones Internacionales / Edificio / Normativa /

Escuela

- Historia
- Localización
- Equipo de gobierno
- Administración
- Relaciones Internacionales
- Edificio
- Normativa

Buzón sugerencias

HISTORIA

Presentación

La Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid, se creó el 27 de septiembre 1991. Tiene como misión básica la formación científica y técnica y la preparación para el ejercicio profesional de Ingenieros que han de realizar las funciones que la sociedad actual demanda. Además está comprometida con la creación y desarrollo de la ciencia y de la técnica mediante la investigación y el estudio, colaborando con las empresas en el desarrollo tecnológico del país y de la Comunidad de Castilla y León, contribuyendo de esta forma a la consolidación de Sociedad de la Información.

La Escuela se encarga asimismo, de la gestión administrativa y de la organización de las enseñanzas para la obtención de títulos de Grado y Máster en el ámbito de la Ingeniería de Telecomunicación, coordinando la docencia que imparten los Departamentos de acuerdo con los Planes de Estudio y los objetivos de las titulaciones. Cuenta para ello con unos recursos humanos que, junto con unas instalaciones nuevas y actualizadas, permiten cumplir sus objetivos al nivel de las Escuelas más prestigiosas del país.

La ETSIT, cumpliendo su principio fundacional y a pesar de su juventud, ha estado integrada en todo momento en la Universidad de Valladolid y en la sociedad Castellano-Leonesa, contribuyendo a su desarrollo como lo demuestra su pertenencia a foros de investigación, desarrollo y potenciación de las telecomunicaciones y de la electrónica, tanto nacionales como internacionales, implicándose en particular en la potenciación del mundo empresarial regional, estando presente allí donde las necesidades de estos sectores lo demandan.

Ven a la ETSIT (presentación en Vídeo)

Titulaciones impartidas

Programa de Estudios Conjunto de Grado en Ingeniería de Tecnologías de Telecomunicación y Grado en Administración y Dirección de Empresa (10 plazas de nuevo acceso)

Grado en Ingeniería de Tecnologías de Telecomunicación (70 plazas de nuevo acceso)

Grado en Ingeniería de Tecnologías Específicas de Telecomunicación (100 plazas de nuevo acceso)

- Mención en Ingeniería de Sistemas de Telecomunicación
- Mención en Ingeniería de Sistemas Electrónicos
- Mención en Ingeniería Telemática

Máster en Ingeniería de Telecomunicación (50 plazas de nuevo acceso)

Máster en Big Data Science (20 plazas de nuevo acceso)

Máster en Investigación en Tecnologías de la Información y las Comunicaciones (40 plazas de nuevo acceso)

Doctorado en Tecnología de la Información y las Telecomunicaciones

LA ESCUELA EN LAS REDES

SITIO EN ESPAÑOL / ENGLISH SITE

(c) 2016 :: Escuela Técnica Superior de Ingenieros de Telecomunicación
Campus Miguel Delibes
Paseo Belén 15
47011 Valladolid, España
983423660

Ilustración 33 - Página secundaria de propósito general.

La respuesta del cliente fue muy positiva, y a partir de este punto se considera que la renovación del portal ha llegado a un estado apto para comenzar a desarrollar la versión para móviles y tabletas, ya que el único trabajo restante en el diseño de las páginas para escritorio consiste en la corrección de errores.

2.2 – RESPONSIVE WEB DESIGN

Como se ha explicado anteriormente, el Responsive Web Design consiste en el diseño de páginas web que adapten su estructura al tamaño de la pantalla en la que se visualizan. Esto se realiza principalmente a través de clases CSS que entran en funcionamiento en función de ciertos parámetros detectados por el navegador. Este proceso contrasta con corrientes de diseño web para móvil obsoletas en las que, en lugar de adaptar y recolocar los elementos en función de la pantalla, se servía una página nueva construida especialmente para ese tamaño de pantalla.

La característica de CSS que permite esta funcionalidad son las Media Queries. Una Media Query consiste en la especificación de un tipo de medio y cero o más expresiones que comprueben si se cumplen unas condiciones en alguna de las características de dicho medio. Estas Queries, o “consultas”, son expresiones lógicas que pueden ser *true* (verdadera) o *false* (falsa). La consulta es verdadera o *true* cuando el medio en el que se está ejecutando el código cumple con las condiciones especificadas, y es falsa de lo contrario.

A continuación de la consulta puede haber cero o más reglas CSS. En todo momento, todas las reglas CSS incluidas dentro de Media Queries que se evalúen como verdaderas son aplicadas, y ninguna de las reglas CSS dentro de Media Queries que evalúan como falsas se aplica.

El Responsive Web Design emplea estas Media Queries comprobando la resolución de pantalla en la que se está renderizando la página web, y aplicando reglas CSS que configuran los elementos de manera óptima en función de dicha resolución.

```
@media screen and (max-width: 1000px) and (min-width: 700px) {
  a {
    color: red;
  }
}

@media screen and (max-width: 700px) and (min-width: 400px) {
  a {
    color: blue;
  }
}
```

Código 4 - Ejemplo de uso en CSS de Media Queries. Arriba: regla para pantallas entre 700px y 1000px de ancho. Abajo: regla para pantallas entre 700px y 400px de ancho.

En el ejemplo anterior, la regla CSS del primer bloque se aplicará únicamente en pantallas con un ancho de entre 700 y 1000 píxeles, pintando el texto de todos los hipervínculos de la página de color rojo. Si la pantalla tiene un ancho entre 400 y 700 píxeles, se aplicará la regla del segundo bloque, pintando los hipervínculos de color azul. En la práctica, la principal aplicación de Media Queries en el Responsive Web Design consiste en reorganizar la disposición de los elementos dentro de la página, modificar sus tamaños, e incluso ocultar o hacer visibles ciertos elementos, en función de la resolución.

2.2.1 –HERRAMIENTAS PARA DISEÑO ADAPTATIVO (Bootstrap)

Cuando se hace uso de Media Queries para crear diseños web adaptativos, se está añadiendo un nivel de complejidad extra a la creación y mantenimiento de las hojas de estilo CSS. El tradicional flujo de trabajo con CSS consiste en añadir clases e IDs a los elementos HTML de la página, y dotar a esas clases e IDs de propiedades CSS creando una serie de reglas en las hojas de estilos. Al emplear Media Queries se añade un paso adicional, pues se vuelve necesario repetir dichas reglas varias veces, en función de los tamaños de pantalla deseados, para reorganizar los elementos de la página. Esta tarea resultaría enormemente trabajosa de no ser por la existencia de *frameworks* especializados.

Con la popularización de esta corriente de diseño web han surgido gran cantidad de librerías y *frameworks* CSS que incorporan multitud de clases y reglas, basadas en Media Queries, específicamente dedicados al diseño para móviles. Estas herramientas permiten al desarrollador centrarse en el diseño de la página eliminando la necesidad de escribir todas las hojas de estilo, de manera que la principal tarea consiste en aplicar las clases ya preparadas a los elementos de la página.

Para la realización de la versión responsiva del portal de la Escuela, se analizaron los principales *frameworks* de código abierto con el objetivo de determinar el que más se adecuara a las necesidades del proyecto. Durante el periodo en el que se realizó este TFG, los dos *frameworks* para Responsive Web Design más extendidos eran Bootstrap, desarrollado por Twitter, y Foundation, de Zurb. Elegimos emplear Bootstrap de Twitter por las siguientes razones:

- Extensa comunidad, siendo el *framework* de diseño para móviles más popular en el mercado.
- Enormes opciones de personalización disponibles al descargar el framework desde su página web, incluidos módulos para elementos HTML como cabeceras y barras de navegación.
- Adaptación más sencilla de tipografías a tamaños de pantalla.
- Experiencia previa con este *framework* gracias a proyectos personales anteriores.

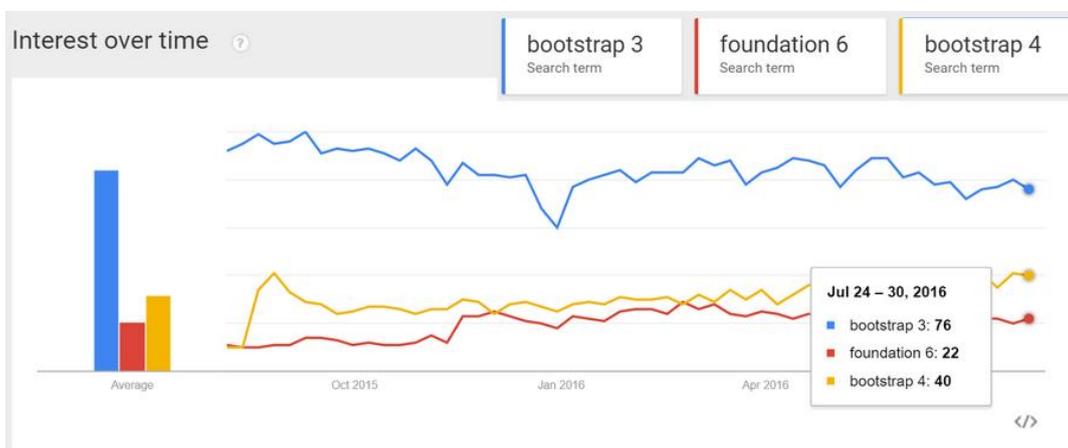


Ilustración 34 - Comparación de frameworks para diseño web móvil. Fuente: <https://trends.google.com/trends/explore?date=today%2012-m&q=bootstrap%203,foundation%206>

Hay que destacar que, tanto Bootstrap como Foundation, son más que simples colecciones de clases para Responsive Web Design. Ambas cuentan con una gran colección de herramientas de diseño web front-end que serán útiles para la realización del portal de la Escuela. Concretamente, Bootstrap cuenta con multitud de componentes listos para ser incluidos en las páginas sin apenas trabajo, como barras de navegación, botones, alertas...

Bootstrap emplea un sistema llamado “de rejilla” o *grid* para colocar los elementos dentro de una página. A través de las clases de CSS incorporadas, se divide la página en cualquier número de secciones horizontales o filas. Cada una de estas filas está dividida a su vez, independiente del tamaño de la pantalla en doce secciones verticales iguales, o columnas, formando una matriz o rejilla cuyas celdas se reparten entre los elementos de la página. Añadiendo clases CSS predefinidas a cada elemento, se puede especificar el número de columnas que se desea que abarque dicho elemento, repartiendo de esta manera el ancho de la sección entre sus componentes en función de las clases asignadas a cada uno de ellos.

El funcionamiento para adaptar la rejilla a diferentes tamaños de pantalla se basa en el uso de varias de estas clases simultáneamente. Se puede especificar una clase para cada tamaño de pantalla y estas se seleccionarán automáticamente gracias a las Media Queries. De esta manera, un mismo elemento ocupará diferente cantidad de columnas en función de la resolución del medio en el que se visualiza. Otras clases permiten realizar distintas manipulaciones en la rejilla, como crear espacios vacíos, o reordenar los elementos en función de la resolución.

Típicamente se asigna un mayor número de columnas a un elemento cuanto menor sea el tamaño de pantalla. De esta manera, en pantallas pequeñas los elementos serán más anchos y por tanto más fácilmente legibles. Si varios elementos suman más de doce columnas de ancho en total, los últimos elementos se desplazarán hacia abajo creándose nuevas filas. Todo esto significa que cuando se visualiza página creada con un sistema de rejilla en un móvil, muchos elementos que en escritorio se colocaban en horizontal se verán ahora uno encima del otro, creando páginas más alargadas pero más legibles.

Por defecto, Bootstrap crea cuatro divisiones para englobar a todos los tamaños de pantalla existentes, si bien los límites que separan estas divisiones pueden personalizarse. El desarrollador puede especificar clases a sus elementos para cada una de estas divisiones y, así, crear hasta cuatro configuraciones diferentes.

Clase CSS	Nombre	Resolución
.col-xs-*	Extra pequeño	Menos de 768px (móviles)
.col-sm-*	Pequeño	Más de 768px (tabletas)
.col-md-*	Medio	Más de 992px (escritorio)
.col-lg-*	Grande	Más de 1200px (escritorios grandes)

Tabla 6 - Clases y medidas por defecto para Responsive Web Design en Bootstrap

Para asignar anchos a un elemento en función del tamaño de la pantalla, basta con añadirle una o varias clases CSS de la tabla, sustituyendo el asterisco por la cantidad de columnas de ancho (de las 12 posibles) que se desea que abarque.

Aunque existan cuatro categorías de tamaño, en la práctica el tamaño “grande” existe principalmente para dar soporte a aplicaciones web con interfaces complejas, y no se hace distinción entre “medio” y “grande” cuando se trata de páginas web de propósito general. Por esta razón la versión *Responsive* del portal de la Escuela dará soporte a los tamaños “extra pequeño”, “pequeño” y “grande”.

Bootstrap es un *framework* “mobile first”. Esto significa que por defecto siempre se aplicarán primero los estilos dedicados a pantallas pequeñas, de móviles. En el caso de que la pantalla sea más grande que el tamaño de móviles, se aplicará el estilo del siguiente nivel (tabletas) sobre el anterior, sobrescribiendo sus estilos, y así hasta aplicar el estilo correspondiente al tamaño de la pantalla en la que se renderiza. Si no se ha creado un estilo para el tamaño en el que se está renderizando, este sistema implica que se usarán en su lugar los estilos aplicados anteriormente, para tamaños más pequeños, sirviendo a modo de *fallback*. Por esto al no especificar un estilo para tamaño grande, se aplicarán siempre los estilos del tamaño mediano en pantallas de más de 992px de ancho.

2.2.2 – CREACIÓN DEL PORTAL WEB ADAPTABLE

El siguiente paso consistió en determinar el reordenamiento de los elementos en función del tamaño de la pantalla. Los diseños ya mostrados de las páginas principal y secundarias corresponden a los tamaños “medio” y “grande”, por lo que queda solamente diseñar el aspecto en pantallas pequeñas (tabletas) y extra pequeñas (móviles).

Como ya se ha explicado, a medida que las resoluciones de pantalla se hacen más pequeñas lo normal es que los elementos de una página web ocupen una mayor proporción del ancho de la pantalla. Además es conveniente revisar la ordenación de los elementos ya que, por defecto, los situados más a la izquierda se colocarán encima de los situados a la derecha. Es posible que, por cuestiones de diseño y usabilidad, se desee que un elemento que no esté situado a la izquierda dentro de su sección horizontal se coloque encima de los demás al visualizarse en pantallas más pequeñas. Es el caso de los elementos de la primera sección en la página principal de la Escuela, donde el carrusel de imágenes es el objeto de mayor importancia dentro de la sección, pero acabaría situado por debajo de la línea temporal de Twitter en una pantalla pequeña. Para solucionar esto Bootstrap permite añadir clases que desplazan los elementos en relación a los que tienen a su derecha e izquierda.

Haciendo uso de estas características se crean los dos nuevos diseños para el portal, tanto para la página principal como para las páginas secundarias. Se han reordenado elementos para colocar en la parte superior los más importantes, como el carrusel de imágenes o la navegación. Además, se ha creado una barra de navegación desplegable, aprovechando los objetos prefabricados de Bootstrap, para el tamaño “extra pequeño”, de manera que la barra mantenga sus funcionalidades a la vez que ocupa un espacio mínimo de la pantalla cuando no se está usando.

Los cambios en las páginas se testearon empleando la herramienta de “Vista de Diseño Adaptable” del navegador Firefox, que permite alterar el tamaño del *viewport* (el área sobre el que se renderiza una página) para visualizar el contenido como si se hiciera desde diferentes dispositivos y resoluciones. También se emplearon varios smartphones y una tableta para testear el trabajo en dispositivos reales.

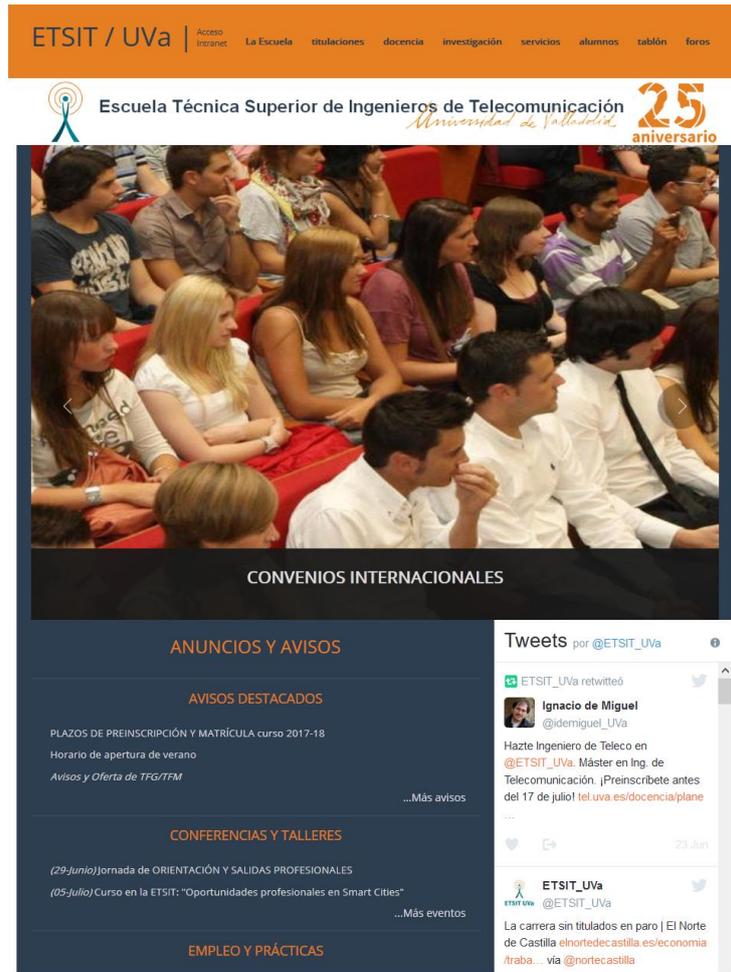


Ilustración 35 - Vista de la nueva página principal en dispositivos móviles. Izquierda: smartphone (360x640px). Derecha: tableta (768x1024).

Finalizado este trabajo, se sometió de nuevo al portal al test de adaptación a móviles (*mobile friendly*) de Google. Los resultados fueron muy positivos, superando todos los controles y obteniendo la calificación de “optimizada para móviles”. El éxito en esta prueba significa que el sitio web de la Escuela aparecerá con mucha mayor prioridad en búsquedas realizadas desde dispositivos móviles.

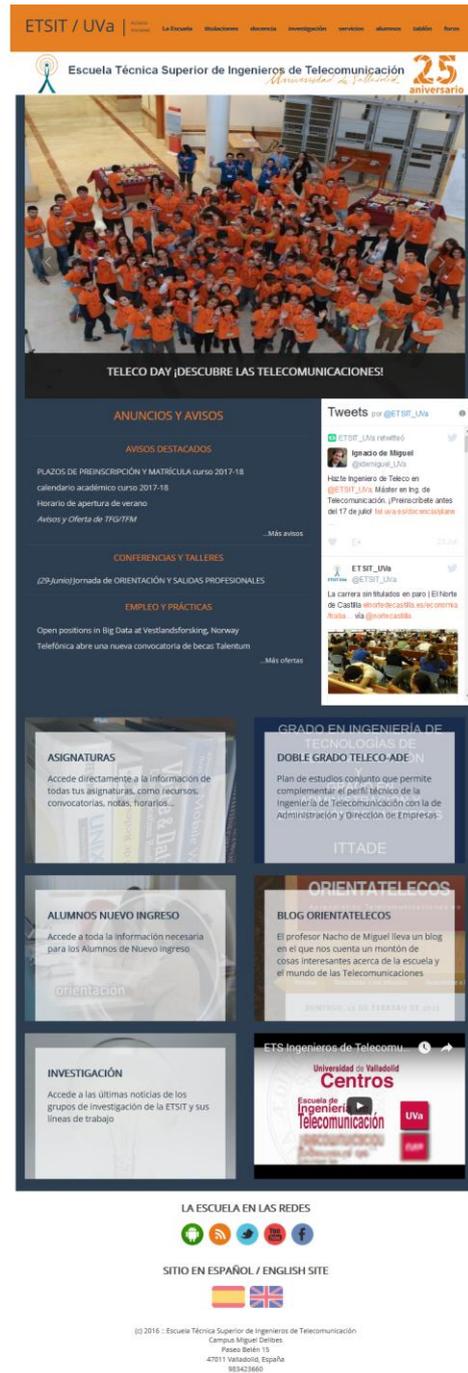


Ilustración 36 - Vista de la página principal completa en pantallas móviles. Izquierda: smartphone (360x640px), dividido en dos. Derecha: tableta (768x1024px).

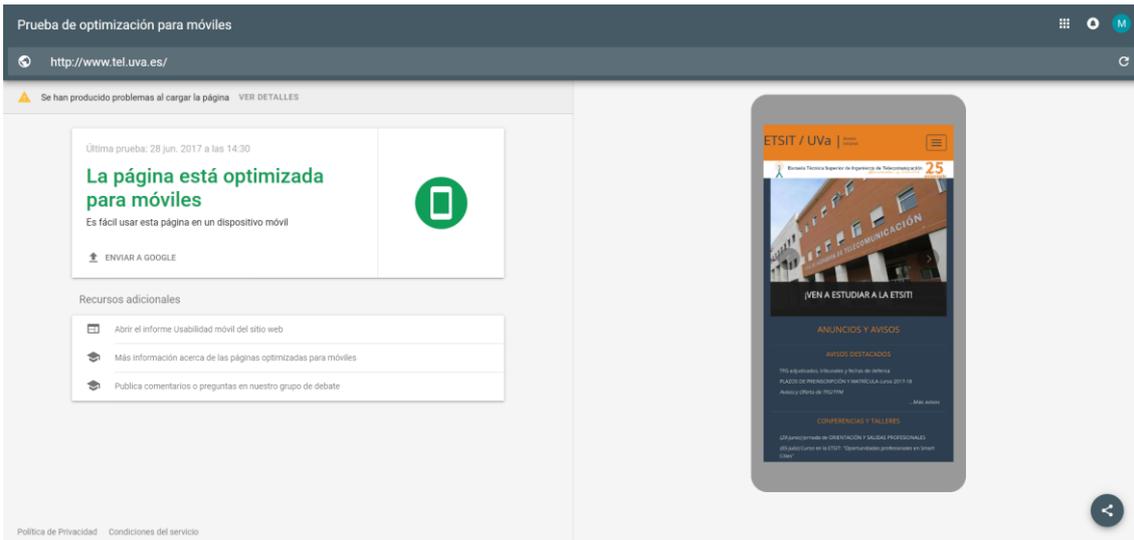


Ilustración 37 - Resultados del test de optimización para móviles de Google en la nueva versión del portal de la Escuela

Una vez finalizada la fase de Responsive Web Design, con el visto bueno de la Junta de la Escuela, el portal estaba listo para ser publicado, por lo que se procedió a la migración de los cambios desde el servidor de pruebas desa.tel.uva.es al servidor de producción (público) www.tel.uva.es. El siguiente paso consistirá en la optimización/programación verde del portal.

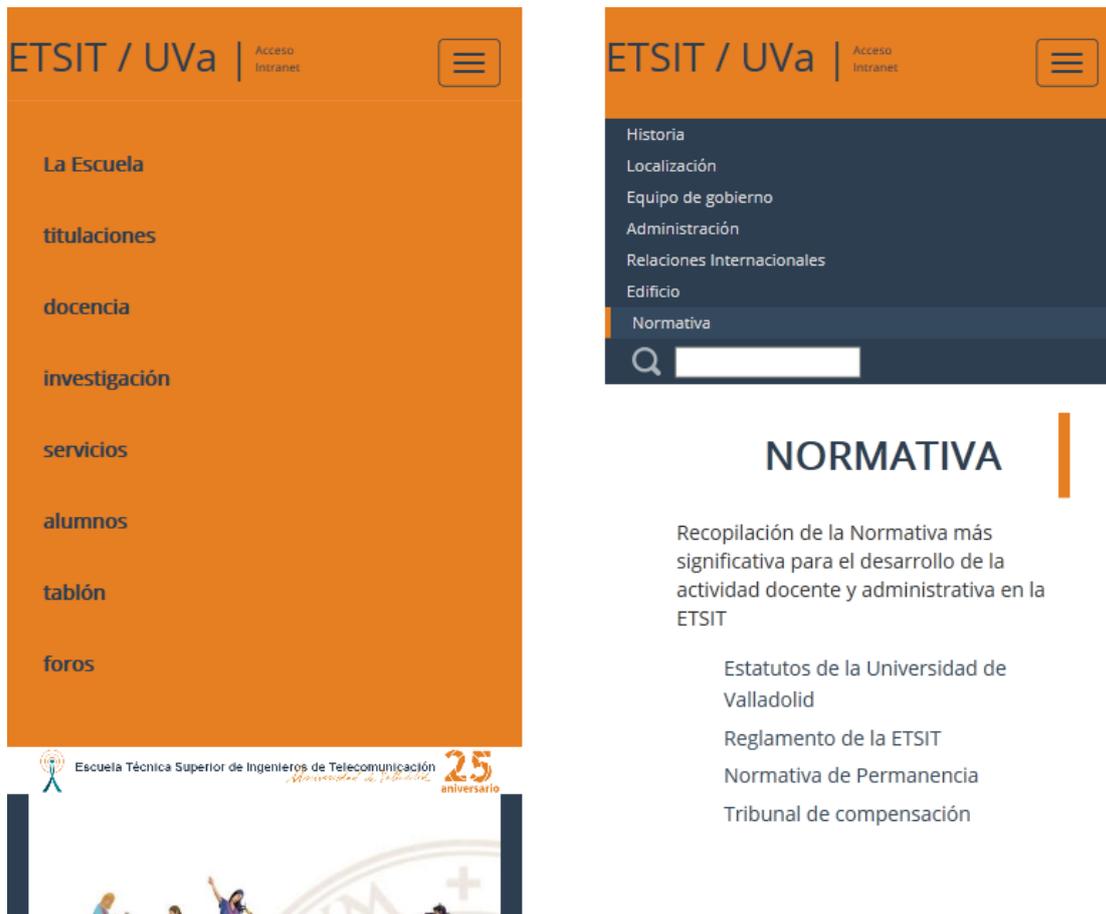


Ilustración 38 - Izquierda: detalle de la barra de navegación superior desplegada en móvil. Derecha: Ejemplo de página secundaria en móvil con barra de navegación principal plegada, y barra de navegación secundaria visible.

2.3 – PROGRAMACIÓN VERDE

Una vez concluida la fase de rediseño y publicado el nuevo portal, comenzó la fase de programación verde, con la que se pretendió dotar de medidas de eficiencia energética a la web. Con estas medidas se quiere reducir la huella medioambiental derivada del funcionamiento de este portal.

Como se comentó anteriormente, los principales frentes de actuación, en cuanto a optimización/programación verde, dentro del foco de este TFG son reducir el ancho de banda consumido (en cuanto al “peso de la página en bytes”) y el gasto de energía en el cliente. Esto se hará a través de la parte *front-end* de la página principalmente, ya que es el ámbito en el que se ha desarrollado la mayor parte del trabajo hasta ahora. Para mejorar la eficiencia energética en el lado del servidor sustancialmente, sería necesario hacer un estudio exhaustivo acerca del funcionamiento del *back-end* del portal lo cual, como ya se mencionó, cae fuera de los límites de este Trabajo de Fin de Grado.

Gracias a lanzar el nuevo portal al público antes de comenzar la fase de optimización, se puede contar con datos realistas de rendimiento que serían difíciles de obtener en un entorno de desarrollo en el que la versión del producto no es la final. Esto es especialmente importante en el caso del rendimiento en el servidor, ya que ni la carga soportada, ni el servidor propiamente dicho, son iguales en un entorno de pruebas que en el de producción. Otros datos, como el gasto energético en el cliente, o el consumo de ancho de banda, son más sencillos de obtener por lo que no sería necesario disponer de la página en el entorno de producción para trabajar con ellos.

Los primeros pasos consistieron en analizar el desempeño de la página en las diferentes áreas, especialmente el consumo de ancho de banda y el gasto energético en dispositivos móviles. A partir de los datos obtenidos se diseñaría un plan de acción para optimizar el portal, buscando mejorar dichas cifras lo más posible sin perder funcionalidades.

2.3.1 – ANÁLISIS DEL CONSUMO DE RED

2.3.1.1 - INTRODUCCIÓN

El volumen de descargas para cargar una página es un factor vital en el éxito o fracaso de un portal web. Un peso excesivamente elevado tiene un efecto directo en los tiempos de carga de la página, y este factor a su vez redundante en la percepción que tienen los usuarios de la misma. Un estudio realizado por Akamai, una de las compañías de Redes de Distribución de Contenido (CDN) del mundo, determinó que reducir el tiempo de carga de las páginas en un segundo se traducía en un 7% más de ventas en promedio, y tres segundos aumentaban esta cifra hasta el 21%. Además, un aumento de dos segundos en el tiempo de carga suponía multiplicar por dos la probabilidad de que un visitante abandone la página antes de completar su carga (Young, 2017). Estas cifras son aún mayores cuando se tiene en cuenta únicamente los usuarios que navegan desde *smartphones*. Si bien el portal de la Escuela no está relacionado con el comercio electrónico, las conclusiones de estos estudios son extrapolables a cualquier ámbito: los tiempos de carga tienen una gran influencia a la hora de determinar si un visitante estará más o menos interesado en el contenido de una página.

Por otra parte, los crecientes volúmenes de descargas de las páginas web suponen un problema para el medio ambiente: mayores transferencias de datos implican la necesidad de construir redes de internet más potentes, aumentan el consumo eléctrico de dichas redes (así como de los dispositivos de servidor y cliente), y obligan a aumentar la cantidad de servidores dedicados. Se vio en el capítulo “1.3.4.1 - El problema medioambiental de la era de las comunicaciones y la necesidad de la computación verde” como todos estos factores suponen un elevado impacto medioambiental.



Ilustración 39 - Comparación de volumen de descargas medio de una página web en 2010 (izquierda) frente a 2017 (derecha). Fuente: <http://httparchive.org/interesting.php>

El factor del volumen de descargas también tiene una importancia especial para los usuarios de teléfonos móviles que, a menudo, están sujetos a tarifas por sus operadoras en función de las cifras totales de datos descargados.

La optimización del portal de la Escuela en este ámbito pretende por tanto crear un sitio web más ligero y rápido, más cómodo para el usuario, ético y responsable con el medio ambiente.

2.3.1.2 - ANÁLISIS DEL PORTAL

Para medir consumo de ancho de banda se emplea la herramienta “Red” del navegador Firefox, que permite visualizar de manera detallada todo el uso de red generado por una página concreta. Con la herramienta abierta se debe recargar la página completamente, eliminando los restos de ficheros de la página almacenados en la memoria caché del navegador, para obtener una lista de todos los archivos descargados con detalles como el tamaño, el dominio desde el que se descargaron, o el tiempo que tomó descargarlos.

Se analizarán por separado la página principal y una página secundaria sin contenido multimedia, para comparar los resultados. Se espera que la página principal presente unas cifras de ancho de banda y consumo mucho mayores que el resto de las páginas, y sea el foco de los esfuerzos sin dejar de lado la optimización general del portal. La importancia de ser el punto de acceso casi todos los usuarios hacia otras secciones del portal, hace que en la página principal sean tolerables unos números que en páginas secundarias no tendrían cabida. Aún así es necesario encontrar un compromiso entre rendimiento y usabilidad que será el foco de este capítulo.

Estado	Método	Archivo	Dominio	Causa	Tipo	Transferido	Tama...	0 ms	10.24 s
200	GET	/	www.tel.uva.es	document	html	5,78 KB	28,67 KB	533 ms	
200	GET	pl.css	www.tel.uva.es	stylesheet	css	12,10 KB	65,73 KB	37 ms	
200	GET	css?family=Open+Sans	fonts.googleapis.com	stylesheet	css	607 B	2,32 KB	83 ms	
200	GET	jquery-1.11.3.min.js	www.tel.uva.es	script	js	32,54 KB	93,71 KB	56 ms	
200	GET	general.js	www.tel.uva.es	script	js	2,09 KB	7,49 KB	39 ms	
200	GET	bootstrap.min.js	www.tel.uva.es	script	js	1,74 KB	4,70 KB	42 ms	
200	GET	logo5.png	www.tel.uva.es	img	png	49,24 KB	49,24 KB	140 ms	
200	GET	portada_25aniversario.jpg	www.tel.uva.es	img	jpeg	60,65 KB	60,65 KB	79 ms	
200	GET	portada_etsit_recta.jpg	www.tel.uva.es	img	jpeg	38,81 KB	38,81 KB	79 ms	
200	GET	RankingUniversidadEmpresa.jpg	www.tel.uva.es	img	jpeg	61,10 KB	61,10 KB	136 ms	
200	GET	excelente.jpg	www.tel.uva.es	img	jpeg	16,92 KB	16,92 KB	110 ms	
200	GET	ityade.jpg	www.tel.uva.es	img	jpeg	35,01 KB	35,01 KB	131 ms	
200	GET	MentumCarrusel.jpg	www.tel.uva.es	img	jpeg	48,23 KB	48,23 KB	228 ms	
200	GET	TelecoDayCarrusel.jpg	www.tel.uva.es	img	jpeg	59,16 KB	59,16 KB	232 ms	
200	GET	Erasmus.jpg	www.tel.uva.es	img	jpeg	52,44 KB	52,44 KB	422 ms	
200	GET	Genso.jpg	www.tel.uva.es	img	jpeg	37,17 KB	37,17 KB	334 ms	
200	GET	SalonGrados.jpg	www.tel.uva.es	img	jpeg	38,23 KB	38,23 KB	331 ms	
200	GET	BibliotecaCampus.jpg	www.tel.uva.es	img	jpeg	55,55 KB	55,55 KB	380 ms	
200	GET	graduacion1516carrusel.jpg	www.tel.uva.es	img	jpeg	36,58 KB	36,58 KB	357 ms	
200	GET	Aula.jpg	www.tel.uva.es	img	jpeg	40,25 KB	40,25 KB	387 ms	
200	GET	DeportesUVA.jpg	www.tel.uva.es	img	jpeg	38,79 KB	38,79 KB	418 ms	
200	GET	widgets.js	platform.twitter.com	script	js	32,29 KB	113,30 KB	53 ms	
200	GET	asignaturas.jpg	www.tel.uva.es	img	jpeg	14,78 KB	14,78 KB	411 ms	
200	GET	ITTADE.jpg	www.tel.uva.es	img	jpeg	26,58 KB	26,58 KB	434 ms	
200	GET	orientaj.jpg	www.tel.uva.es	img	jpeg	13,49 KB	13,49 KB	441 ms	
200	GET	orientatelecos.jpg	www.tel.uva.es	img	jpeg	16,19 KB	16,19 KB	433 ms	
200	GET	bombilla.jpg	www.tel.uva.es	img	jpeg	4,91 KB	4,91 KB	443 ms	
200	GET	android.png	www.tel.uva.es	img	png	2,89 KB	2,89 KB	450 ms	
200	GET	1,84 KB	1,84 KB	453 ms	

Tabla 7 - Resultados de la utilización de red en la página principal (tel.uva.es)

Analizando los resultados obtenidos, se puede ver que la cantidad total de datos descargados para mostrar la página principal es de 3,92MB, repartidos entre 81 peticiones, en 10,94 segundos. Sin embargo, si analizamos con más detalle la gráfica podemos destacar que, a partir de los cerca de 2,1 segundos desde la recarga de la página, todas las peticiones y los datos transferidos provienen de los dominios twitter.com, twimg.com (CDN de Twitter) y youtube.com. Esto significa que el contenido de la página está cargado a los 2,1 segundos, y desde ese momento la línea temporal de Twitter y el video de Youtube tardan otros 8

segundos en cargarse. Además, cada 30 segundo se realiza una nueva petición a twimg.com. Esta petición se corresponde con la actualización en tiempo real de la línea temporal de Twitter, que busca nuevos mensajes cada 30 segundos.

Filtrando por el dominio de procedencia de los archivos, se observa un volumen de descarga de 1,03MB desde twimg.com (servidor de imágenes de Twitter) y twitter.com, y 1,56MB desde youtube.com. Esto significa que no solo ralentizan preocupantemente la carga de la página, sino que los *widgets* de Twitter y Youtube suponen un 66% del total de datos descargados para visualizar la página principal.

Desde el dominio tel.uva.es se descargan 1,13MB. De estos, 758KB se corresponden con imágenes, y el resto con archivos de texto (.html, .css y .js). La mayor parte del volumen de descargas procede del carrusel de fotografías de la página principal, al ser estas las imágenes de mayor resolución.

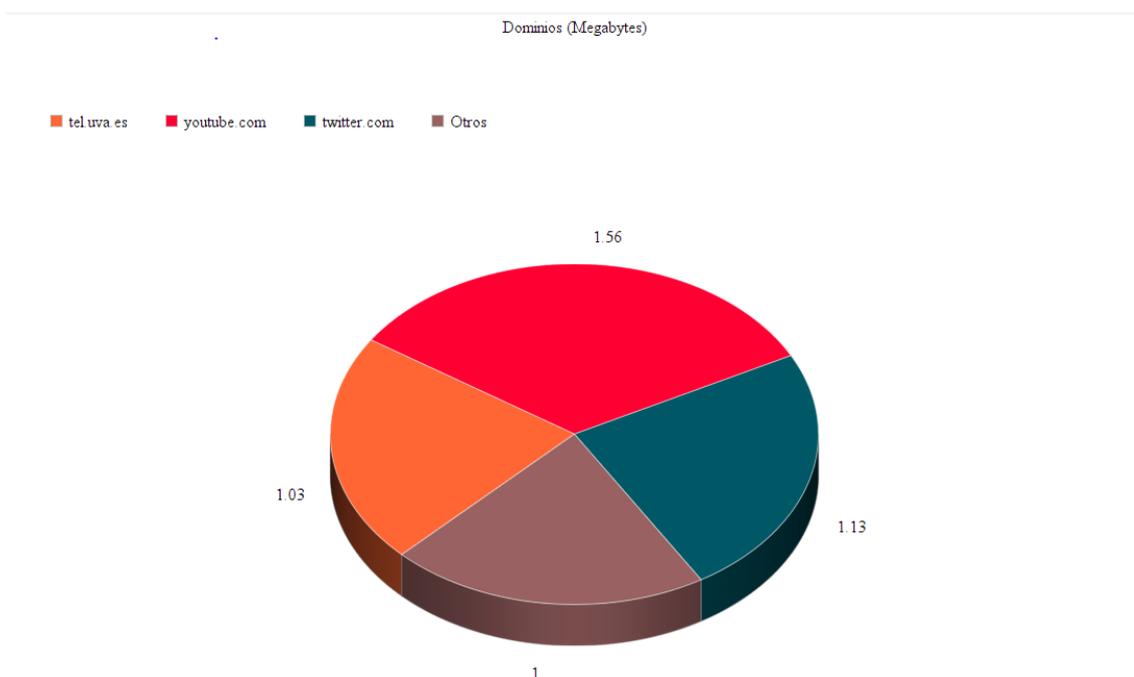


Ilustración 40- Distribución de volumen de descargas para tel.uva.es, en función del dominio de procedencia. Valores en megabytes.

Para realizar un análisis de alguna de las páginas secundarias, se ha seleccionado la página de “Historia”, alojada en www.tel.uva.es/información/historia. Esta página no contiene ningún elemento multimedia, únicamente texto plano y enlaces a otras páginas internas, por lo que será ideal para obtener una idea del peso de una página secundaria básica. El contenido de imágenes es muy pequeño, y tampoco tiene elementos de Twitter o Youtube, que solo se encuentran en la página principal.

Estado	Método	Archivo	Dominio	Causa	Tipo	Trans...	Tama...	0 ms
200	GET	historia.htm	www.tel.uva.es	docum...	html	3,83 KB	14,40 KB	332 ms
200	GET	pl.css	www.tel.uva.es	stylesheet	css	12,10 KB	65,73 KB	292 ms
200	GET	css?family=Open+Sans	fonts.googleapis...	stylesheet	css	607 B	2,32 KB	141 ms
200	GET	jquery-1.11.3.min.js	www.tel.uva.es	script	js	32,54 KB	93,71 KB	170 ms
200	GET	general.js	www.tel.uva.es	script	js	2,09 KB	7,49 KB	135 ms
200	GET	bootstrap.min.js	www.tel.uva.es	script	js	1,74 KB	4,70 KB	132 ms
200	GET	icono_mail.png	www.tel.uva.es	img	png	365 B	365 B	71 ms
200	GET	icono_lupa.png	www.tel.uva.es	img	png	531 B	531 B	85 ms
200	GET	android.png	www.tel.uva.es	img	png	2,89 KB	2,89 KB	85 ms
200	GET	rss.png	www.tel.uva.es	img	png	1,84 KB	1,84 KB	82 ms
200	GET	ttr.png	www.tel.uva.es	img	png	1,38 KB	1,38 KB	82 ms
200	GET	ytb.png	www.tel.uva.es	img	png	1,60 KB	1,60 KB	19 ms
200	GET	fbcbk.png	www.tel.uva.es	img	png	0,99 KB	0,99 KB	84 ms
200	GET	flag_sp.png	www.tel.uva.es	img	png	397 B	397 B	91 ms
200	GET	flag_uk.png	www.tel.uva.es	img	png	3,57 KB	3,57 KB	85 ms
200	GET	cJZKeOuBm4kERxqtaUH3VtXRa8T...	fonts.gstatic.com	font	woff2	13,64 KB	13,64 KB	145 ms
200	GET	pl.css	www.tel.uva.es	styleshe...	css	cacheado	65,73 KB	

Tabla 8 - Resultados de la utilización de red en una página secundaria (Historia)

El peso total de esta página es mucho menor que el de la página principal, 281,25KB, aproximadamente un 7% de lo que pesaba la primera. Se han realizado 17 peticiones, y la página se cargó en aproximadamente 1,3 segundos, sin ninguna petición adicional a partir de ese instante. De los 281KB, la gran mayoría, 240KB, se reparten entre los diferentes archivos de Javascript y CSS, archivos de texto.

Se han realizado las mismas mediciones para las páginas web de otras facultades y escuelas analizadas en el capítulo “Análisis de portales web de ámbito similar”, con el fin de tener una referencia para comparar los resultados del presente capítulo. El procedimiento para realizar las mediciones fue el mismo que el empleado para las del portal de la Escuela, con la única diferencia de que no se analizaron páginas secundarias. Esto es debido a que, por una parte el propósito y estructura de las páginas secundarias es enormemente dispar entre los portales de los diferentes centros; y por otro lado los resultados obtenidos al analizar las páginas secundarias de la Escuela han sido muy satisfactorios, por lo que se centrarán los esfuerzos de este bloque de trabajo en optimizar la página principal.

Página	Peso	Transferido	Peticiones	Tiempo
ETSETB	4,52mb	3,05mb	61	9s
ETSIIT Granada	1,15mb	462kb	59	3s
ETSIT UPM	1,88mb	1,34mb	46	6s
Inf.Uva	914kb	514kb	95	7s
Físicas UCM	630kb	630kb	28	5s

Tabla 9 - Análisis del consumo de red y tiempo de carga de otros portales web de Centros Universitarios analizados.

Se observa que, con la salvedad de la ETSETB de Barcelona, los portales analizados son sustancialmente más ligeros y rápidos que el portal de la ETSIT de Valladolid en su presente estado. Además, el grueso del volumen descargado en la página de la ETSETB se corresponde con imágenes muy pesadas en el carrusel de imágenes central, similar al del portal de la

Escuela, y tuvo grandes variaciones en sucesivos análisis a medida que los elementos anunciados en el carrusel se eliminaban o sustituían.

Se espera en la siguiente sección reducir el consumo de ancho de banda para, al menos, equipararlo con los niveles de las demás páginas analizadas, y si es posible acelerar la carga de la página principal reduciendo las peticiones HTTP y eliminando cuellos de botella.

2.3.2 – MEDIDAS PARA REDUCIR EL CONSUMO RED

Como se ha visto, los elementos de Youtube y Twitter conllevan una parte muy grande del problema al suponer casi dos tercios del peso de la página principal y alargar en 7 segundos la carga de la misma. En cuanto a los elementos descargados desde tel.uva.es, las imágenes son con diferencia el conjunto más alarmante ya que suponen casi un 75% del peso transferido desde dicho dominio. Además el número total de peticiones es muy elevado en comparación con las otras páginas.

Para mejorar todos estos y otros aspectos se tomaron las siguientes medidas:

- **Eliminar el video de Youtube:** eliminar el vídeo incrustado directamente en la página principal, y sustituirlo por una imagen y un enlace al video alojado en el portal Youtube.com. Si bien desde la publicación del nuevo portal ha mejorado la visibilidad de la Escuela en las redes sociales, el video de Youtube no ha recibido mucha atención por parte del público, consiguiendo menos de una visita diaria desde su lanzamiento. Disponer del video incrustado en la página web de la Escuela es vistoso y cómodo para su visualización, pero el enorme peso descargado solamente para mostrar el *widget* (en caso de reproducir el video, la descarga sería aun mayor) es un precio demasiado alto. Eliminar este objeto reducirá más de 1,5MB del total, y se espera que agilice la carga al ser uno de los elementos que la ralentizaba junto con la línea temporal de Twitter.
- **Compresión de imágenes:** El peso elevado de los archivos de imagen procedentes del dominio www.tel.uva.es suponen una fuerte carga, no solo para el cliente que debe descargarlas, sino también para el servidor de la Escuela que las debe enviar a todos los clientes. Se tomó la medida de realizar una compresión con pérdidas a todas las imágenes JPEG y PNG de la página. Una compresión con pérdidas implica una reducción en el tamaño de un archivo a cambio de una pérdida irreversible de información. “En el procesado de imagen, a menudo no es necesaria una reconstrucción fiel, siempre que la compresión de los datos no cause cambios significativos en la imagen” (Iowa, 2001). La mayoría de imágenes puede soportar una compresión significativa sin un deterioro visible en la calidad de la imagen. En este caso, se ha empleado la herramienta magikimg para realizar este proceso, empleando el siguiente comando, que reducía la calidad de la imagen al 75%:

```
convert -strip -interlace Plane -quality 75% input output
```

Código 5 - Comando utilizado para la compresión con pérdidas de archivos de imagen.

Aunque la calidad de las imágenes se fijara al 75%, ninguna de ellas sufrió un deterioro visible y su tamaño se redujo notablemente, en ocasiones a menos del 20% del original.

Hay que destacar que el ancho de banda dedicado a las imágenes se puede ver ampliado o reducido en función del estado del carrusel de la página principal, ya que el número de elementos en un determinado momento puede variar en función de los eventos a los que

se quiera dar visibilidad. Sin embargo, siempre será una buena práctica comprimir los archivos de imagen en la medida de lo posible sea cual sea la cantidad y el tamaño de las mismas.



Ilustración 41 - Ejemplo de compresión con pérdidas de imágenes. Izquierda: sin comprimir, 252kB. Derecha, compresión al 75% de calidad, 48kB.

- **Limitar el impacto generado por el *widget* de Twitter:** En la tabla de uso de red se puede ver que junto con el video de Youtube, el *widget* de Twitter supone uno de los objetos más pesados, con más de 1MB de datos descargados. La decisión de eliminar el video incrustado de Youtube se tomó teniendo en cuenta el poco éxito que había cosechado y, sobre todo, el hecho de que al sustituirlo por una imagen y un enlace no se estaba perdiendo información: solamente se obliga al usuario a cargar la página de Youtube si quisiera ver el vídeo, en lugar de hacerlo directamente en el portal de la Escuela.

El problema en caso de hacer algo similar con el *widget* de Twitter, es que su contenido es dinámico y cambia cada vez que se publica algún mensaje en el perfil de la Escuela. Además, a diferencia del video incrustado, este objeto proporciona información a simple vista. Sustituir la línea temporal por un enlace eliminaría completamente toda la información visible directamente en el portal. Con esto en cuenta, y vista la satisfactoria acogida del perfil de la Escuela en la red social de Twitter (con más de 600 seguidores), se intentó encontrar una solución que:

- o Redujese la cantidad de datos descargados por el *widget*.
- o Acelerarse la carga del objeto o, al menos, no interfiriese con la carga del resto de la página.

Para cumplir estos objetivos se abordaron dos medidas diferentes:

- o **Limitar la cantidad de contenido que el *widget* puede cargar.** El código para incrustar el *widget* está generado automáticamente por una herramienta de Twitter. La herramienta proporciona el código que se debe pegar en la página, de manera que cualquier persona sin grandes conocimientos de desarrollo web pueda utilizarlo, pero Twitter también puso a disposición de los desarrolladores una API con la que modificar o desarrollar sus propios *widgets*. Empleando esta API, se modificó el *widget* del portal para limitar el número de mensajes cargados a los 5 más recientes, de manera que para ver mensajes más antiguos el usuario

deba entrar en la página de Twitter. De esta manera se reduce la descarga de datos a cambio de sólo mostrar la información más reciente.

- **Ejecutar el código Javascript que carga el contenido del *widget* de forma asíncrona**, y una vez terminada la carga del resto de la página. Por lo general, el código Javascript es ejecutado de forma síncrona, es decir, el intérprete de Javascript del navegador avanza línea por línea, ejecutando las instrucciones o funciones de la línea completamente antes de avanzar a la siguiente. De esta manera se preserva la integridad de la memoria y naturaleza determinista del proceso en todo momento, con el posible problema de generar cuellos de botella, en los que trozos más complejos retrasan la ejecución del resto del código. Esto es especialmente problemático cuando se cargan contenidos de terceras partes, al depender de la conexión entre el cliente y el servidor de la tercera parte.

En ocasiones es posible que sea conveniente (o necesario) ejecutar procedimientos fuera del flujo normal del programa. Para ello, Javascript permite marcar *scripts* concretos para ser ejecutados asincrónicamente. El proceso para cargar *scripts* de terceros asincrónicamente consiste en crear un objeto del tipo *script*, dar a su atributo “*async*” el valor *true*, proporcionarle una *URL* desde la que cargar el *script*, y añadir el objeto a la lista de *scripts* del documento HTML.

El *widget* de Twitter es un ejemplo de estos cuellos de botella dentro de la página, ya que aúna las características de tener un *script* complejo, y dependiente de un servicio de terceros. Por esta razón se realizará su carga de forma asíncrona.

```
function load(url) {  
  script = document.createElement('script');  
  script.async = true;  
  script.src = url;  
  scripts.parentNode.insertBefore(script, scripts);  
}
```

Código 6 - Código Javascript para la carga asíncrona de *scripts*.

Se espera que estas medidas contribuyan a reducir notablemente el peso de la página principal, reducir el número de peticiones HTTP realizadas a contenido de terceros, y recortar el tiempo de espera desde que la página comienza a cargar hasta el instante en el que se puede usar. Después de este instante, el *widget* de Twitter continuará con su carga.

- **Carga de recursos desde CDN:** el portal de la Escuela hace uso de varias librerías y recursos de uso extendido en toda la red, concretamente jQuery en Javascript y Bootstrap y Normalizr en CSS, además de la tipografía OpenSans. Actualmente estos recursos están alojados en el servidor de la Escuela y se sirven con la página web. Una manera de optimizar el rendimiento consiste en servir estos recursos a través de una “Red de Distribución de Contenidos”, o CDN (concretamente, nosotros usaremos la CDN de Google, que sirve todos estos recursos). El beneficio de esta práctica reside en el uso

extendido de la misma ya que, al visitar cualquier página que también sirva el mismo recurso desde el mismo CDN, dicho recurso quedará almacenado en la caché del navegador. Al visitar después el portal de la Escuela, estos recursos ya se encontrarán almacenados en la caché y será innecesario descargarlos, eliminando peticiones HTTP y tráfico en el servidor y en el cliente.

La clave de esta práctica consiste en que los recursos utilizados son muy comunes en internet y la inmensa mayoría de usuarios ha visitado en algún momento páginas web que emplean Bootstrap y jQuery. Concretamente, un 26% de los portales web de internet emplean el servicio de CDN para librerías de Google, el mismo que usará el portal de la Escuela.

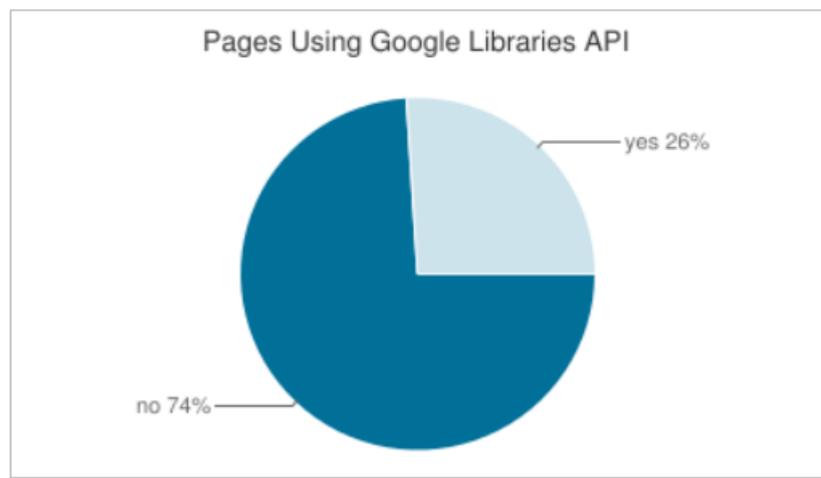


Ilustración 42 - Páginas que usan la API de librerías de Google. Fuente: <http://mobile.httparchive.org/interesting.php#googlelibs>

Con esta medida se ahorra una mayor cantidad de tráfico en el lado del servidor, que ya nunca tiene que enviar dichos archivos al ser servidos por terceras partes, que en el del cliente. Este último notará la mejora siempre y cuando haya visitado ya esta página o cualquiera de las que empleen este mismo sistema y librería.

- **Empaquetado, minificación, y compresión de archivos de texto:** Los archivos de texto plano, entre los que se encuentra el código HTML, CSS y Javascript, poseen un gran potencial para ser reducidos a través de las prácticas conocidas como minificación, compresión y empaquetado.

Empaquetado es un sencillo proceso que consiste en combinar todos los archivos de código de un tipo dentro de un único archivo. Concretamente, supondría fusionar todos los ficheros CSS dentro de uno solo por un lado, y todos los archivos Javascript en otro archivo por otro lado. Con esto no se ganarían muchos bytes al ser esencialmente el mismo texto, pero se reduciría la cantidad de peticiones HTTP realizadas, agilizando la carga de la página y reduciendo sobrecarga introducida por las cabeceras de los archivos. Al haberse trasladado la carga de las librerías a una CDN externa, solo queda un archivo de CSS y otro de Javascript por lo que no se puede aplicar esta práctica.

La **minificación**, específica de los archivos de texto y, sobre todo, para Javascript, consiste en reescribir el código de manera que las instrucciones sean exactamente las mismas, pero ocupe el menor número de caracteres, reduciendo así el número de bytes a transferir. Esto se consigue mediante multitud de técnicas como cambiar los nombres de las variables y las funciones, eliminar caracteres innecesarios como espacios y tabulaciones, o comentarios, etc. El proceso ofusca el código de manera que se vuelve ilegible para el observador humano, pero el navegador interpretará exactamente lo mismo que haría antes de la minificación.

Además la transformación es irreversible ya que no se podrán recuperar los caracteres eliminados ni los identificadores anteriores. Por esto, para poder continuar programando sobre estos archivos es necesario mantener una versión sin minificar empleada para desarrollo, y una versión minificada empleada únicamente en el servidor de producción.

Se minificarán todos los archivos .css y .js alojados y servidos desde www.tel.uva.es. Sería beneficioso minificar también los archivos .html pero, dado que estos archivos son generados en la compilación al iniciar el servidor Tomcat, la tarea es mucho más complicada.

```
var cars;
for (i = 0; i < cars.length; i++) {
    text += cars[i] + "<br>";
}

var c;for(i=0;i<c.length;i++){text+=c[i]+"<br>"}
```

Código 7 - Ejemplo de minificación de código Javascript. Arriba, código original, 77 caracteres. Abajo, código minificado, 48 caracteres.

Para minificar los archivos se emplearon dos herramientas de código abierto que se pueden instalar en un servidor y ejecutar localmente:

- Uglify-js para los archivos de Javascript
- Clean-css para los archivos CSS

```
uglifyjs input.js > output.min.js
cleancss -o output-min.css input.css
```

Código 8 - Comandos para minificar archivos. Arriba: minificación Javascript con *uglify*. Abajo, minificación CSS con *cleancss*.

Por último, la **compresión** consiste en reducir el tamaño de un archivo (no necesariamente de texto) realizando operaciones a nivel de bit, en lugar de a nivel de carácter, como se hacía en la minificación. A diferencia de la minificación, un navegador no es capaz de interpretar directamente el resultado de la compresión. Por fortuna este

proceso es reversible, de manera que el cliente puede descomprimir los archivos para trabajar con ellos. El resultado es una reducción en el tráfico entre servidor y cliente a cambio de carga en el procesador para comprimir/descomprimir el archivo.

El formato empleado para la compresión de archivos para HTML es GZIP. Habilitaremos la compresión en el servidor, fijando la variable "compression" al valor "on" en el archivo de configuración de Tomcat.

```
<Connector
  port="8080"

  disableUploadTimeout="true"
  compression="on"
  compressionMinSize="2048"
  compressableMimeType="text/html,text/xml,text/plain,text/css,application/javascript"
/>
```

Ilustración 43 - Fragmento del objeto "Connector" de la configuración de Tomcat que controla la compresión GZIP. La línea "compression="on" activa la compresión GZIP

- **Creación de hojas de *sprites* de CSS:** Una hoja de *sprites* de CSS es un archivo de imagen que contiene varias imágenes pequeñas que se usan de manera independiente dentro de una página. La razón para unir varias imágenes pequeñas en un gran archivo es el ahorro en cuanto a peticiones HTTP, al pasar de muchas peticiones individuales a una única. Tener muchas imágenes pequeñas puede producir un gran retraso a la hora de cargar una página ya que las peticiones HTTP requieren una negociación, y además los navegadores suelen limitar el número de peticiones concurrentes.

Una vez combinadas las imágenes en un solo archivo, la manera de cargarlas dentro de la página web varía. En lugar de introducir la ruta de la imagen dentro de un elemento de imagen en el código HTML, se añade una clase CSS a dicho elemento, y se establece la imagen como imagen de fondo de dicha clase CSS. El elemento tiene un tamaño que solo permite ver una pequeña parte de su imagen de fondo, concretamente el tamaño de la imagen que queremos ver. Variando la posición de la imagen de fondo de la clase, se puede visualizar cualquiera de las imágenes más pequeñas que conforman el archivo.

Aunque cada elemento que emplea *sprites* tiene toda la imagen como imagen de fondo, la manera en la que actúan los navegadores modernos hace que solo se rendericen los píxeles que se están visualizando, por lo que no se genera carga adicional a la hora de renderizar.

Se ha generado una hoja de *sprites* para todas las imágenes del portal menores de 5KB, reduciendo el número de peticiones sin aumentar apenas el peso de la página.

- **Eliminación de código CSS y Javascript incrustado o *inline*:** El código incrustado o *inline* es el que se encuentra dentro de los archivos .html, mezclado con código HTML. Aunque es posible en algunas ocasiones reducir el número de peticiones HTTP realizadas incrustando código, esto facilita la tarea de *cachear* los ficheros al navegador, reduciendo el tráfico generado en subsiguientes visitas, especialmente en portales con archivos de CSS grandes

(el nuestro lo tiene, ya que concentra todo el código CSS propio en un solo archivo) y gran cantidad de páginas.

Las páginas del portal de la Escuela han acumulado con el paso del tiempo gran cantidad de trozos de código CSS y Javascript incrustado. Se ha llevado a cabo la labor de analizar página por página, trasladando en la medida de lo posible todos estos trozos de código a los archivos .css y .js principales. Aunque no se apreciará un efecto inmediato en el ancho de banda consumido ni en el tiempo de carga, los beneficios serán aparentes en posteriores visitas al disponer de más código CSS y Javascript *cacheado*.

Tras la implementación de las medidas planteadas para reducir el peso de las descargas de la página principal, se repitieron los análisis de uso de red realizados en la sección 2.3.1. Se espera una reducción apreciable en el ancho de banda y los tiempos de carga de la página principal.



43 peticiones, 745,72 KB (transferido: 567,03 KB), 4,12 s

Ilustración 44 - Resumen de los resultados de tras la optimización del ancho de banda.

El volumen total de descargas ha bajado a menos de 1MB en total, si bien es necesario tener en cuenta que esto es en el caso ideal, en el que el navegador del usuario admite compresión GZIP y ya ha visitado otras páginas que sirven las mismas librerías desde CDN. Por otra parte, el tiempo de carga también se ha reducido a 4,12 segundos, menos del 50% de respecto a los primeros resultados.

2.3.3 – CONSUMO ENERGÉTICO

Como ya se ha explicado anteriormente, en el ámbito de la web coexiste un amplio conjunto de sistemas, desde los servidores y los centros de datos, hasta los dispositivos personales desde los que se navega, como ordenadores o teléfonos móviles, pasando por elementos de transporte como las redes de cable, fibra óptica, o redes inalámbricas. Todos estos elementos, sin excepción, consumen electricidad, y en cada uno de ellos entra en juego multitud de factores condicionantes, sobre muchos de los cuales es imposible para un desarrollador tener influencia.

De la mano del desarrollador web cae, si nos ceñimos expresamente a las responsabilidades directas de dicho puesto de trabajo, el consumo de energía generado por el navegador al visualizar su página web. Además de esto, tiene influencia sobre el consumo del servidor, en tanto que la página web que desarrolla consume recursos al ser servida en un entorno de producción. También genera un gasto energético el uso de las redes de transmisión empleadas para transferir dicha página desde el servidor hasta los clientes, factor acrecentado a medida que lo hace el peso de la página.

Las secciones anteriores, especialmente la 2.3.2 – “Medidas para reducir el consumo de red” han abordado temas relacionados con la optimización de recursos en la transferencia de la página, lo cual redundará en un ahorro energético en el servidor, el medio de transmisión, y el cliente. Sin embargo, aún no se ha hecho hincapié en las medidas necesarias para reducir el consumo derivado de la *visualización* (en contraposición a la transferencia, sobre la que sí se ha trabajado) de las páginas del portal.

En esta sección se tiene por objetivo estudiar los principales focos de consumo de energía eléctrica de la visualización de una página web en un navegador. Se abordarán dichos focos con la intención de optimizar el portal, y se analizará la eficacia de dichas medidas, en cuanto a los resultados energéticos obtenidos.

2.3.3.1 – ANÁLISIS DEL CONSUMO ENERGÉTICO DEL PORTAL

El primer paso en esta fase consistió en tomar mediciones acerca del consumo de recursos hardware por parte de la página web. Para ello se empleó la herramienta “Análisis de Rendimiento”, incluida dentro del paquete de herramientas para desarrolladores web del navegador Chrome. Esta herramienta graba la utilización de recursos durante un periodo determinado y muestra un análisis detallado de cada una de las áreas medidas.

En esta ventana se muestran datos acerca de:

- Uso del procesador gráfico (GPU) en forma de Frames por Segundo (FPS) e interacciones.
- Uso del procesador central (CPU) en forma de ciclos y árbol de llamadas Javascript.
- Uso de memoria (RAM) en cuanto a memoria asignada y Heap (Memoria disponible para asignar)
- Uso de red en cuanto a volumen y línea temporal de descargas.

De los datos anteriores, los más interesantes para el presente análisis serán el uso de GPU y, especialmente, el uso de CPU, por ser los elementos de hardware que más gasto de energía eléctrica suponen. Como comparación, un módulo de memoria RAM DDR3 consume menos de 10W de potencia (JEDEC, 2008), frente a los varios cientos de los chips CPU y GPU más modernos (RealHardTechX, 2017). Además el estado de la memoria (en cuanto a estar sometida a mayor o menor uso) tiene una influencia menor en el consumo de potencia. Por estas razones se obviarán las medidas relacionadas con la memoria RAM. El uso de red ya fue optimizado en el capítulo 2.3.2 y por lo tanto se excluirá también del análisis.

El consumo de energía en una CPU o GPU está estrechamente relacionado con el uso de ciclos de procesador, es decir, el tiempo empleado activamente en procesar instrucciones, en contraposición al tiempo que el procesador permanece en estado de reposo, o esperando a que se completen las operaciones de lectura/escritura. La energía consumida por el procesador es directamente proporcional al uso de ciclos, de manera que se buscará la manera de optimizar estas cifras en la mayor medida posible.

Para realizar el análisis del portal se tomarán mediciones de dos intervalos de tiempo separados. El primer intervalo será el de la carga inicial de la página. Este se considera desde el momento en el que se realiza la petición inicial HTTP desde el cliente, hasta el momento en el que se han cargado y renderizado todos los elementos que forman la página y esta está lista para usarse. El segundo intervalo consistirá en el estado en reposo de la página, es decir, desde el final de la carga en adelante, el tiempo en el que el usuario está visualizando propiamente la página web.

El primero de estos intervalos es por definición más corto e intensivo que el segundo, al comprender la carga de todos los elementos y servirá para analizar el uso de ciclos de procesador en los momentos iniciales. El segundo intervalo servirá para analizar el uso de ciclos durante la lectura de la página, un periodo de tiempo de menor carga pero de duración mucho mayor.

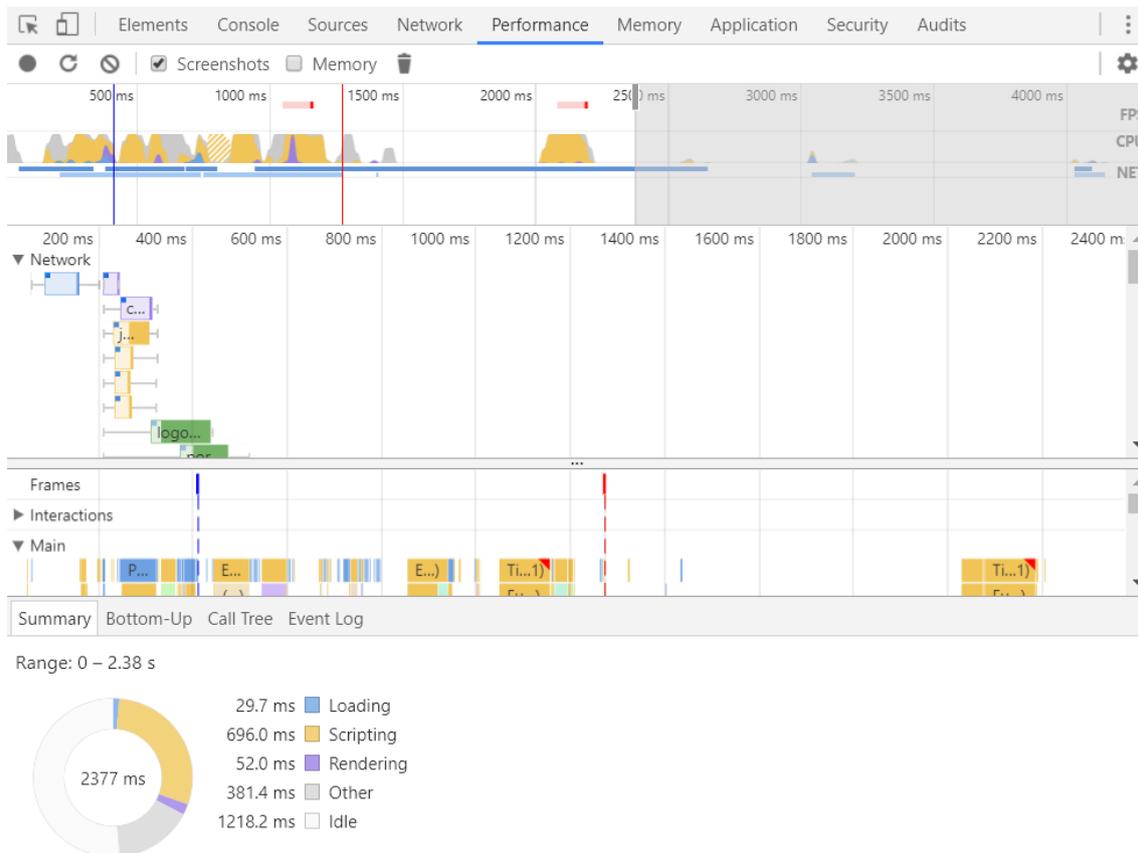


Ilustración 45 - Resultados del test de rendimiento para tel.uva.es (Carga)

Tarea	Tiempo (milisegundos)
Carga (Loading)	29,7
Javascript (Scripting)	696,0
Renderizado (Rendering)	52,0
Espera (Idle)	1218,2
Otros (Other)	381,4

Tabla 10- Desglose de tiempo de procesador por tareas para tel.uva.es (Carga)

La herramienta de Análisis de Rendimiento de Chrome contabilizó automáticamente los primeros 2377 milisegundos desde que se efectuó la primera petición desde el cliente como periodo de carga de la página. Durante este periodo, aproximadamente el 50% (1158,8ms) de ciclos de procesador disponibles fueron empleados para cargar la página. La mayor parte de estos (696,0ms) fueron empleados para procesar scripts de Javascript, mientras que la carga del árbol HTML y el renderizado tuvieron un impacto muy pequeño (29,7ms y 52ms respectivamente). El procesador se mantuvo en espera 1218,2ms, de los cuales la mayoría se corresponden con el tiempo de recepción de datos a través de internet, ya que las operaciones de lectura/escritura durante la carga de esta página web concreta son prácticamente nulas. Para concluir, 381,4ms fueron empleados en la categoría “otros”.

De los resultados se desprende que el mayor foco de utilización de ciclos de procesador es el código de Javascript. La categoría “otros” comprende tareas no categorizables sobre las que no se puede profundizar en su análisis, y las categorías de renderizado y carga suponen un efecto despreciable. Es por tanto que se centrarán los esfuerzos en optimizar el código javascript

ejecutado durante la carga de la página principal, para mejorar los resultados en cuanto al uso de ciclos del procesador.

Para optimizar un script, independientemente de su lenguaje, el primer paso es definir los objetivos. En este caso, la optimización pretende reducir el consumo de recursos por parte de dicho código en forma de ciclos de procesador y uso de memoria. Para conseguir dicho objetivo, se decidió abordar el código fuente desde dos frentes:

- **Macroscópico:** Consiste en analizar el funcionamiento del código a gran escala: ¿Son eficientes los algoritmos tomados para realizar una determinada tarea? ¿Se podría rediseñar una función para que completase el mismo trabajo en menos tiempo y con menos recursos? ¿Se está malgastando memoria o instrucciones en código muy convolucionado?

Responder este tipo de cuestiones y solucionarlas no suele requerir un dominio profundo del lenguaje, pero sí un buen entendimiento del código, de la función que realiza y de su estructura. En este caso se trabajó principalmente en mejorar el código que maneja el carrusel de imágenes, por ser el script más complejo de la página (exceptuando el código servido por terceros, como el del *widget* de Twitter).

- **Microscópico:** Independientemente del algoritmo diseñado para abordar un problema, y de su calidad, un código siempre es susceptible de ser inspeccionado y mejorado a pequeña escala. Si el aspecto macroscópico de la optimización se basa en determinar si se han tomado la serie de pasos correctos para resolver un problema, el aspecto microscópico trata de determinar si dichos pasos han sido implementados de manera eficiente. Para ello es más necesario un conocimiento del lenguaje empleado, que del script que se está optimizando, pues este se divide en pequeñas unidades independientes formadas por pocas o incluso una instrucción única.

El desarrollador de Javascript necesita conocer bien el funcionamiento del lenguaje y de cómo son tratadas internamente funciones, las instrucciones, las variables, el espacio de nombres... de tal manera que pueda detectar implementaciones poco eficientes del lenguaje en el código, y optimizarlas. A continuación se muestran algunos de los ejemplos de optimización a nivel microscópico implementados en el código del portal de la Escuela:

- o Almacenamiento de variables en bucles: La variable de control de un bucle se emplea en cada iteración para comprobar si se debe continuar dentro de dicho bucle. En los casos en los que su acceso no sea directo, por encontrarse en otro espacio de nombres, u obtenerse a través de una función, es más eficiente guardar su valor en una nueva variable local. Esto evita repetir el acceso a otros niveles de la pila de llamadas en cada iteración.
- o Uso correcto de operadores: Algunos operadores matemáticos actúan de manera diferente si son colocados antes o después de la variable sobre la que operan. Es el caso, por ejemplo, del operador ++. En ambos casos suma 1 al valor de la variable a la que acompaña. Sin embargo, si se coloca antes de la variable, se realiza la suma antes que el resto de operaciones de la instrucción. Si se coloca después, se realizará la suma al terminar el resto de operaciones y comprobaciones. Este segundo caso requiere más instrucciones de bajo nivel, ya que almacena una copia de la variable antes de incrementarla, que será usada

para la operación. El resultado es que “var++” requiere más ciclos de procesador que “++var”. En ocasiones en las que sea indiferente el orden de las operaciones, es recomendable usar por tanto la segunda opción, a pesar de que la primera es la más vista generalmente.

Un ahorro de varios ciclos de procesador puede resultar despreciable a simple vista, pero si dicho ahorro se da dentro de un bucle con gran cantidad de iteraciones, se puede conseguir una mejora significativa.

A continuación se repitieron las medidas para el segundo intervalo temporal: desde el fin de la carga de la página en adelante, para analizar el consumo en reposo. En total se grabó el rendimiento de la página durante 180 segundos, el tiempo máximo que permitía la herramienta de Chrome.

En la gráfica de rendimiento a lo largo del tiempo, los frames por segundo (FPS) se muestran en la parte superior en verde. Debajo de la gráfica FPS se muestra la gráfica de uso de ciclos de procesador, y más abajo se encuentra el árbol de llamadas de Javascript, en la gráfica con la etiqueta “Main”.

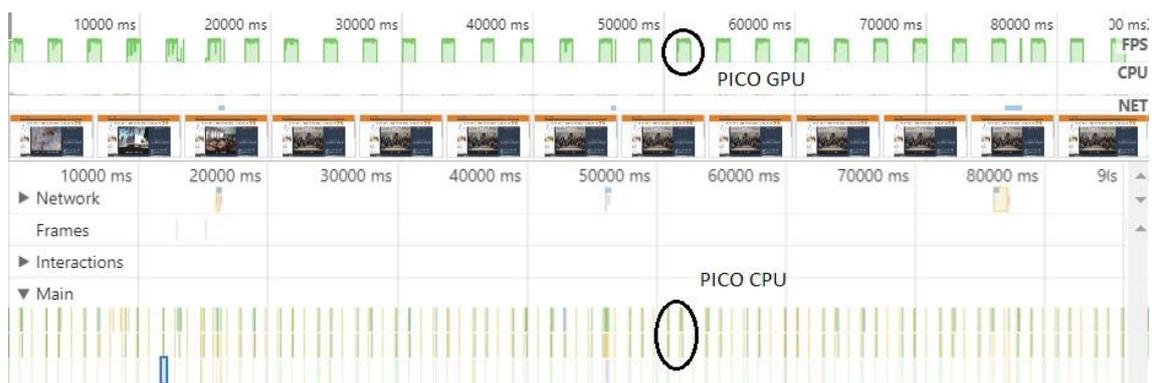


Ilustración 46 - Gráficas de uso de GPU y CPU a lo largo del tiempo. Se han recortado los resultados hasta la mitad del intervalo (90s) por falta de relevancia.

En los resultados obtenidos se aprecia un uso de ciclos muy bajo en la gráfica de CPU, con multitud de picos periódicos de uso de GPU (visto en la gráfica de FPS) y CPU (en el árbol de llamadas). Estos picos están sincronizados temporalmente, y se corresponden con los cambios de imagen en el carrusel de imágenes.

Se aprecia como los picos en el árbol de llamadas aparecen por pares. Una inspección detallada de los nodos del árbol encontrados en dichos picos, revela que los ciclos se emplean para las siguientes tareas:

- En el primer pico de cada par: disparar el evento del temporizador que provoca el cambio de imagen, recalcular los elementos HTML, actualizar el árbol HTML y repintar la página.

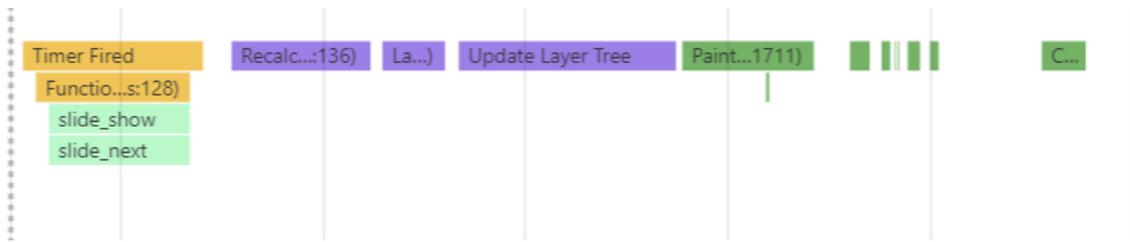


Ilustración 47 - Árbol de llamadas del primer pico de cada par.

- En el segundo pico de cada par se repiten las tareas del primero, sin aparecer el evento del temporizador. Se recalculan los elementos HTML y se repintan los píxeles de la página.



Ilustración 48 - Árbol de llamadas del segundo pico de cada par.

Además de las llamadas observadas, entre cada uno de estos pares picos se dispara el uso de la CPU al aumentar los frames por segundo. Esto es debido a la animación que se produce al cambiar la imagen del carrusel, con la imagen anterior difuminándose hasta desaparecer, y la imagen nueva apareciendo gradualmente.

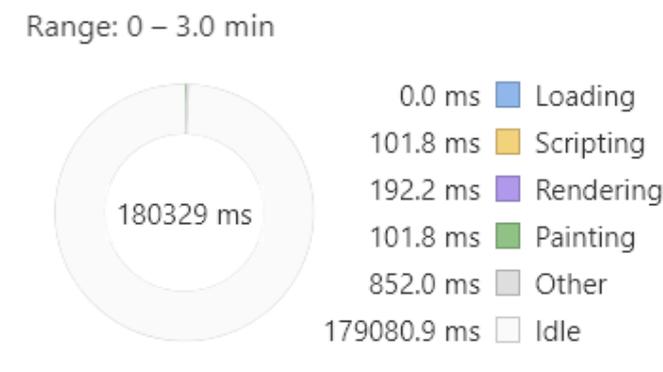


Ilustración 49 - Distribución del uso de ciclos de CPU y GPU en estado de reposo.

A pesar de que cada uno de estos ciclos no está generando una gran carga individual, hay que observar que estos se repiten periódicamente mientras la página está en reposo. En total, durante los 180 segundos de análisis se acumularon 495,8ms de uso de procesador en tareas de ejecución de scripts, renderizado y repintado, una cantidad que no es despreciable si se compara con los datos obtenidos para la fase de carga de la página.

En su presentación en el Foro de Desarrolladores de Intel de 2010 acerca del impacto en la batería de los programas de software en reposo, Manuj Sabharwal y Eunice Chang llegaron a la conclusión de que los temporizadores empleados por software en reposo (y las interrupciones y llamadas al sistema derivadas de su uso) producían un gasto energético muy significativo (Sabharwal, 2010).

Por otra parte, la guía para desarrolladores web de Google detalla cómo el repintado de los píxeles de una página, y el recalculado de los elementos HTML y su estilo conllevan una gran carga para el procesador (Lewis, 2017).

Estas dos áreas mencionadas son las principales culpables del gasto energético de la página web de la Escuela una vez finalizada su carga. Una solución directa y que implicara resultados significativos giraría alrededor del tiempo entre cambios de imágenes del carrusel de la página.

Se tomó la decisión de modificar el código Javascript del carrusel para que las imágenes no roten periódicamente en el carrusel de la página principal en dispositivos móviles. En lugar de ello, se mostrará la imagen de mayor importancia (especificada en el código HTML), y el usuario podrá navegar entre las demás imágenes mediante los botones de “anterior” y “siguiente” incluidos en el carrusel.

El fin de esta medida es ahorrar batería en los dispositivos móviles a la vez que se mantiene parte de la funcionalidad, y mantener el 100% de funcionalidad en ordenadores de sobremesa, los cuales no dependen de una batería.

Tras las optimizaciones implementadas, se repitieron las tomas de datos de rendimiento en la página principal. Se observó una importante mejora en el consumo de ciclos de procesador, especialmente los relacionados con Javascript, descendiendo de los 696ms anteriores a 312ms, menos del 50%.

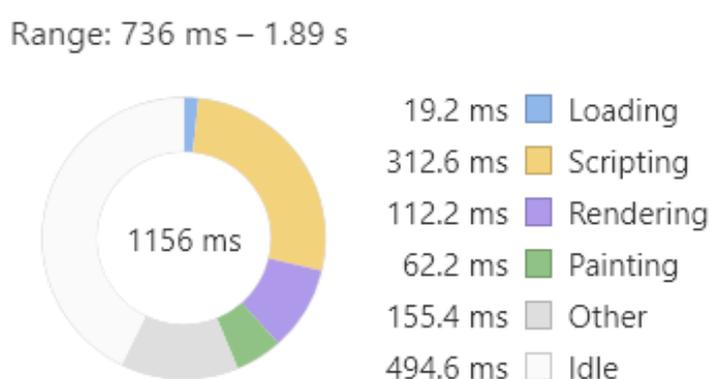


Ilustración 50 - Distribución del uso de ciclos de CPU y GPU durante la carga, tras la optimización.

Además, el tiempo en espera o “Idle” también mejoró significativamente. Aunque esto no represente un descenso en el consumo energético, sí indica una carga más rápida, gracias al mayor aprovechamiento de ciclos de procesador respecto a ciclos en espera.

En cuanto a los resultados en estado de reposo, el mayor efecto se observó en los dispositivos móviles gracias al cambio en el carrusel de imágenes. Al no emplear temporizadores ni repintados, el uso de ciclos de procesador en este periodo descendió enormemente en todas las categorías.

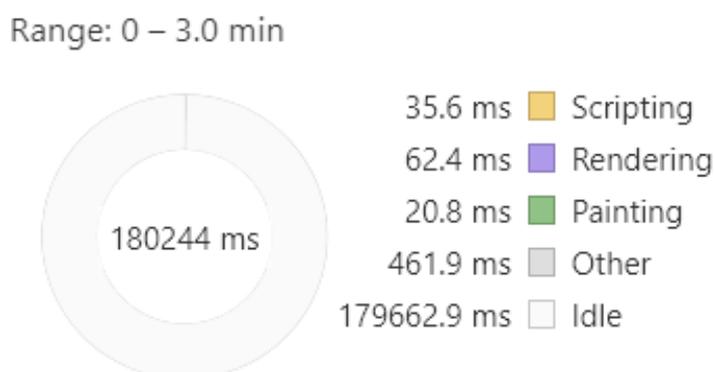


Ilustración 51 - Distribución del uso de ciclos de CPU y GPU en estado de reposo tras la optimización

Para ganar una visión general del rendimiento energético del portal después de las medidas adoptadas, se recurrió a una herramienta on-line gratuita desarrollada por el consorcio francés Green Code Lab (formado por las empresas Inria, UmanIT, Greenspector y Ademe), del cual se ha hablado anteriormente por promover y concienciar sobre la Computación Verde en todo el mundo y organizar el concurso Green Code Lab Challenge.

La herramienta se llama Web Energy Archive, es libre para uso de cualquier persona y se encuentra en la URL <http://www.webenergyarchive.com>. Los usuarios pueden suscribir sus páginas web a la base de datos de la herramienta. Periódicamente, la herramienta tomará muestras de los principales marcadores energéticos y elaborará un reporte disponible para todo el público e individual para cada página.

Los detalles para la página de la Escuela se pueden encontrar en la URL <http://www.webenergyarchive.com/fr/websites/view/tel.uva.es>. Como resumen, Web Energy Archive muestra una gráfica similar a las etiquetas de eficiencia eléctrica encontradas en los electrodomésticos, en la que se emite una valoración global de la página en cuanto a gasto energético y consumo de recursos.

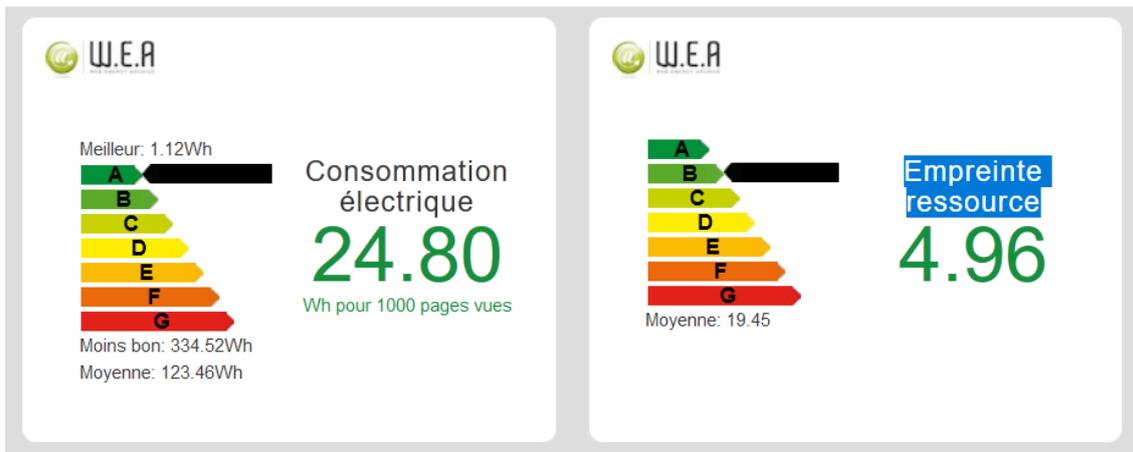


Ilustración 52 - Resultados de eficiencia energética de tel.uva.es en Web Energy Archive.

Se puede observar como la página de la Escuela obtiene valoraciones muy altas, entrando en la mejor categoría posible (A) en consumo eléctrico con 24,80W/h por 1000 visualizaciones, y en la segunda mejor en cuanto al uso de recursos (B) (4,96 en el baremo de 0 a 50, siendo mejor resultado cuanto más baja sea la cifra). Como comparación, la media de las páginas analizadas por Web Energy Archive es mucho peor que los resultados obtenidos por el portal de la Escuela: 123,46Wh por 1000 visualizaciones y 19,45 en uso de recursos).

A la vista de los resultados obtenidos, se considera que el estado del portal en cuanto a su consumo energético es altamente satisfactorio, y se da por concluido el trabajo.

3: CONCLUSIONES Y DISCUSIÓN

Concluido el trabajo, es momento de realizar una breve retrospectiva de los puntos clave desarrollados en materia de desarrollo web y programación eficiente o “verde”, y hacer una valoración general de los resultados obtenidos.

- **Hace falta un cambio de mentalidad hacia una computación más sostenible y ética.** La Computación Verde surge de la necesidad de concienciar a la comunidad de las Tecnologías de la Información (IT) sobre un problema que, a menudo, puede pasar desapercibido: Los ordenadores, los servidores, los centros de datos... todos los sistemas que forman parte del complejo ecosistema de la computación y de internet suponen un enorme impacto en el medio ambiente.
- **El éxito de un portal web pasa, en gran medida, por conocer al usuario.** Se ha visto, a través de varios estudios, cómo los hábitos de navegación de la población dan forma a las páginas web que visitamos cada día.
 - o La colocación de los elementos dentro del conjunto de la página no es aleatoria, sino que responde a una predisposición por parte del visitante a buscar ciertos objetos en lugares concretos.
 - o La página principal y, fundamentalmente, la parte de la página principal visible sin necesidad de hacer *scroll*, es un punto vital del diseño: casi todos los ojos que visitan un portal pasan por esa sección.
 - o Sabiendo qué buscan los usuarios es posible colocar accesos directos, fácilmente visibles, para atraer la atención y facilitar la navegación.
- **Los usuarios de dispositivos móviles no pueden ser ignorados.** Se ha superado la barrera tras la cual los usuarios de internet pasan más horas navegando con su teléfono móvil que con su ordenador. Dar al visitante una experiencia agradable, manejable, adaptada a las necesidades de su dispositivo, es un paso de gigante sobre los competidores que no implementan diseños *mobile-friendly*. Más que nunca gracias al buscador Google y a su política de promocionar webs adaptadas para móviles, es imposible ignorar esta situación.
- **El empleo de contenido de terceros requiere especial atención.** Cuando se desarrolla una página web, incluir contenido servido por terceras partes puede ser un gran añadido. Sin embargo, el control que se tiene sobre dicho contenido suele ser muy bajo, estando muchas veces confinado a elementos *<iframe>* cargados mediante APIs limitadas. Esta falta de control puede llevar a descuidar el impacto de estos contenidos. *Widgets* como los de Twitter o Youtube conllevan una descarga de datos elevada e incluso código Javascript complejo, que de no ser vigilado puede tener un gran impacto en el peso y el tiempo de carga de una página web
- **Las imágenes y el Javascript son los grandes lastres de una página web.** Los archivos de imagen suponen, con diferencia, el mayor volumen de datos descargados al cargar una página. “Una imagen vale más que mil palabras”: los archivos de imagen ayudan a ilustrar

el contenido, añaden color, y atraen la atención del visitante, pero su uso redundante en la descarga de datos y el gasto de batería en dispositivos móviles. Técnicas como la compresión con pérdidas pueden aliviar esta carga, pero siempre existirá el peligro de usar demasiadas imágenes. El código Javascript, por su parte, tiene un gran impacto en el uso del procesador y por consiguiente en el consumo de energía y los tiempos de carga. Es fundamental, a la hora de diseñar y desarrollar una página, hacerlo teniendo en cuenta las necesidades reales del proyecto y no excediéndose en el uso de imágenes y Javascript, muchas veces por cuestiones estéticas.

El nuevo portal de la Escuela ha sido creado cumpliendo los objetivos marcados y teniendo siempre en cuenta los factores mencionados. Pese a la limitación de trabajar principalmente sobre el *front-end*, se ha conseguido una web que cumple con múltiples controles de calidad, como el test *mobile-friendly* de Google o el de consumo energético de Green Code Lab en Web Energy Archive. El resultado del Trabajo es un portal más moderno, más manejable, más ético, adaptado a todo tipo de dispositivos y optimizado en cuanto a su carga y su consumo energético.

4: TRABAJO FUTURO

El trabajo realizado en este Trabajo de Fin de Grado deja las puertas abiertas a futuras ampliaciones en los campos del desarrollo web y la computación verde sobre el portal de la Escuela. A continuación se presentan algunas de dichas ampliaciones, proponiendo a la Escuela que en un futuro lleve a cabo algunas de estas medidas para mejorar el portal.

- **Estudio y optimización energética del back-end:** Aunque se han conseguido excelentes resultados en la optimización energética del nuevo portal en el lado del cliente (front-end), apenas se ha trabajado en el área del servidor (back-end). Esto ha sido debido a la gran cantidad de carga trabajo que ya conllevaba el proceso de remodelación del portal sumado a la implementación de las medidas de computación verde, y no a la imposibilidad de realizar un trabajo similar en el back-end.

La arquitectura interna del portal fue construida, como proyecto de Fin de Carrera, hace XXXX años, y desde entonces se ha mantenido prácticamente intacta. Se propone como trabajo futuro, quizás como Trabajo de Fin de Grado de un alumno de la Escuela, el análisis de dicha arquitectura y adaptación de medidas de eficiencia energética que optimicen el funcionamiento del servidor, convirtiendo al portal en una página 100% Verde.

El servidor Apache Tomcat recibe constantemente nuevas funcionalidades y actualizaciones por parte de la fundación Apache y, por ser de código abierto, de los miles de colaboradores en todo el mundo. Si bien el analista de red de la Escuela realiza un gran trabajo de mantenimiento y administración del servidor y el portal en general, sería beneficioso un estudio del estado del arte del desarrollo de back-end empleando Tomcat y, si procediera, una renovación de la arquitectura del servidor.

- **Integración de la página principal dentro del CMS interno del portal:** Como se comentó en el capítulo 1, el portal web de la Escuela dispone de un Sistema Gestor de Contenidos a través del cual, trabajadores de la Escuela pueden crear y gestionar páginas sin necesidad de conocimientos de desarrollo web ni del funcionamiento específico del portal de la Escuela. Se ha hecho un esfuerzo durante la remodelación del portal y, especialmente, la página principal, para que la modificación de los módulos de los que se compone la misma sea lo más sencilla posible. Añadir o modificar elementos del carrusel de imágenes o de las noticias de interés es, gracias a esto, una tarea relativamente sencilla. Sin embargo, para realizar estos cambios es necesario tener acceso al código fuente de la página, así como a subir imágenes al servidor. Esto no está a disposición de la mayoría de profesores de la Escuela, por lo que dependen del analista de red para llevar a cabo estas tareas.

Se propone la integración de estos módulos dentro del CMS interno, para que interactúen de la misma manera que con el tablón de anuncios. Este tablón es accesible desde el CMS, y su contenido se ve reflejado en la página principal al compilarse esta con los avisos extraídos de la Base de Datos. Sería ideal que el personal con acceso al CMS pudiera añadir nuevas entradas al carrusel de imágenes y a los enlaces de interés, de manera que la página principal se actualizara automáticamente con estos contenidos al ser compilada durante el arranque.

Para llevar a cabo esta tarea, sería necesario:

- Añadir la funcionalidad al CMS para subir imágenes (con unas características de resolución y peso limitadas para evitar que se dispare el peso de la página) a las categorías de carrusel y enlaces de interés, con los títulos y textos que correspondan.
- Modificar la plantilla de la página principal de portal, para realizar la carga de contenido del carrusel y los enlaces de interés de forma dinámica en lugar de mantenerlos incrustados en el código HTML. La carga se podría realizar de dos maneras:
 - o En tiempo de compilación, mediante código Java que lea las entradas correspondientes en la Base de Datos. Tiene la ventaja de la disponibilidad inmediata de dicho contenido en el lado del cliente, al servirse el documento HTML con los elementos ya incrustados en el código.
 - o De forma asíncrona empleando AJAX. Se pide mediante Javascript el documento que contiene las URLs y los textos de cada elementos, y se añaden programáticamente los objetos HTML a la página. Tiene la ventaja de realizar la descarga de imágenes una vez la página está cargada, de manera que los textos y los enlaces son accesibles antes de comenzar la descarga de los elementos más pesados.

Esta medida contribuiría a la creación de un portal más dinámico, en el que el contenido es gestionado de manera colaborativa y se elimina el cuello de botella existente, al tener que pasar todo este contenido por una sola persona.

- **Automatización de procesos para optimizar el consumo de red:** Durante la fase de optimización de consumo de red (Capítulo 2.3.2), se llevaron a cabo varias medidas que contribuirían a reducir la descarga de datos de las páginas del portal. Algunas de estas medidas, como la compresión GZIP de los archivos transferidos, es permanente. El problema aparece con algunas de estas medidas que son, por su naturaleza, percederas, en la medida en que se modifican los recursos sobre los que se han aplicado. Como ejemplo, las imágenes con las que más ahorro de datos se consiguió durante la compresión fueron las del carrusel de imágenes. Esta mejora se irá perdiendo a medida que el contenido de dicho carrusel se vaya actualizando, y las imágenes comprimidas sean sustituidas por otras nuevas, no comprimidas. Se proponen una serie de medidas para automatizar el proceso de optimización y evitar que el trabajo realizado se deteriore con el tiempo:

- Compresión automática de todas las imágenes subidas al servidor. Esto es especialmente importante si se llevara a cabo la integración del carrusel y los enlaces de interés en el CMS, ya que no se tiene control sobre los archivos subidos por el personal.

- Empaquetado y minificación automática de archivos CSS y Javascript. Los archivos Javascript y CSS empleados para el desarrollo no pueden estar minificados, ya que no serían legibles. Además, se presupone que en el futuro se añadirán nuevas funcionalidades, librerías y estilos que aumenten la cantidad de archivos CSS y Javascript. Aunque el administrador de la

página web puede realizar el empaquetado y minificación de dichos archivos cada vez que exista algún cambio, se recomienda que este proceso se lleve a cabo de manera automatizada, generando nuevos archivos de producción cada vez que haya modificaciones en los archivos de desarrollo.

La implantación de estas tareas se podría llevar a cabo con herramientas existentes de automatización de tareas para desarrollo web. Concretamente, la herramienta Grunt permite hacer estas y más tareas mediante Javascript ejecutable de forma local gracias al entorno de Javascript Node.js (<https://gruntjs.com>). Grunt recibe una lista de tareas a realizar de forma automática ejecutando scripts de Javascript de manera local a través de Node.js. Existen multitud de scripts generados por la comunidad y, debido a su carácter colaborativo, cualquier persona puede crear y distribuir nuevos scripts para automatizar tareas.

Se recomienda adoptar Grunt para automatizar la compresión de imágenes y el empaquetado y minificación de los archivos CSS y JS, de manera que estas tareas sean realizadas periódicamente de manera automática.

Adoptar estas medidas de automatización supondría la perduración de los beneficios conseguidos durante este trabajo durante mucho más tiempo, reduciendo la carga de trabajo requerida para su mantenimiento.

- **Recursos multimedia diferenciados para pantallas pequeñas:** En su estado actual, el servidor envía el mismo documento HTML para todos los dispositivos, independientemente de su tamaño de pantalla. Esto es un efecto directo de la adopción del Diseño Web Adaptativo (Responsive), frente a corrientes anteriores en las que se mantenían y servían documentos diferentes para cada tamaño de pantalla esperado. Las ventajas de este sistema ya se han enumerado en capítulos anteriores y, sin embargo, presenta ciertas desventajas. Una de ellas surge al servir los mismos archivos a todos los dispositivos, y cambiar únicamente la manera en que se renderizará su contenido a través de CSS: las imágenes enviadas a pantallas pequeñas tienen el mismo peso y resolución que las enviadas a un ordenador de escritorio. Esto, en muchas ocasiones, es un gasto de recursos excesivo e innecesario, ya que las pantallas pequeñas no representarán fielmente las imágenes con resolución mayor.

Se podría solventar este problema disponiendo de varias copias de cada una de las imágenes incluidas en el portal, adaptadas a distintas resoluciones (por ejemplo, adaptadas a los cuatro tamaños de pantalla soportados por defecto por Bootstrap). El documento HTML enviado incluiría URLs apuntando a las imágenes con la resolución adecuada. Esta tarea supondría:

- Modificar el código del servidor para que detecte el tipo de dispositivo desde que se realizó la petición y envíe un archivo de imagen diferente en función de dicho dispositivo.
- Mantener todos los archivos de imagen necesarios para cada resolución soportada e imagen incluida en el portal, incluyendo la creación de nuevos archivos a diferentes resoluciones cada vez que se añada una imagen nueva.

La tarea de mantener los archivos de imagen a diferentes resoluciones puede ser facilitada si se aplica la automatización de tareas con Grunt propuesta anteriormente. Sería necesario programar el script que, periódicamente, analizara las imágenes incluidas en las páginas del portal, buscara las copias de dichas imágenes para las diferentes resoluciones, y generase automáticamente las que no encontrara durante su análisis.

Una alternativa para la primera tarea, consistente en añadir la funcionalidad al código del servidor para servir diferentes archivos de imagen en función del dispositivo, es emplear estilos CSS para las imágenes, en lugar de etiquetas HTML de imagen. Esto consistiría en, en lugar de especificar la ruta de la imagen dentro del atributo "src" de la etiqueta HTML de imagen, especificarlo como imagen de fondo, en la propiedad "background-image" del CSS. A través de Media Queries se asignaría una imagen diferente en función de la resolución de la pantalla.

Una tercera opción consistiría en emplear el Elemento <picture> de HTML5 en lugar de los elementos . Pictures permite establecer más de un archivo fuente, con diferentes condiciones, de manera que únicamente se renderizará la primera fuente que cumpla las condiciones.

Adoptar esta medida supondría reducir considerablemente la descarga de datos producida por archivos de imagen en dispositivos con pantallas pequeñas, sin efecto alguno en pantallas grandes como ordenadores portátiles o de sobremesa. Sin embargo, si no se realiza ninguna de las automatizaciones sugeridas, requerirá una alta carga de trabajo para su mantenimiento.

Bibliografía

Ando, R. (4 de 11 de 2013). Google lifts post-Nokia hopes with Finnish data centre investment. *Reuters* .

Baldé, C. W. (2014). *The global e-waste monitor - 2014, United Nations University*. Recuperado el 7 de 7 de 2017, de <https://i.unu.edu/media/unu.edu/news/52624/UNU-1stGlobal-E-Waste-Monitor-2014-small.pdf>

Beck, K. O. (2 de 2001). *Principles Behind the Agile Manifesto*. Recuperado el 7 de 7 de 2017, de <http://agilemanifesto.org/principles.html>

Brito, J. G. (2016). *El uso de redes sociales por parte de las universidades a nivel Institucional. Un estudio comparativo*. Recuperado el 7 de 7 de 2017, de http://www.um.es/ead/red/32/laaser_et_al.pdf

CISCO. (2016). *Cisco Global Cloud Index: Forecast and Methodology, 2015–2020*. Recuperado el 7 de 7 de 2017, de <http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf>

CISCO. (28 de 3 de 2017). *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Pape*. Recuperado el 7 de 7 de 2017, de <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>

ETBC, E. T. (13 de 1 de 2016). *Facts and Figures on E-Waste and Recycling*. Recuperado el 7 de 7 de 2017, de http://www.electronicstakeback.com/wp-content/uploads/Facts_and_Figures_on_EWaste_and_Recycling1.pdf

HTTPArchive. (2017). *Mobile Trends*. Recuperado el 7 de 7 de 2017, de <http://mobile.httparchive.org/trends.php?s=All&minlabel=Aug+1+2013&maxlabel=Apr+15+2017#bytesTotal&reqTotal>

Iowa, U. o. (2001). *Image Data Compression*. Recuperado el 7 de 7 de 2017, de <http://user.engineering.uiowa.edu/~dip/lecture/DataCompression.html>

JEDEC, J. S. (4 de 2008). *JEDEC Standard, DDR3 SDRAM*. Recuperado el 7 de 7 de 2017, de <http://mermaja.act.uji.es/docencia/is37/data/DDR3.pdf>

Lewis, P. (5 de 6 de 2017). *Reduce the scope and complexity of style calculations*. Recuperado el 7 de 7 de 2017, de <https://developers.google.com/web/fundamentals/performance/rendering/reduce-the-scope-and-complexity-of-style-calculations>

Marcotte, E. (25 de 5 de 2010). *Responsive Web Design*. Recuperado el 7 de 7 de 2017, de <https://alistapart.com/article/responsive-web-design>

Meeker, M. (31 de 5 de 2017). *Internet Trends 2017 - Code Conference*. Recuperado el 7 de 7 de 2017, de <http://dq756f9pzlyr3.cloudfront.net/file/Internet+Trends+2017+Report.pdf>

Nielsen, J. (17 de 4 de 2006). *F-Shaped Pattern For Reading Web Content*. Recuperado el 7 de 7 de 2017, de <https://www.nngroup.com/articles/f-shaped-pattern-reading-web-content/>

Nielsen, J. (22 de 3 de 2010). *Scrolling and Attention*. Recuperado el 7 de 7 de 2017, de <https://www.nngroup.com/articles/scrolling-and-attention/>

OFcom. (2015). *Adults' media use and attitudes*. Recuperado el 7 de 7 de 2017, de https://www.ofcom.org.uk/__data/assets/pdf_file/0014/82112/2015_adults_media_use_and_attitudes_report.pdf

O'Neill, M. G. (2010). *Green IT for Sustainable Business Practice: An ISEB Foundation Guide*.

RealHardTechX. (6 de 2017). *POWER REQUIREMENTS FOR GRAPHICS CARDS*. Recuperado el 7 de 7 de 2017, de www.realhardtechx.com/index_archivos/Page362.htm

Sabharwal, M. C. (2010). *Impact of Idle Software on Battery Life*. Recuperado el 7 de 7 de 2017, de Intel Developer Forum 2010: <https://www.intel.ua/content/dam/doc/guide/idle-software-battery-life-idf2010-presentation.pdf>

Shebabi, A. S. (6 de 2016). *United States Data Center Energy Usage Report*. Recuperado el 7 de 7 de 2017, de Ernest Orlando Lawrence Berkeley National Laboratory: https://eta.lbl.gov/sites/default/files/publications/lbnl-1005775_v2.pdf

Vela, J. (2 de 2011). *The Environmental Impact Of Broadcasting Networks*. Recuperado el 7 de 7 de 2017, de BROADBAND PROPERTIES: http://www.bbcmag.com/2011mags/janfeb11/BBP_JanFeb11_EnvironmentallImpact.pdf

Vertatique. (25 de 3 de 2015). *Average Power Use Per Server*. Recuperado el 7 de 7 de 2017, de <http://www.vertatique.com/average-power-use-server>

Young, J. B. (19 de 4 de 2017). *Akamai Online Retail Performance Report: Milliseconds Are Critical*. Recuperado el 7 de 7 de 2017, de <https://www.akamai.com/us/en/about/news/press/2017-press/akamai-releases-spring-2017-state-of-online-retail-performance-report.jsp>