

REDISEÑO DE UN BLOG WORDPRESS IMPLEMENTANDO CCS GRID Y PWA

Memoria de Proyecto Final de Grado/Máster

GRADO EN MULTIMEDIA

Gestión y publicación de contenidos

El presente cuadro de texto tiene solamente fines informativos y no debe ser incluido en la memoria del alumno.

ACERCA DE LOS CONTENIDOS EN ESTE DOCUMENTO

Este documento incluye estilos predeterminados de texto, ejemplos de citas bibliográficas, notas a pie e inserción de figuras (imágenes y gráficos) y tablas, así como sección de bibliografía e índices automatizados listos para usar.

ACERCA DE LOS CAPÍTULOS DE ESTE DOCUMENTO

Aquellos capítulos con el título en color negro son obligatorios para todos los TF, mientras que aquellos en color gris son aquellos opcionales, susceptibles de ser incluidos en la memoria según el tipo de TF realizado. Es recomendable adaptar el orden de los capítulos a la naturaleza del TF a realizar, e incluso combinar dos o más capítulos en uno si se considera oportuno.

Autor: Javier Murillo Ruiz

Consultor: David Alcubierre Arenillas

Profesor: César Pablo Córcoles Briongos

18/06/2018

Créditos/Copyright

Una página con la especificación de créditos/copyright para el proyecto (ya sea aplicación por una parte y documentación por la otra, o unificadamente), así como la del uso de marcas, productos o servicios de terceros (inclusive códigos fuente). Si una persona distinta al autor colaboró en el proyecto, debe quedar explicitada su identidad y qué hizo.

El título de esta página sólo se muestra en esta pauta a nivel informativo.

Licencias recomendadas



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

B) GNU Free Documentation License (GNU FDL)

Copyright © 2018 Javier Murillo Ruiz.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

Dedicatoria/Cita

Después de siete años de carrera, comenzó un intenso periodo de unos meses para realizar el TFG. Ha sido un periodo de aprendizaje intenso que me ha llevado a ampliar mis conocimientos en el mundo de las tecnologías web pero también a hacer sacrificios en el ámbito personal. Por eso me gustaría dedicarlo a todas las personas que me han apoyado y ayudado durante este periodo.

Abstract

El proyecto aborda la implementación de una serie de tecnologías y rediseños que modificaran y mejoraran de forma sustancial un blog realizado en WordPress, se trata del blog de la asignatura “Lenguajes y estándares web¹”.

Los objetivos del proyecto son claros, realizar un rediseño que permita a un blog de Wordpress utilizar un sistema de diseño basado en CSS Grid, es decir que la interfaz siga una estructura de rejilla e implementar algunas características de las *Progressive web app* (PWA) para hacerlo más rápido, eficiente y a la vez *responsive*.

¹ <http://multimedia.uoc.edu/blogs/htmlcss/> (última consulta : 19/03/2018)

Abstract (english version)

The project addresses the implementation of a series of technologies and redesigns that will modify and substantially improve a blog made in WordPress, it is the blog from the subject "Languages and web standards".

The objectives of the project are clear, to make a redesign that allows a Wordpress blog to use a design system based on CSS Grid, that is to say that the interface follows a grid structure and implement some features of the Progressive web app (PWA) to make it faster, more efficient and at the same time responsive.

Agradecimientos, Notaciones y Convenciones

Agradecimientos

Me gustaría agradecer a Damian Junquera experto en desarrollo back-end la elección de las herramientas adecuadas para llevar a cabo el TFG, especialmente el framework basado en CSS Grid.

Convenciones

Las palabras en inglés irán en itálica, para el código fuente se usará la fuente "Code".

Índice

1. Introducción/Prefacio	10
2. Descripción/Definición/Hipótesis	11
3. Objetivos	13
3.1 Principales.....	13
3.2 Secundarios	13
4. Marco teórico/Escenario	14
4.1 CSS GRID.....	14
4.2 PROGRESSIVE WEB APP (PWA)	15
4.3 Implantación de CSS Grid en WordPress	15
4.4 Implantación de PWA en WordPress	16
5. Contenidos.....	18
5.1 Introducción teórica.....	18
5.2 Migración de contenidos	19
5.3 Tipos de contenidos	19
6. Metodología	20
7. Arquitectura de la aplicación/sistema/servicio.....	21
7.1 Cliente.....	21
7.2 Servidor.....	21
7.3 Base de datos	21
8. Plataforma de desarrollo	22
8.1 Software.....	22
8.2 <i>Plugins</i> utilizados	22
8.3 Hardware	22
9. Planificación.....	23
10. Proceso de trabajo/desarrollo.....	26
10.1 Primera entrega	26
10.2 Segunda entrega	26
10.3 Tercera entrega	26
11. APIs utilizadas	27
12. Diagramas UML.....	28
12.1 Diagrama de caso de uso.....	28
12.2 Sitemap	29
13. Prototipos.....	30

13.1 Lo-Fi	30
13.2 Hi-Fi.....	34
14. Guiones	38
15. Perfiles de usuario	39
16. Usabilidad/UX	40
16.1 Interacción con el administrador.....	40
16.2 Interacción con el usuario.....	40
17. Seguridad	41
18. Tests	42
18.1 Test versión de la versión alfa	42
18.2 Test versión de la versión Beta.....	42
18.3 Test de accesibilidad	43
18.4 Google PageSpeed.....	44
19. Versiones de la aplicación/servicio.....	45
19.1 Versión Alpha	45
19.2 Versión Beta	45
19.3 Versión definitiva	45
20. Requisitos de instalación/implantación/uso.....	46
20.1 Software	46
20.2 Hardware.....	46
20.3 Formación/Conocimientos	46
21. Instrucciones de instalación/implantación	47
22. Instrucciones de uso	51
23. Bugs.....	52
24. Proyección a futuro.....	53
25. Presupuesto.....	54
25.1 Equipo humano.....	54
25.2 Equipamiento técnico	54
25.3 Costes – valor del proyecto	54
26. Análisis de mercado	55
26.1 Estudio previo para la implantación de CSS Grid	55
26.2 Estudio previo para la implantación de PWA	55
27. Marketing y Ventas	56
28. Conclusión/-es	57
Anexo 1. Entregables del proyecto.....	58
Anexo 2. Código fuente (extractos)	59

Anexo 3. Librerías/Código externo utilizado.....	60
Anexo 4. Capturas de pantalla	67
Anexo 5. Guía de usuario	68
Anexo 6. Libro de estilo	69
Anexo 7. Glosario/Índice analítico	70
Anexo 8. Bibliografía.....	71
Anexo 9. Vita	72

Figuras y tablas

Lista de imágenes, tablas, gráficos, diagramas, etc., numeradas, con títulos y las páginas en las que aparecen.

Índice de figuras

Figura 1: Vista principal de la plantilla actual	10
Figura 2: Ejemplo CSS grid.....	14
Figura 3: Diseño web adaptativo	19
Figura 4: Logo ithemes Secutity	41
Figura 5: Test versión Alfa	42
Figura 6: Test versión Beta	42
Figura 8: PageSpeed test.....	44
Figura 8: Test versión Beta	45
Figura 9: Añadir nueva entrada.....	51

1. Introducción/Prefacio

Actualmente el mundo web avanza a pasos agigantados y a menudo nacen nuevas tecnologías que mejoran la anterior y nos lleva a plantearnos su implementación. Esta situación es la que ha llevado a realizar el estudio y valorar la posible implantación de CSS Grid y características de Progressive Web Apps (PWA) en un determinado blog de la Universitat Oberta de Catalunya.

El blog que se desea rediseñar está basado en WordPress un CMS (Sistema de gestión de contenidos) que junto con Drupal es uno de los más utilizados para este fin. Aunque durante mucho tiempo WordPress estaba considerado una plataforma solo para crear blogs ha ido evolucionando y permite también la creación de páginas web para empresas, tiendas online... etc. Esta evolución se debe en gran parte al desarrollo de *plugins* y temas o plantillas por parte de la comunidad, que ha permitido expandir sus fronteras y sus usos.

Dicho blog se puede encontrar actualmente en la URL <http://multimedia.uoc.edu/blogs/htmlcss/> (15/05/2018) . Nos encontramos con un CMS , para ser más exactos WordPress que utiliza una plantilla denominada “Blog_UOC” corriendo bajo la versión 4.9.5 de WordPress , además utiliza la librería js denominada “jquery” en su versión 1.12.4 . El tema es responsive resuelve adecuadamente los cambios de resolución. Estructuralmente dispone de cuatro bloques, un *header* o cabecera en la parte superior, un *footer* o pie, una sección de contenidos en el centro y en la parte izquierda el menú principal de navegación.



Figura 1: Vista principal de la plantilla actual

El principal problema de la plantilla actual viene al realizar un chequeo de accesibilidad donde muestra siete problemas conocidos y 96 problemas potenciales, entre ellos podemos destacar fallos en el contraste de los colores, no se puede identificar correctamente el idioma del texto, el código de lenguaje es incorrecto, no se puede utilizar solo con el teclado... en general el nivel de accesibilidad es muy mejorable.

2. Descripción/Definición/Hipótesis

Incorporar CSS Grid y PWA otorgará al blog existente una serie de características que mejoren su experiencia de uso en gran medida.

Css Grid es un layout que facilita el siempre complejo proceso de colocar y distribuir los elementos a lo largo de una página. Presenta un sistema de rejilla bidimensional para usar en CSS que permitirán posicionar las principales áreas de una página o pequeños detalles de la interfaz de usuario. Esto permitirá adaptar la página a todo tipo de dispositivos sin perder prestaciones, además facilitará el rediseño del blog pudiendo repartir los elementos en columnas (una o varias).

Las *progressive web apps* (PWA) son aplicaciones web que utilizan las últimas tecnologías disponibles en los navegadores para ofrecer una experiencia en móviles lo más parecida a una app nativa, siempre adaptándose al dispositivo ya sea móvil, Tablet, Ordenador...

Esto dotará al blog una mejora sustancial en su versión móvil, sobre todo, ya que notificará cuando existan nuevos, la velocidad de trabajo en móvil y en web será mucho más rápida, la experiencia de uso será más sencilla e intuitiva que la actual y se podrá acceder al blog sin conexión a la red.

Se ha optado por mantener Wordpress ya que es uno de los CMS más intuitivos y fáciles de utilizar que existen hoy en día por lo que se ha procedido a estudiar la mejor forma de introducir estas tecnologías en Wordpress.

Para el caso de la tecnología CSS Grid la implementación se realizará a través de la adaptación de una plantilla incorporando en el código CSS de la misma las funciones del *framework* indicado.

Las características PWA se incluirán mediante *plugins* ya que existe hoy en día multitud desarrollados que conjuntados de la forma adecuada permiten hacer funcionar un CMS Wordpress como si de una *progressive web app* se tratara. Los *plugins* se especificarán en detalle en los apartados siguientes, básicamente dotarán al blog de características innovadoras para acercarlo hacia el campo de las aplicaciones tanto móviles como de escritorio.

La base del rediseño será la utilización de una nueva plantilla de Wordpress, esta plantilla parte de base de una plantilla denominada "bones" la cual es gratuita y está pensada para que los desarrolladores partan de esta para comenzar a realizar modificaciones sobre la misma. La plantilla desarrollada se ha denominado "TFG" estará basada en HTML 5 y será *responsive* gracias a la utilización del *framework* "Simple Grid" la cual utiliza la metodología CSS Grid. Dispondrá de un nivel de accesibilidad AA por lo tanto dispondrá de opciones como texto alternativo a imágenes, sugerencias ante errores, títulos de las paginas...

Rediseño de un blog Wordpress implementando CSS Grid y PWA , Grado en Multimedia, Javier Murillo Ruiz

Dicha plantilla TFG utilizara también la conocida librería de javascript “*jQuery*”, se trata de una librería rápida y pequeña que contiene muchas funciones de cara a la manipulación de ficheros HTML y manejo de eventos, esta librería se ha utilizado sobre todo de cara la implementación de la versión móvil de la página web.

El paso previo será la instalación y puesta en marcha de WordPress en una su versión más básica para después comenzar con las implantaciones junto con una instalación de Xampp para poder ejecutar de forma local la página web.

3. Objetivos

3.1 Principales

Objetivos clave del TF.

- Implantación de CSS Grid
En este objetivo se incluyen todas las tareas de instalación y configuración o programación del entorno adecuado para crear el sistema de rejillas. Implantación y desarrollo del diseño web utilizando Simple CSS Grid junto con las etiquetas HTML correspondientes.
- Añadir características de progressive web app
Instalación, configuración y puesta en marcha de todos los *plugins* necesarios para convertir el blog en una PWA.

3.2 Secundarios

- Utilizar *plugin* Gutenberg
Instalar y configurar este novedoso *plugin*, conseguir integrarlo con el resto de extensiones de forma satisfactoria.
- Rediseñar el blog con una estética atractiva
En función de los requisitos básicos que se propongan y según la línea grafica a seguir el blog deberá tener una estética moderna y atractiva, mejorando la actual.
- Hacer que el blog sea accesible, acercarse lo mejor posible al grado de accesibilidad doble A incluyendo encabezados y etiquetas para describir mejor el tema, utilizar textos alternativos para las imágenes o un correcto uso del color.

Contenedor: Definirá la cuadrícula o rejilla

Ítem: Bloques que contiene una cuadrícula, elemento contenedor

Celda: Cada cuadro de la cuadrícula es una celda.

Área: región de celdas

Banda o grid tack: Grupo horizontal o vertical de celdas de una cuadrícula-

Línea: Separador de cuadrículas, puede ser horizontal o vertical.

Por lo tanto, antes de iniciar la programación de la nueva plantilla se ideará el rediseño en función de las filas y columnas deseadas añadiendo los ítems necesarios siempre teniendo en cuenta la estructura de grid.

4.2 PROGRESSIVE WEB APP (PWA)

Las *progressive web apps (PWA)* son aplicaciones web que utilizan las últimas tecnologías disponibles en los navegadores para ofrecer una experiencia en móviles lo más parecida a una app nativa , siempre adaptándose al dispositivo ya sea móvil, Tablet, Ordenador... Al ser aplicaciones web no requieren ni instalación ni descarga de ningún mercado de apps.

Por lo tanto, la implementación de una PWA debe de centrarse en facilitar la integración móvil creando una experiencia de uso lo más parecida a la de una aplicación nativa. Para conseguirlo se trabajará en los siguientes aspectos:

- Buen rendimiento tanto en escritorio como en móviles, que cargue lo más rápido posible.
- Una interfaz que parezca nativa.
- Posibilidad de trabajar sin conexión.

Poder enviar notificaciones a los usuarios, como una app nativa.

4.3 Implantación de CSS Grid en WordPress

Para crear un sistema de rejillas bidimensional hará falta previamente diseñar acorde a este sistema, los elementos de la página serán distribuidos en columnas, por lo tanto, la primera fase será identificar los elementos y distribuirlos adecuadamente.

Una vez que se tengan preparados los prototipos tanto Lo-fi como Hi-fi se procedería a la implantación en el sitio Wordpress.

Esta implantación se ha llevado a cabo utilizando el framework de CSS Grid llamado "Simple Grid"². Se trata de un sistema basado en CSS Grid con las siguientes características:

-*Responsive*: Está preparado para que los contenidos se adapten a múltiples dispositivos.

-Ligero: El CSS que incorpora no recargará la web por lo tanto no aumentará los tiempos de carga.

-Simple: Fácil de usar e integrar en los proyectos.

-12 columnas: Sistema de 12 columnas para crear de forma sencilla sitios web *responsive*. Las columnas tendrán filas que a su vez tendrán contenedores con un ancho máximo de 960px.

-Código fácil: Las clases CSS se manejan de forma sencilla, se deberán de llamar a una clase u otra en función del número de columnas que quieres que ocupe el contenido.

La implantación en WordPress se ha llevado a cabo partiendo de la plantilla base, a la que se han incorporado las clases CSS necesarias para que funcione correctamente el *framework*. Dichas clases han sido utilizadas con los contenedores Div y demás etiquetas HTML para que el WordPress resultante tenga el aspecto diseñado previamente en los Wireframes.

4.4 Implantación de PWA en WordPress

Como se comentaba en la introducción una PWA presenta una serie de características o requerimientos particulares. Se pretende cumplir con estos requerimientos a través de la utilización de *plugins* para WordPress.

Tras una investigación de los múltiples existentes se han elegido los siguientes:

WP Smush³:

Plugin que optimizará las imágenes y sus tamaños, permite elegir un tamaño máximo para las imágenes eligiendo el ancho y el alto permitido, esto permitirá que el blog cargue lo más rápido posible dando sensación de fluidez, como si de una aplicación instalada se tratara.

Plugin Gutenberg⁴:

² <http://simplegrid.io/>

³ <https://es.wordpress.org/plugins/wp-smushit/>

Este plugin ha sido lanzado recientemente y pretende facilitar la creación y edición de entradas en Wordpress. Gutenberg organiza todo el contenido en bloques, ya sean párrafos, citas, imágenes, galerías... Que lo hacen ideal para atraer a principiantes.

La edición por bloques permite que los elementos sean independientes y se pueden realizar ediciones u ordenaciones sin afectar al resto de elementos de la entrada.

Gutenberg permite utilizar nuevos elementos sin necesidad de incrustar código HTML como puede ser:

- Escritura en doble columna
- Tablas de datos
- Botones de acción (CTA)
- Galerías simétricas
- Inserción de widgets como contenido
- Inserción de publicaciones de redes sociales mediante la introducción simple de su URL
- Personalización de imágenes de portada

Gracias a esta extensión la interfaz de uso será mucho más sencilla de utilizar pareciendo casi nativa.

WP Super Cache ⁵:

Es una extensión de WP que guardará en memoria gran parte de los datos agilizando así la carga de la página web. Crea archivos HTML estáticos que serán enviados al cliente en lugar de los scripts PHP de WordPress Por lo tanto se podrá utilizar aunque los usuarios no estén conectados.

OneSignal⁶:

Permite el envío de notificaciones tanto a móviles como a través de los navegadores de escritorio.

⁴ <https://es.wordpress.org/plugins/gutenberg/>

⁵ <https://wordpress.org/plugins/wp-super-cache/>

⁶ <https://es.wordpress.org/plugins/onesignal-free-web-push-notifications/>

5. Contenidos

5.1 Introducción teórica

TIC aplicado a la enseñanza

Las TIC han permitido grandes avances en el área educativa, cada vez es más habitual la utilización de pizarras electrónicas, *tablets* y portátiles en las clases que permiten el acceso a multitud de información de una forma rápida y eficaz. Particularmente la utilización de blogs en la enseñanza permite al profesor ampliar o reforzar los conocimientos que imparte siendo por lo tanto una herramienta muy útil donde se pueden crear artículos o subir imágenes y videos para dicho fin.

Se pueden destacar tres usos de un blog en el ámbito académico:

- Herramienta de gestión del conocimiento: Ya que se pueden enlazar recursos propios o externos y categorizarlos.
- Ayuda al alumnado a reflexionar sobre su aprendizaje: Los propios estudiantes pueden crear sus blogs y por lo tanto sus propios artículos de investigación.
- Blog como herramienta de trabajo colaborativa: Alumnos y profesores comparten blog para llevar a cabo trabajos conjuntos.

El trabajo desarrollado implica la creación de un blog para su utilización en el ámbito académico, en el profesor podrá redactar artículos de interés para la asignatura y hacer que estén disponibles en el tiempo y para todas las clases. De esta forma se evita tener que estar creando un post en cada clase del campus y realizar esa misma operación cada año.

Nuevas formas de diseñar

El mundo web está en constante evolución ya que se tiene que adaptar continuamente a las nuevas formas de acceso a la red. Antiguamente siempre se accedía a través de un ordenador, pero con la llegada de los *smartphones*, portátiles convertibles y *tablets* se ha iniciado un proceso de adaptación de todos los sitios webs para estos nuevos dispositivos.

De esta necesidad de adaptación nace el diseño *Responsive*⁷ o diseño web adaptable que se encargará de adaptar la apariencia de las páginas web al dispositivo que esté utilizando para visitarlas.



Figura 3: Diseño web adaptativo

Este nuevo diseño ha llevado a multitud de páginas web a actualizar sus diseños, uno de los motivos es que el buscador Google va a empezar a penalizar a aquellas páginas que no dispongan de un diseño *responsive*.

5.2 Migración de contenidos

El trabajo llevado a cabo es un rediseño por lo tanto es importante que quede claro cómo van a mantenerse los contenidos. Lo más lógico y menos costoso es aplicar la nueva plantilla al blog actual y por lo tanto evitarse el traslado a un nuevo blog.

Por lo tanto, en el nuevo diseño estarán presentes las mismas categorías que en el blog antiguo, estas serán: HTML, CSS, Accesibilidad, Rendimiento, Software. El hecho de que ya se encuentren creadas las categorías solo haría necesario instalar la plantilla y los *plugins* asociados a la misma. Cuando esté totalmente configurada bastaría con cambiar el tema desde el selector de apariencia de la web de administración de Wordpress.

5.3 Tipos de contenidos

Tanto las entradas existentes como las futuras nuevas entradas podrán albergar videos, archivos de audio e imágenes, tanto enlazados como incrustados en los artículos del blog.

⁷ https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable

6. Metodología

Para llevar a cabo el rediseño del blog se ha utilizado el mismo proceso de diseño que se hubiera seguido para realizar uno nuevo desde cero ya que es la mejor forma de enfocarlo con una perspectiva totalmente nueva y atractiva.

Tiene cinco pasos que son los siguientes:

1-Entender

Con el fin de que nuestra idea pueda alcanzar el éxito deseado tendremos que delimitar el contexto lo máximo posible para comprender y afrontar de la mejor manera el desafío al que nos enfrentaremos. Por lo tanto, será clave aprender acerca de las partes interesadas saber sus metas, necesidades que es lo que buscan y si hay alguna otra solución.

2-Idear

La mejor forma de idear es sintetizar, tras interpretar los *Insights* (clave o solución a un problema) encontrados buscar la mejor metodología para conseguir una solución que de diferencia. Lo que tienes que dar como resultado un buen número de ideas. Será importante también captar problemas para poder buscar sus posibles soluciones.

3-Prototipar

Una vez extraída la esencia de la idea y con la metodología clara se la empieza a dar forma, la mejor manera es empezar con prototipos o *wireframes* sencillos incluso realizados en papel para luego poder realizar *mokups* más completos y que sirvan de “maqueta” del producto final. De esta forma produces algo tangible, descubres los retos a los que te vas a enfrentar y sus sutilezas.

4-Evaluar

Con el prototipo finalizado, antes de lanzarse a desarrollar se deben de buscar los posibles problemas a los que se enfrentará el equipo.

5- Desarrollar

Cuando el prototipo ha sido validado, ya sea por el cliente o por el equipo se puede comenzar el desarrollo y ejecución, llevar a cabo el rediseño y finalmente mostrarlo al público. Este último paso no solo incluye elaborar la versión final sino que también entra dentro la comercialización y marketing, el soporte, los manuales y el *branding*.

7. Arquitectura de la aplicación/sistema/servicio

A continuación, se detalla la arquitectura necesaria para el correcto funcionamiento del blog, para especificar los requisitos mínimos se tiene en cuenta la versión 4.9.5 de WordPress.

7.1 Cliente

Al tratarse de un CMS (WordPress) el cliente utilizado tanto para la creación de post como para su lectura por parte de los usuarios finales será un navegador web (como Internet Explorer , Mozilla o Chrome). Para que el 100% del blog funcione es necesario que el navegador utilizado sea compatible con las notificaciones “Web Push” , por lo tanto las versiones mínimas serán las siguientes:

Sistema operativo	Aplicaciones y versiones Soportadas
Android	Chrome 42, Edge 1.0.0
Mac OS	Chrome 42 , Firefox 44 , Safari
Ubuntu / linux	Chrome 42 , Firefox 44
Windows	Chrome 42 , Firefox 44

En el resto de navegadores o dispositivos (como IOS) el blog se mostrará correctamente pero no funcionaran las notificaciones web.

7.2 Servidor

En el servidor se almacenan los ficheros que componen el núcleo de WordPress así como los *plugins* y los temas (*Themes*) que tenga disponibles para utilizar. Dicho servidor podrá correr bajo Windows server o Linux ya que PHP es compatible con ambos sistemas operativos, la versión de PHP deberá de ser 7.2 o superior.

La estructura base de WordPress está compuesta por tres carpetas:

wp-admin: Donde están los archivos necesarios para el funcionamiento del panel de administración.

wp-content: Donde se encuentran los archivos que componen la interfaz de usuario (imágenes,*plugins*,plantillas..)

wp-includes: Resto de archivos para el funcionamiento básico de WordPress.

7.3 Base de datos

WordPress utiliza una BBDD MySQL, un sistema de bases de datos relacionales organizado en tablas. La instalación de WordPress creara una serie de tablas en la BD seleccionada, estas se podrán consultar y

editar aunque no es estrictamente necesario para el funcionamiento de WordPress ya que se pueden hacer las operaciones desde la interfaz de administración. La versión mínima de MySQL bajo la que funcionará correctamente WordPress será la 5.6 o superior o MariaDB 10.0 o superior.

8. Plataforma de desarrollo

8.1 Software

El desarrollo se ha llevado a cabo utilizando los siguientes programas:

- Sublime Text 3 build 3143 : Editor de textos utilizado para edición del código HTML, PHP y CSS.
- Xampp: Servidor local que permite probar las páginas web sin necesidad de acceso a internet. Está basado en Apache (PHP e incluye servidores de BBDD como MySQL ,versión 10.1.30-MariaDB) y SQLite. Se ha utilizado la versión 3.2.2.
- WordPress 4.9.5
- Adobe XD CC 2018 (Experience Design): Herramienta de prototipado para crear *wireframes* animados.

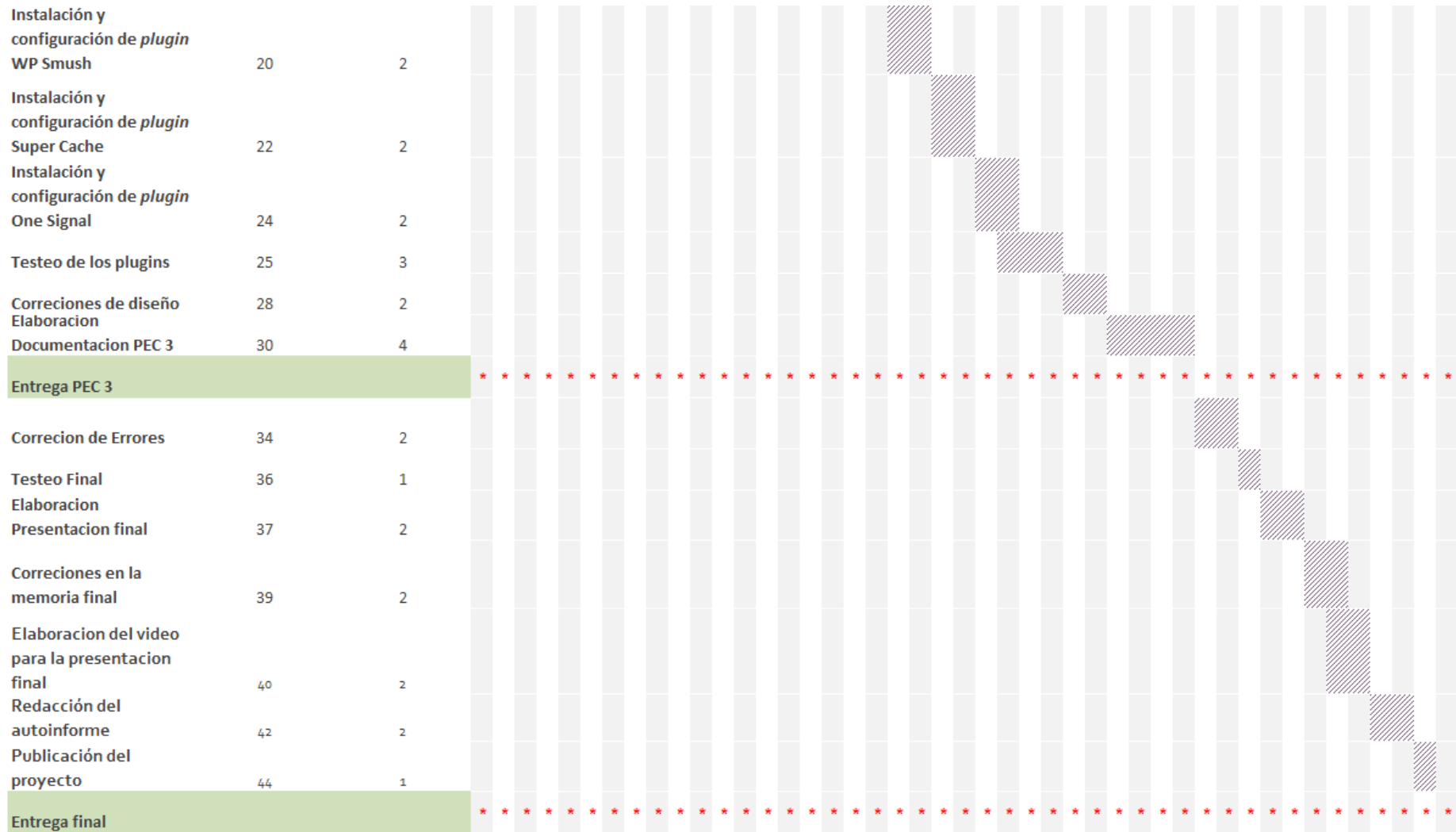
8.2 *Plugins* utilizados

- Gutenberg: Versión 2.4.0
- OneSignal Push Notifications: Versión 1.16.7
- Posts in Page: Versión 1.3.1
- WP Smush: Versión 2.7.8
- WP Super Cache: Versión 1.6.0

8.3 Hardware

- Equipo de escritorio: Utilizado para desarrollo, prototipado y pruebas.
- Smartphone Android: Utilizado para pruebas *responsive* y test de la PWA.

Rediseño de un blog Wordpress implementando CSS Grid y PWA , Grado en Multimedia, Javier Murillo Ruiz



Fases (Perfil)	Fecha inicio	Fecha fin
Definición de pre-requisitos (Jefe de proyecto)	03/03/2018	03/03/2018
Definición de objetivos (Jefe de proyecto)	04/03/2018	04/03/2018
Definición de Metodología (Jefe de proyecto)	05/03/2018	05/03/2018
Entrega PEC 1	06/03/2018	06/03/2018
Elaboración de Mokups lo fi (Diseñador Web)	08/03/2018	08/03/2018
Elaboración de Mokups hi fi (Diseñador Web)	09/03/2018	09/03/2018
Elaboración del diseño final del blog (Diseñador Web)	10/03/2018	10/03/2018
Instalación Xamp (Programador web)	11/03/2018	11/03/2018
Instalación WordPress (Programador web)	12/03/2018	12/03/2018
Instalación y configuración del <i>plugin</i> Gutenberg (Programador web)	13/03/2018	14/03/2018
Instalación y configuración de Simple CSS Grid (Programador web)	15/03/2018	17/03/2018
Implementación del diseño de grids propuesto (Programador web)	18/03/2018	21/03/2018
Elaboración documentación PEC 2 (Jefe de proyecto)	22/03/2018	26/03/2018
Entrega PEC 2	04/04/2018	04/04/2018
Instalación y configuración de <i>plugin</i> WP Smush (Programador web)	09/04/2018	11/04/2018
Instalación y configuración de <i>plugin</i> Super Cache (Programador web)	12/04/2018	14/04/2018
Instalación y configuración de <i>plugin</i> One Signal (Programador web)	15/04/2018	17/04/2018
Testeo de los plugins (Programador web)	18/04/2018	21/04/2018
Correcciones de diseño (Diseñador Web)	22/04/2018	24/04/2018
Elaboracion Documentacion PEC 3 (Jefe de proyecto)	24/04/2018	28/04/2018
Entrega PEC 3	06/05/2018	06/05/2018
Corrección de Errores (Jefe de proyecto)	07/05/2018	09/05/2018
Testeo Final (Jefe de proyecto)	10/05/2018	11/05/2018
Elaboración presentación final (Jefe de proyecto)	12/05/2018	14/05/2018
Correcciones en la memoria final (Jefe de proyecto)	15/05/2018	17/05/2018
Elaboración del video para la presentación final (Jefe de proyecto)	18/05/2018	20/05/2018
Redacción del autoinforme (Jefe de proyecto)	21/05/2018	23/05/2018
Publicación del proyecto (Jefe de proyecto)	24/05/2018	24/05/2018
Entrega final	18/06/2018	18/06/2018

10. Proceso de trabajo/desarrollo

10.1 Primera entrega

Durante este periodo de tiempo se ha trabajado en los aspectos preliminares del proyecto.

La definición de pre-requisitos fue la primera acción a llevar a cabo, aquí se definieron las tecnologías a implantar así como sus requisitos. A este trabajo le siguió el análisis de los objetivos a cumplir tanto principales como secundarios saliendo como resultado dos objetivos primarios y dos secundarios. Por último, se definió la metodología, al tratarse de un rediseño que propuso como metodología el Manual de proceso de diseño.

10.2 Segunda entrega

En esta segunda fase del proyecto se ha iniciado el proceso de prototipado superada ya la fase de entendimiento e ideación del proyecto.

Se han elaborado los *mokups* tanto en Low res como en Hi Res que dan una idea de cómo quedara la página web una vez finalizado el re-diseño. Con la propuesta de diseño clara se inició la instalación del software necesario para implantar el entorno de desarrollo, básicamente consistió en la instalación de Xampp, WordPress. Con la base instalada comenzó la instalación de los primeros *plugins* y el inicio del desarrollo de la plantilla o *theme* partiendo de una pre existente llamada "bones". Se entrega la maquetación de la página web en formato desktop finalizada, faltando las versiones móviles de la misma.

10.3 Tercera entrega

En esta fase ya se entrega el proyecto casi terminado, están en funcionamiento tanto la versión desktop como la versión móvil. Además, se encuentran instalados, activados y configurados los *plugins* que dotan al blog de características de una PWA. Por lo tanto el proyecto ya entra en su fase final faltando solo el chequeo de errores y las pruebas finales.

11. APIs utilizadas

Indicar las APIs de terceros que se hayan utilizado para el trabajo, y detallar su uso.

WordPress

Al tratarse del rediseño de un blog a través de un CMS se hace caso primordial el uso de un CMS. Se optó por WordPress que básicamente es el motor del blog rediseñado. Este permitirá crear artículos, personalizar la interfaz y añadir funcionalidades mediante el uso de *plugins*.

Simple Grid

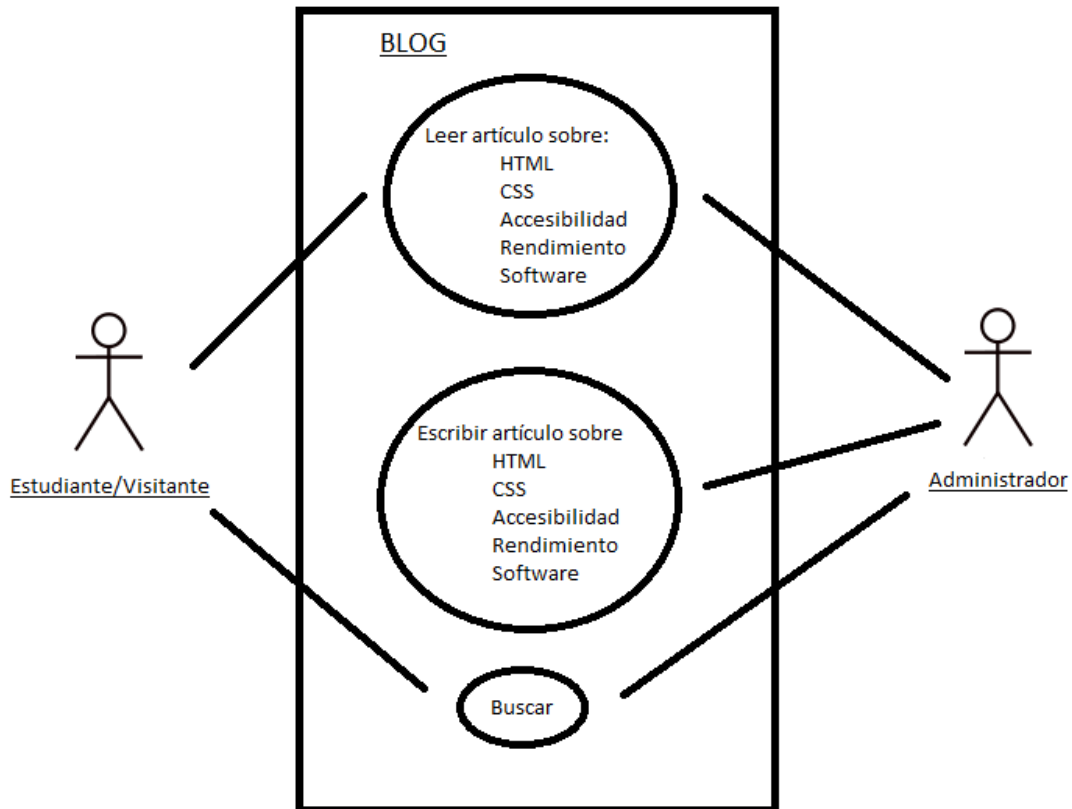
Se trata de un *framework* para realizar diseños *responsive* basado en la metodología CSS Grid. Se ha utilizado para maquetar el blog y adaptarlo al diseño que se propuso en los *wireframes*. Simple Grid dispone de una serie de clases que permiten dividir los elementos contenedores en columnas adaptables a distintas resoluciones. Dichas clases se han aplicado a las etiquetas PHP / HTML correspondientes para aplicar el la tecnología de esta API.

JQuery

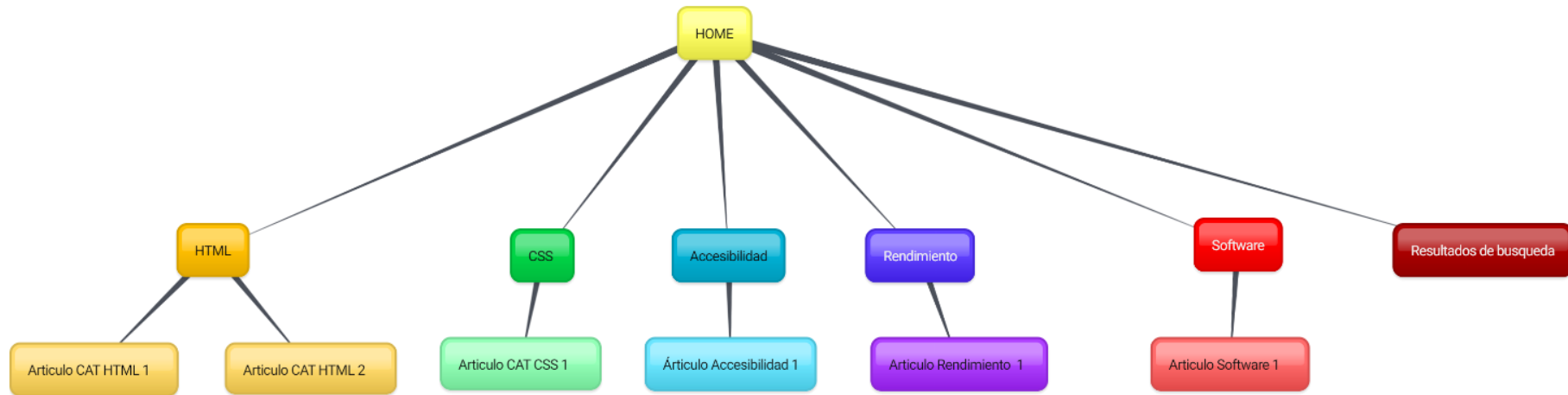
Se ha utilizado la versión v1.12.4 de jQuery , principalmente ha servido para controlar los eventos del menú hamburguesa cuando al resolución es ajustada a dispositivos móviles.

12. Diagramas UML

12.1 Diagrama de caso de uso



12.2 Sitemap



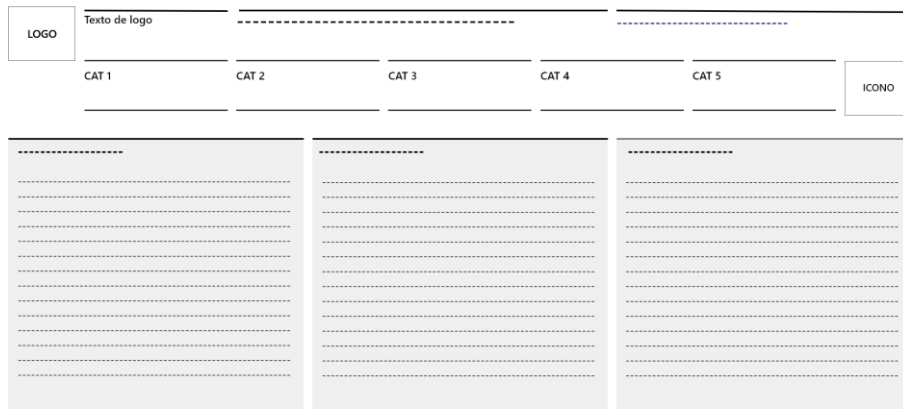
created with www.bubbl.us

13. Prototipos

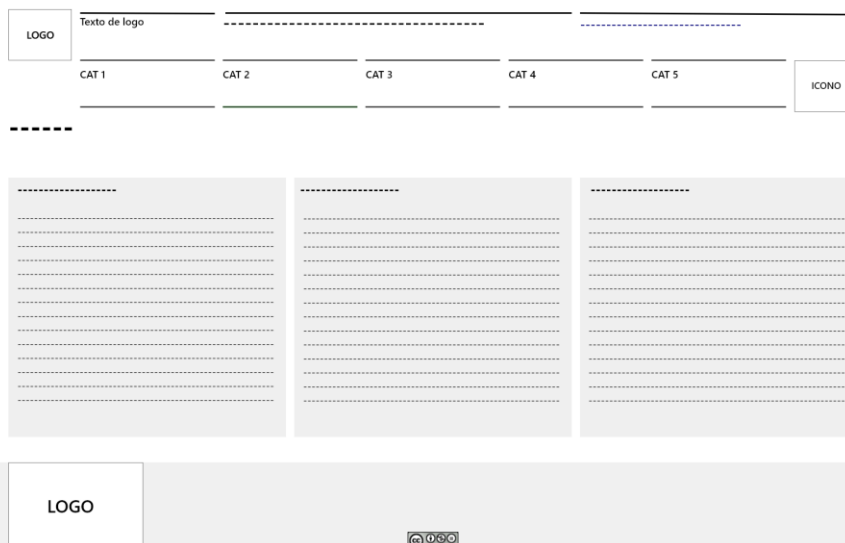
13.1 Lo-Fi

Versión desktop:

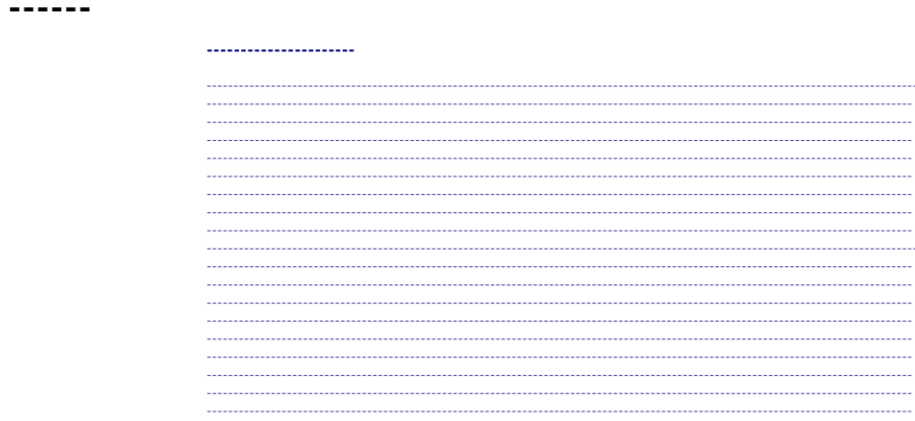
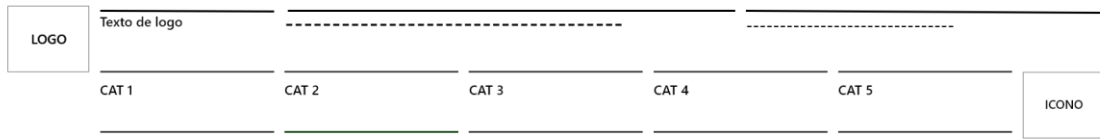
Pantalla principal



Pantalla de selección de categoría

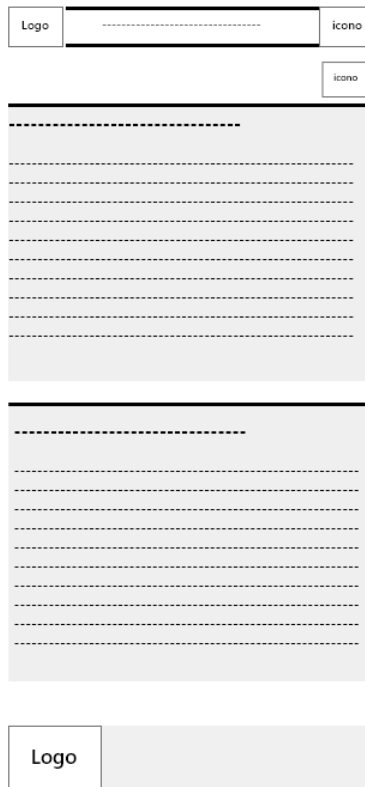


Pantalla de detalle



Versión móvil

Pantalla principal



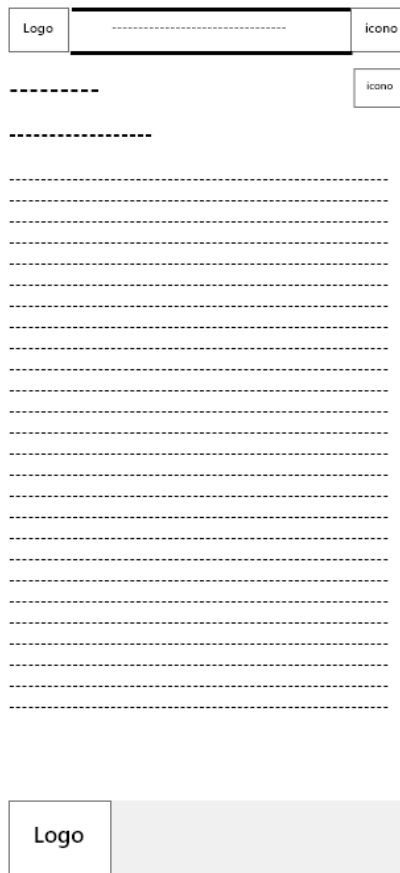
Selección de menú móvil

A mockup of a mobile menu selection screen. At the top left is a box labeled "Logo". To its right is a close button represented by an "X" symbol. Below these are six horizontal menu items, each consisting of a dashed line representing text and a small square checkbox to its right. The items are separated by thin horizontal lines.

Pantalla de selección de categoría

A mockup of a category selection screen. At the top left is a box labeled "Logo". To its right is a search input field with a dashed line for text and a button labeled "icono". Below the search field is a list of category items, each with a dashed line for text and a small square checkbox to its right. The items are separated by thin horizontal lines. At the bottom of the screen is a footer area containing a box labeled "Logo" on the left and a grey rectangular area on the right.

Pantalla de detalle



13.2 Hi-Fi

Versión desktop:

Pantalla principal

UOC Universitat Oberta de Catalunya

Lenguajes y estandares Web

Grado en Multimedia

HTML CSS Accesibilidad Rendimiento Software

Otros selectores CSS

(Material adaptado y traducido de https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors) Hemos visto en los materiales de la asignatura selectores del estilo [atributo], que representa los elementos con un atributo de nombre atributo y [atributo=valor], que representan, naturalmente, los elementos con atributo atributo con valor exactamente valor. Pero disponemos de otros selectores relacionados a través de correspondencia de patrones...[Leer más »](#)

Para saber más sobre accesibilidad

Conceptos básicos de accesibilidad y Pruebas de accesibilidad. Easy Checks – A First Review of Web Accessibility (W3C, incluido dentro de su Web Accessibility Initiative) How to use NVDA and Firefox to test your web pages for accessibility (2009) The Accessibility Project: A community-driven effort to make web accessibility easier. Una presentación...[Leer más »](#)

Optimizadores de CSS

CSSTidy es un 'parser' y optimizador de CSS, disponible como archivo ejecutable (Windows, Linux y OSX). DirtyMarkup es una herramienta de limpieza de código HTML, CSS y JavaScript. CSS LINT es una herramienta de análisis de CSS que sugiere mejoras a nuestro código...[Leer más »](#)

UOC Universitat Oberta de Catalunya

Estudios de Informática, Multimedia y Telecomunicación

CC BY-NC-SA

Pantalla de selección de categoría

UOC Universitat Oberta de Catalunya

Lenguajes y estandares Web

Grado en Multimedia

HTML CSS Accesibilidad Rendimiento Software

CSS

Otros selectores CSS

(Material adaptado y traducido de https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors) Hemos visto en los materiales de la asignatura selectores del estilo [atributo], que representa los elementos con un atributo de nombre atributo y [atributo=valor], que representan, naturalmente, los elementos con atributo atributo con valor exactamente valor. Pero disponemos de otros selectores relacionados a través de correspondencia de patrones...[Leer más »](#)

Tipografía

Una excelente charla introductoria dada por Jason Santa Maria en junio de 2012 en el marco del evento An Event Apart Boston (no entra en aspectos técnicos): Recursos The Elements of Typographic Style Applied to the Web Typedia, A Shared Encyclopedia of Typefaces Artículos Fonts, fabulous fonts: How to use the @font-face ...[Leer más »](#)

Cross-browser

Dada la inmensa variedad de navegadores que existen en el mercado, es extremadamente complicado que una página se vea igual en todos ellos. En ocasiones, eso es deseable: en general no queremos que una página se vea igual en un navegador 'de escritorio' que en la pantalla de un móvil, pero en...[Leer más »](#)

UOC Universitat Oberta de Catalunya

Estudios de Informática, Multimedia y Telecomunicación

CC BY-NC-SA

Pantalla de detalle



CSS

Otros selectores CSS

(Material adaptado y traducido de https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors)

Hemos visto en los materiales de la asignatura selectores del estilo [atributo], que representa los elementos con un atributo de nombre atributo y [atributo=valor], que representan, naturalmente, los elementos con atributo atributo con valor exactamente valor.

Pero disponemos de otros selectores relacionados a través de correspondencia de patrones (pattern matching):

[atributo~=valor]

Representa los elementos con un atributo atributo cuyo valor sea una lista de palabras separadas por espacios y uno de los cuales sea exactamente valor.

[atributo|=valor]

Representa los elementos con un atributo atributo y valor exactamente valor o que comience por valor seguido inmediatamente de un guión - (U+002D).

[atributo^=valor]

Representa los elementos con un atributo atributo y valor comenzando por valor.

[atributo\$=valor]

Representa los elementos con un atributo atributo y valor acabando en valor.

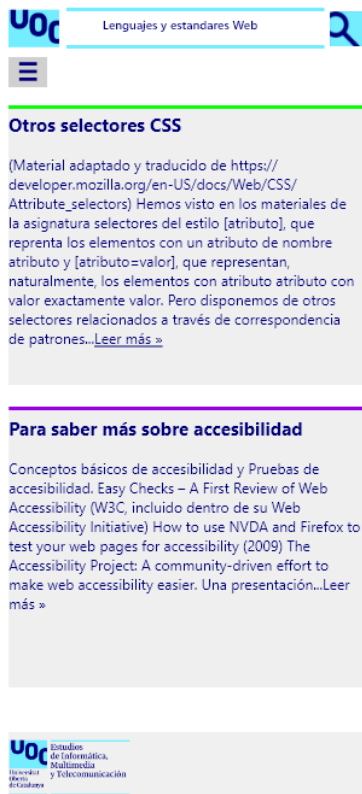
[atributo*=valor]

Representa los elementos con un atributo atributo y valor conteniendo valor.

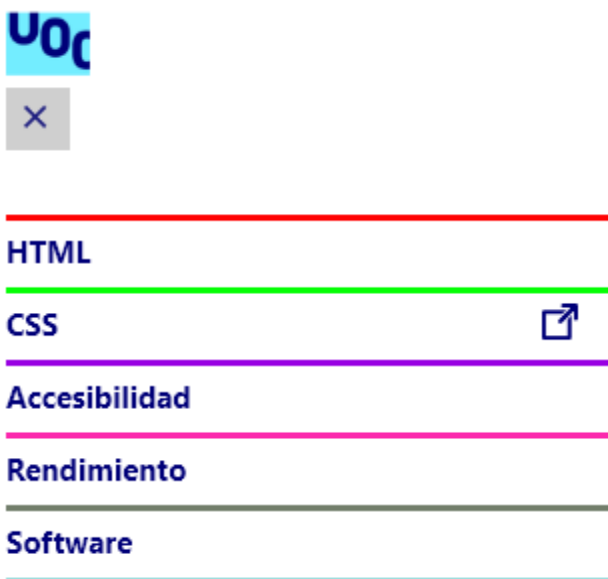


Versión móvil:

Pantalla principal



Selección de menú móvil



Pantalla de selección de categoría



Pantalla de detalle



CSS

Otros selectores CSS

(Material adaptado y traducido de https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors)

Hemos visto en los materiales de la asignatura selectores del estilo [atributo], que representa los elementos con un atributo de nombre atributo y [atributo=valor], que representan, naturalmente, los elementos con atributo atributo con valor exactamente valor.

Pero disponemos de otros selectores relacionados a través de correspondencia de patrones (pattern matching):

[atributo~ =valor]

Representa los elementos con un atributo atributo cuyo valor sea una lista de palabras separadas por espacios y uno de los cuales sea exactamente valor.

[atributo]=valor]

Representa los elementos con un atributo atributo y valor exactamente valor o que comience por valor seguido inmediatamente de un guión - (U+002D).

[atributo^ =valor]

Representa los elementos con un atributo atributo y valor comenzando por valor.

[atributo\$=valor]



14. Guiones

NO APLICA

Guiones creados para el trabajo/proyecto y comentarios a los mismos.

15. Perfiles de usuario

Estudiante: Será alumno de la universidad y específicamente de la materia: Lenguajes y estándares Web. Accederá a la web para la lectura de los artículos con el fin de reforzar o ampliar conocimientos en la materia.

Visitante anónimo: Accederá al blog seguramente a través de alguna búsqueda y estará interesado en el material que se ofrece.

Profesor: Este usuario podrá escribir artículos y modificarlos.

Administrador: Usuario con acceso a todo el blog, se encargará de su administración realizando tareas de mantenimiento como actualizaciones, instalación de *plugins* , configuración de plantillas.

16. Usabilidad/UX

16.1 Interacción con el administrador

El administrador del sitio trabajara con la interfaz de WordPress utilizando su panel de administración, en la parte de la izquierda de la pantalla estarán los menús de WordPress para crear páginas, artículos, instalar plugins.. .etc

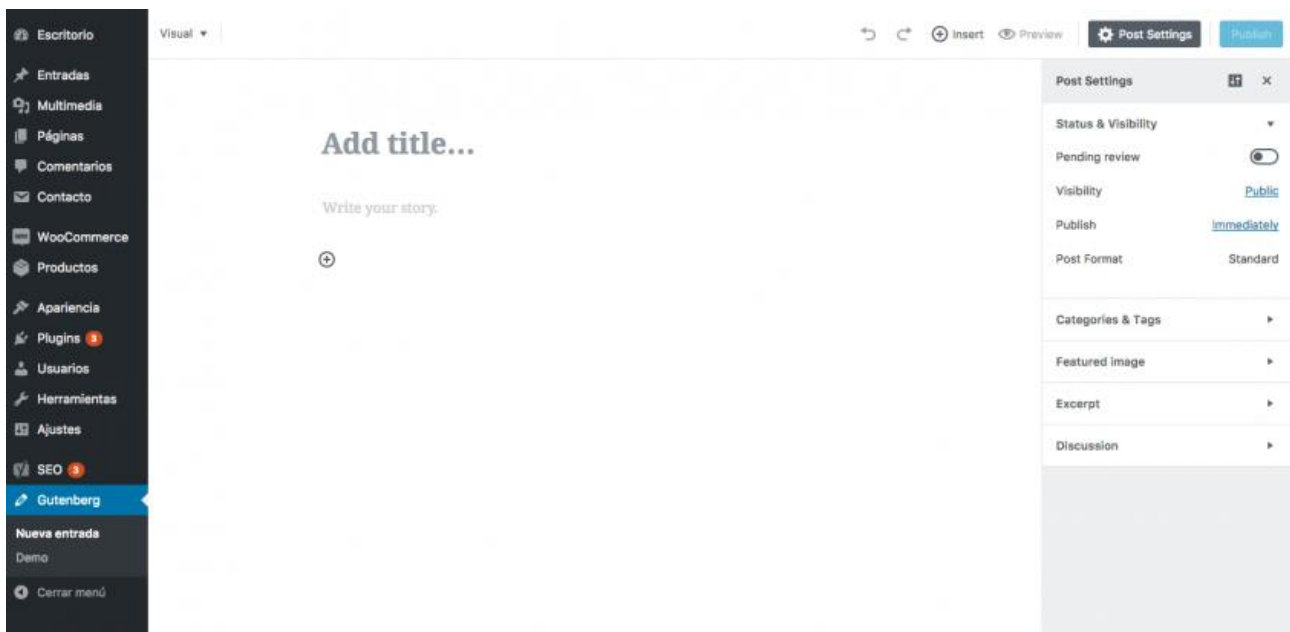


Figura 3: Interfaz WordPress

16.2 Interacción con el usuario

La página web dispondrá de un menú superior desde el cual se podrán seleccionar las distintas categorías de artículos disponibles, para entrar en ellas será necesario pinchar o hacer clic en el nombre de la categoría deseada.

Tanto la página principal como la de categoría dispondrán de varios artículos en función de los que estén dados de alta, dichos artículos mostrar su nombre y una brece introducción, para acceder a la página del articulo será necesario antes hacer clic en el nombre del articulo deseado o en la coletilla “Leer más >>” disponible al final del resumen.

La versión móvil funciona de una forma muy parecida al a versión desktop, la principal diferencia es la forma de mostrar el navbar o menú de navegación. En esta el navbar estará oculto y solo se mostrará seleccionado un botón con tres líneas verticales denominado “menú hamburguesa”

17. Seguridad

El CMS WordPress incorpora un sistema de *login* para acceder al panel de administración, por lo tanto los usuarios con permiso para crear contenido y administradores dispondrán de un usuario y un password . Los usuarios normales usaran el blog a modo de consulta a sí que no será necesario la inclusión de ningún sistema de *login* para ellos.

Para controlar los posibles ataques automatizados o las vulnerabilidades que puedan encontrarse Wordpress se ha utilizado el plugin iThemes Security⁸ .

Este plugin posee números características, pero podemos destacar las siguientes:

- Evita ataques de fuerza bruta a través del baneo de servidores que realizan muchas peticiones de acceso.
- Fuerza la utilización de SSL en las pagina de administración o en cualquier página o entrada.
- Evita que los archivos de WordPress se puedan modificar desde el área de administración.



Figura 4: Logo de ithemes Security

⁸ <https://es.wordpress.org/plugins/better-wp-security/>

18. Tests

18.1 Test versión de la versión alfa

Al tratarse de una versión muy preliminar se han realizado sobre todo test para poner a prueba la parte maquetada y chequear que los elementos en pantalla de la versión desktop continúan en su sitio tras añadir nuevos elementos.

Por lo tanto, este test ha consistido en añadir artículos de test a todas las categorías para comprobar que marca las *cards* con el método de color correcto.

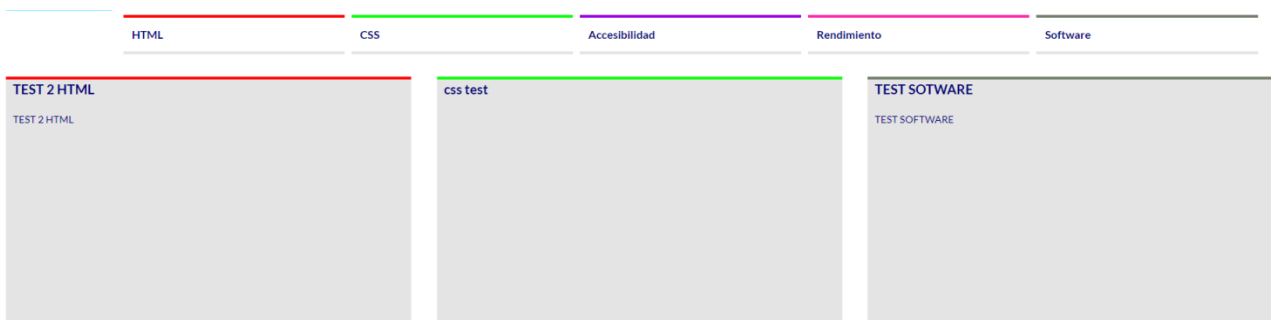


Figura 5: Test versión Alfa

Al mismo tiempo se ha chequeado que las *cards* se agrupan en filas de a tres como estaba definido en el prototipo.

18.2 Test versión de la versión Beta

En esta versión a se ha adaptado la página para que funcione con dispositivos móviles, por lo tanto, se ha utilizado el modo “inspector” de Google Chrome para convertir la página en formato móvil y realizar las pruebas pertinentes.

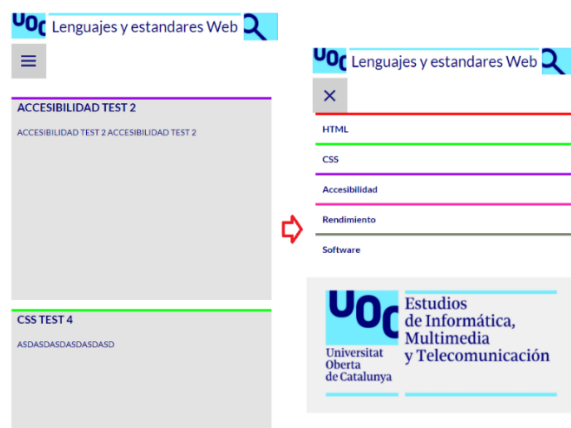


Figura 6: Test versión Alfa

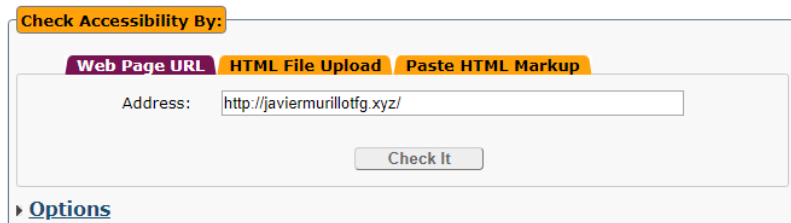
18.3 Test de accesibilidad

Se han realizado un tests de accesibilidad tanto a la nueva plantilla como a la antigua para comprobar las diferencias entre ambas y comprobar que niveles de accesibilidad alcanza cada una. Estos test se han llevado a cabo utilizando la herramienta “AChecker⁹”.

La plantilla antigua o a rediseñar presenta varios errores entre los que podemos destacar:

- No es adaptable 100% el contenido no se podrá presentar de diferentes maneras sin perder información o estructura.
- A nivel de contraste de colores algunos elementos no son muy distinguibles, el contraste entre el fondo y los colores de las fuentes no son los adecuados.
- Existen elementos *label* sin texto.

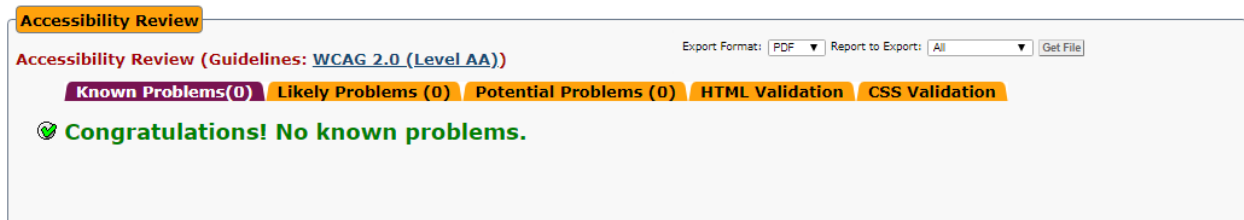
En cambio, si evaluamos la plantilla creada no se muestran errores conocidos de accesibilidad.



Conformance Seal(s) for your Web page

To show your readers that you have taken the care to create a web page that conforms to accessibility guidelines, you may display this(these) conformance seal(s) on the page that validates. Below is the HTML you could use to add the seal(s) to your web page. You may copy the seal image to your local Web site, and adjust the path to the image accordingly:

```
<p>  
<a href="https://achecker.ca/checker/index.php?uri=referer&gid=W3AG2-AA">  
  
</a>  
</p>
```



⁹ <https://achecker.ca/>

18.4 Google PageSpeed

PageSpeed Insight da una nota de 85/100 a la web, lo que está realmente bien indicando que la web tiene una buena velocidad de carga. Es importante señalar que los test se han realizado contra una web alojada en un servidor web de bajo coste, es muy probable que mejorara la nota al ser alojada a un servidor de producción.

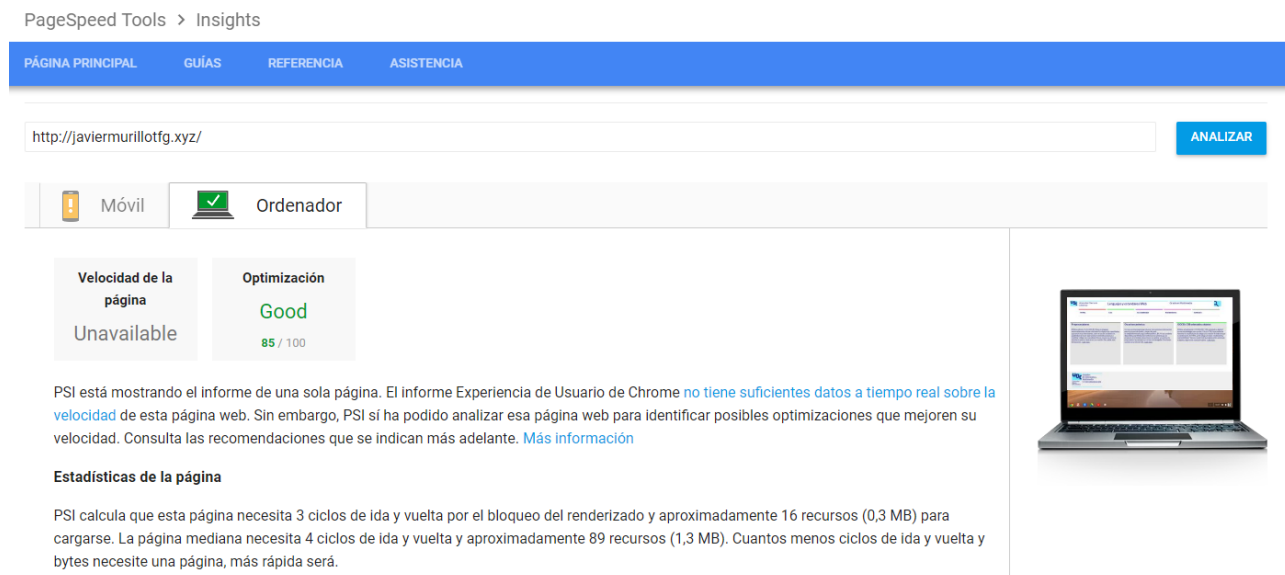


Figura 7: PageSpeed test

El test de google da una serie de recomendaciones a llevar a cabo como la optimización de javacript y optimizar aún más del fichero CSS.

19. Versiones de la aplicación/servicio

19.1 Versión Alpha

En esta primera versión se ha implementado Wordpress junto con los *plugins* “Gutenberg” y Post in page. Además de han dado de alta las categorías del blog, se han creado las paginas necesarias para dichas categorías junto con artículos de prueba. Por último se ha incorporado el diseño basado en CSS grid y propuesto en los wireframes, es importante remarcar que en esta primera versión aun solo funciona en la modalidad desktop, falta por desarrollar la versión móvil.



Figura 8: Versión alfa

19.2 Versión Beta

En esta segunda versión se han implementado todas las características necesarias para hacer funcionar la web en dispositivos móviles, además se ha introducido la funcionalidad “buscador” la cual se podrá utilizar seleccionado el icono con la lupa.

19.3 Versión definitiva

En esta versión se han finalizado todos detalles pendientes, como por ejemplo dar al buscador y a la campana para suscribirse a las notificaciones un estilo acorde con la gama cromática de la web.

20. Requisitos de instalación/implantación/uso

20.1 Software

Cliente

Es necesario la utilización de un navegador web, se recomienda utilizar la última versión disponibles.

Servidor

Sera necesario crear la estructura WordPress antes de instalar la plantilla, por lo tanto, los requerimientos a nivel de servidor serán los mismos que podría tener cualquier otro CMS WordPress:

- PHP 5.2.4 o superior.
- MySQL 5.0.15 o superior.
- No es indispensable, pero se recomienda tener el módulo Apache mod_rewrite.
- Lo ideal, es que el hosting corra sobre una plataforma Linux.

20.2 Hardware

Se necesitará como mínimo de 1 core de 15 GHZ a 64 bits 2 GB de RAM, disco de RAID 1 o superior por hardware, según las especificaciones técnicas del propio Wordpress,

20.3 Formación/Conocimientos

No será necesaria formación específica para la utilización del blog por parte de los usuarios, por parte del profesorado que vaya a crear contenido en el Blog será necesario una breve formación acerca de WordPress y el plugin Gutenberg que se utilizara para crear los artículos.

21. Instrucciones de instalación/implantación

A continuación, se detalla cómo implantar la plantilla creada.

1º Instalar Xampp

En caso de que desee utilizar en local.

2º Crear una db para Wordpress

Desde phpadmin se deberá crear una base de datos para WordPress, el nombre de la bbdd creada en MySQL debe de ser "Wordpress". El usuario root sera quien se conecte para la instalación del CMS así que su contraseña se indicará en el fichero wp-config.php de la carpeta de instalación de Wordpress, en la línea 25, sustituir la que se indica por la que se elija.

3º Instalar Wordpress

Una vez creada la bbdd se puede iniciar la instalación de Wordpress, copiar la carpeta "Wordpress" descargada previamente desde la página web oficial de WordPress.

4º Copiar la plantilla TFG

Copiar la plantilla suministrada en la ruta: \wordpress\wp-content\themes.

5º Utilizar la plantilla tfg

Acceder al panel de administración de WordPress (<http://localhost/wordpress/wp-admin/>) , una vez allí seleccionar la plantilla "TFG" desde apariencia-> Temas.

6º Activar los plugins de apariencia

Los *plugins* deberán ser descargados de sus respectivos sitios web, después habrá copiarlos en su carpeta correspondiente: \htdocs\wordpress\wp-content\plugins y después habrá que activarlos desde el menú "*plugins*" del panel de administración de WordPress. Activar los siguientes:

- Gutenberg
- Posts in Page (sustituir el fichero "posts_loop_template.php" por el suministrado)
- iThemes Security

7º Crear las categorías del blog

Se deberán crear las siguientes categorías (si se desea replicar la misma estructura del blog actual):

- Accesibilidad
- CSS
- HTML
- home
- Principal
- Rendimiento
- Software

8º Crear paginas para categorías

Crear las páginas con el mismo nombre que las categorías para ser asociadas mediante el plugin post in page:

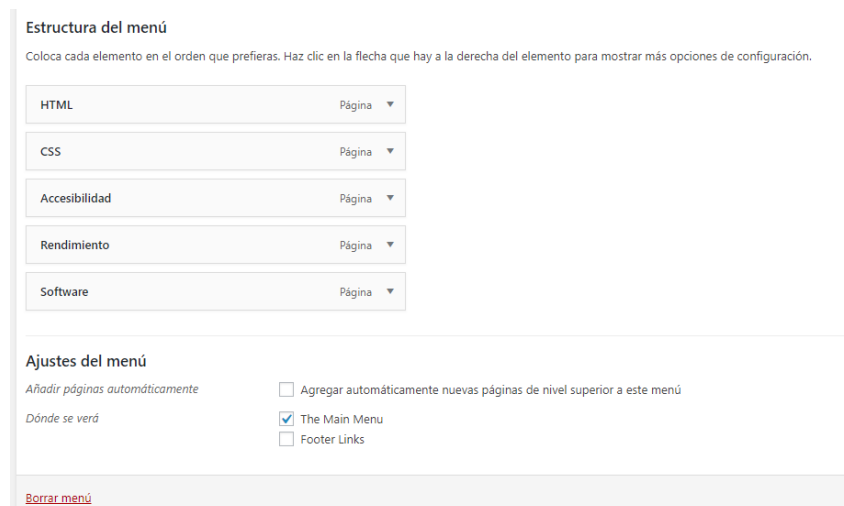
[ic_add_posts category='CAT'] (Cambiar CAT por el nombre de la categoría seleccionada)

Desde el menú personalizar de WordPress seleccionar ajustes de portada y poner la página “Principal” como página de inicio.

9º Gestionar apariencia

Seleccionar color de fondo: blanco.

Crear el menú con las categorías creadas desde “Apariencia” → “Menús”



10º Instalar y activar plugins PWA

- WP Smush.

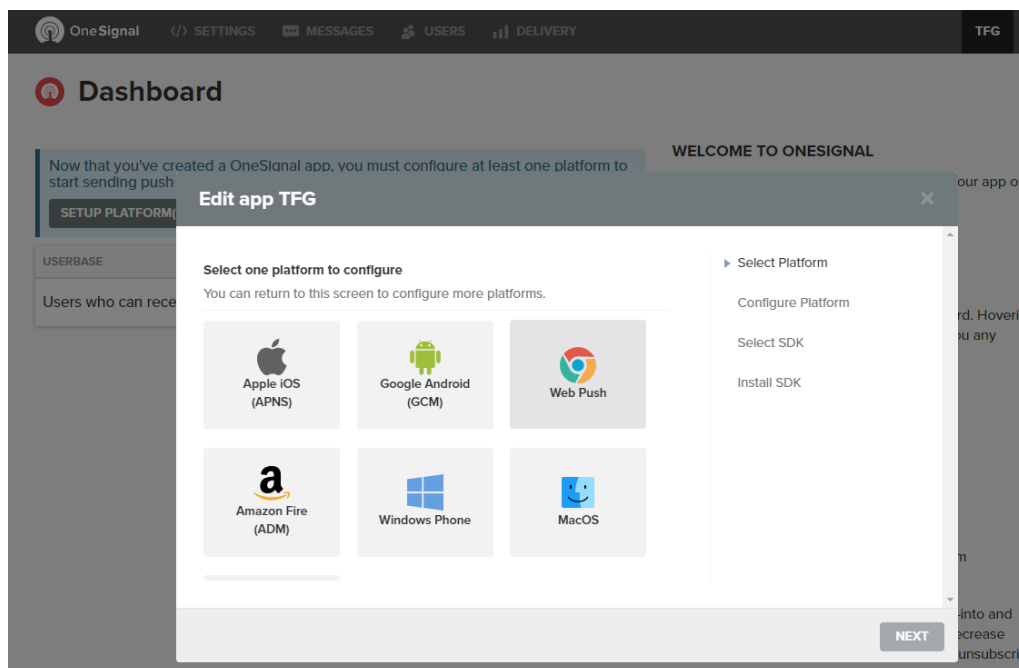
Una vez activado se utilizará como método por defecto para subir contenido multimedia.

- WP Super Cache.

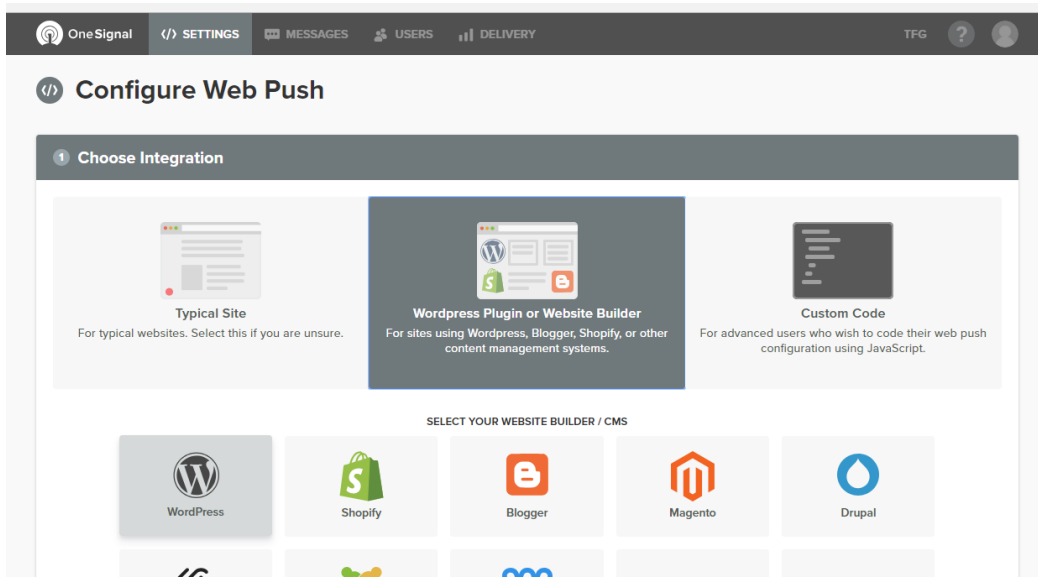
Al activarlo requerirá utilizar una estructura clara de URL para su correcto funcionamiento, dispone de un asistente para llevar a cabo la configuración.

- OneSignal Push Notifications.

Al activa este plugin se habilitan las notificaciones a dispositivos móviles, una vez activado dirigirse a nuevo menú creado en el administrador de WordPress con nombre “OneSignal Push”. Se deberá de crear una cuenta en OneSignal. Una vez creada se añadirá una app desde su página web y se seleccionará Web Push como tipo de plataforma:



Seleccionar integración con WordPress añadiendo los datos del sitio web:



Una vez finalizado el asistente copiar el APP ID y el API KEY y añadirlo en la configuración del *plugin* en WordPress:

Account Settings

My site uses an HTTPS connection (SSL) ?

App ID ?

a2a949bc-b8b2-463a-a0c1-709be4188024

REST API Key ?

MzE2NGEwMTQ4YTY1M2My00ZjBjLTgwMDgtM2YyMTQ4NGQ1ZGRm

OneSignal Label ?

UOC BLOG

Once your site is public, do not change your label. If you do, users will receive duplicate notifications.

Safari Web ID ?

UOC BLOG

Desde la misma ventana activar “Send notifications additionally to iOS & Android platforms”.

22. Instrucciones de uso

La utilización de la plantilla una vez instalada y configurada es sencilla ya que se utiliza del mismo modo que cualquier plantilla WordPress. El contenido se creará desde el panel de administración de WordPress desde entradas – “Añadir nueva”

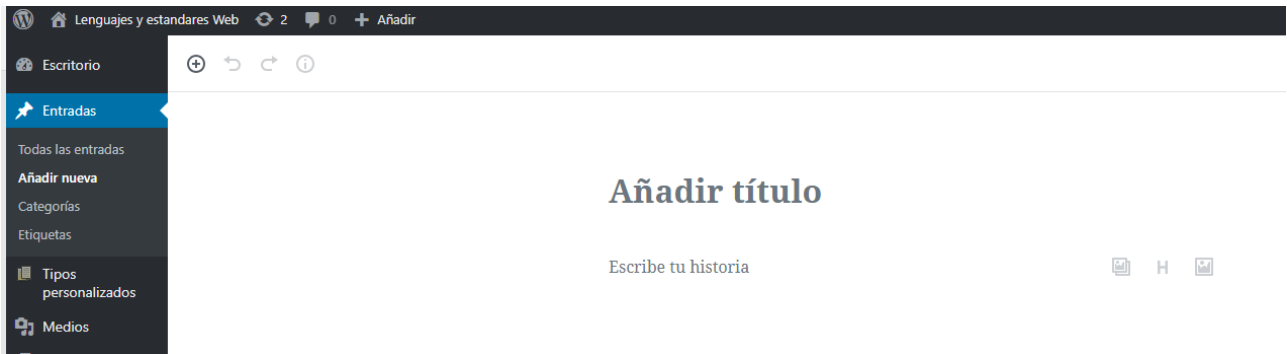


Figura 9: Añadir nueva entrada

23. Bugs

Lista de errores detectados en la funcionalidad del servicio/aplicación.

24. Proyección a futuro

Actualmente la plantilla está preparada para trabajar orientada sobre todo al blog que se pretende rediseñar, “Lenguajes y estándares Web”. Funcionaria correctamente utilizándolo para otras posibles asignaturas, pero sería realizar algunas modificaciones en el código para por ejemplo configurar es esquema de colores de las categorías.

Por lo tanto, una posible mejora a futuro sería la de incluir la gestión de colores por categoría en la propia plantilla WordPress y hacer posible su modificación desde el menú “Apariencia” “Colores” usando siempre la interfaz sin alterar el código fuente de la plantilla.

Por otro lado, sería interesante mantener actualizados los *plugins* y la versión de WordPress, será necesaria la realización de pruebas con las nuevas versiones que vayan saliendo para asegurarse del correcto funcionamiento de la plantilla y permitir así que perviva en el futuro.

25. Presupuesto

25.1 Equipo humano

Para llevar a cabo el rediseño de blog se requerirá solo de una persona que realizará el prototipado y su posterior implementación, así como las pruebas de usuario.

25.2 Equipamiento técnico

Al ser realizado solo por una persona tan solo será necesario un ordenador del que ya se predispone así que no será necesaria la inversión en el equipamiento técnico.

25.3 Costes – valor del proyecto

En este proyecto han intervenido básicamente tres tipos de perfiles:

- Diseñador web: encargado de crear los mookups y realizar la maquetación de la página.
- Programador web: Encargado de la gestión de y creación de CMS, así como de las personalizaciones de la plantilla.
- Director del proyecto: Encargado de coordinar a los dos perfiles anteriores y de realizar toda la documentación asociada al proyecto, así como su presentación.

Teniendo en cuenta esto se procede a calcular las horas de cada perfil y su coste asociado:

TFG			
	Precio por hora	Horas de trabajo	TOTAL
DESARROLLO			
JEFE DE PROYECTO	43.75 €	96	4.200 €
DISEÑADOR	25 €	32	800 €
PROGRAMADOR	25 €	64	1.600 €
TOTAL			6600€

26. Análisis de mercado

26.1 Estudio previo para la implantación de CSS Grid

Aunque finalmente se optó por la utilización de Simple Grid previamente se estuvieron valorando otras opciones para la implementación del *framework*, Estos fueron Flexbox Grid ¹⁰ y Bootstrap ¹¹.

Flexbox grid: Este *framework* permite utilizar diferentes tamaños de columna, alineación y distribución dentro del ancho ventana además las columnas y las filas se pueden distribuir en porcentajes. Destacar también que permite la auto alineación de los elementos.

Bootstrap: Es un kit de herramientas de código abierto que se utiliza para desarrollar con HTML CSS y JS, dispone de muchos componentes pre compilados como botones, estructuras de div's... así como un sistema de cuadrícula sensible.

Se optó por la utilización de Simple Grid debido a que era la más sencilla de integrar en la plantilla de WordPress ya que tanto Flexbox grid como Bootstrap requerían de una mayor integración para satisfacer la misma necesidad y con el mismo nivel de calidad.

26.2 Estudio previo para la implantación de PWA

La implantación de estas características se ha llevado a cabo mediante plugins, la más compleja es la del envío de notificaciones, por lo que se estudiaron la siguiente alternativa previa a la selección final:

Pushcrew ¹²: Permite el envío de notificaciones a múltiples plataformas de dispositivos, además la notificación puede tener más de una opción y se pueden enviar a una hora determinada.

Se descartó el uso de este *plugin* debido a que se trata de un complemento de pago por lo tanto se ha optado por la utilización de OneSignal que es gratuita.

¹⁰ <http://flexboxgrid.com/>

¹¹ <https://getbootstrap.com/>

¹² <https://pushcrew.com/>

27. Marketing y Ventas

Una vez finalizado el proyecto será entregado para su posible uso por parte de la universidad o por parte de algún consultor o profesor. La plantilla esta óptimamente preparada para ser utilizada en la asignatura “Lenguajes y estándares web” ya que esta era el objetivo del rediseño. Esto no elimina el hecho de que pueda ser utilizado como plantilla para otros blogs de otras asignaturas de los distintos grados de la UOC ya que esta sigue la línea visual de la universidad y además se monta sobre WordPress sin limitar su uso por que abre muchísimas posibilidades.

28. Conclusión/-es

En general pienso que el proyecto realizado ha sido una muy buena oportunidad de poner en práctica todo lo que he aprendido durante mis años de estudio en la UOC. De hecho, son muy pocas o ninguna las competencias adquiridas que no he utilizado desde programación o maquetación hasta creación de video para la presentación, organización y gestión de proyectos.

El proceso de trabajo ha resultado más sencillo conforme paso el tiempo, ya que al principio encontrarte ante un “papel en blanco” supone un reto y se tienen muchos frentes abiertos. Un buen estudio previo hizo posible seleccionar los *plugins* necesarios para dotar a la plantilla WordPress creada de las características de *Progressive Web app* (PWA) que actualmente están “de moda”. El diseño se adaptó al nuevo estilo de la Universat Oberta de Catalunya con el fin de permitir los blogs de las asignaturas y en particular a la asignatura “lenguajes y estándares web “.

Ha sido muy satisfactorio poder comprobar como lo diseñado en los *wireframes* se ha llevado casi de forma idéntica a la realidad tanto en versión móvil como en versión de escritorio. Respecto a las características PWA destacar por encima de todas la utilizad del envío de notificaciones tanto al móvil como al ordenador, que permite a los alumnos enterarse de las novedades en los artículos de una asignatura sin tener que acceder al campus o estar constantemente revisando el blog.

Anexo 1. Entregables del proyecto

Lista de archivos entregados y su descripción.

Carpeta TFG : Contiene la plantilla personalizada

Anexo 2. Código fuente (extractos)

Selección de partes relevantes del código fuente del servicio/aplicación creado, descripción y comentarios.

El siguiente if se creó para asignar automáticamente colores a las categorías creadas y de esta forma mostrarlas después la parte superior de cada card.

```
<!--condicional para añadir el color de la categoria en las cards-->
<?php if (in_category('css') ): $categorial = 'cat-css' ?>
<?php elseif (in_category('html') ): $categorial = 'cat-html' ?>
<?php elseif (in_category('accesibilidad') ): $categorial = 'cat-accessibility' ?>
<?php elseif (in_category('rendimiento') ): $categorial = 'cat-efficiency' ?>
<?php elseif (in_category('software') ): $categorial = 'cat-software' ?>
<?php endif; ?>
<!------>
```

Código JQuery que programa el evento que ocurre al pinchar sobre el icono “lupa”

```
/* EVENTOS AL PINCHAR EN EL BUSCADOR */
$("#search_button").click( function() {

    $("#buscador").toggle();

    if ((screen.width>=1600) && (screen.height>=900))
    {
        $("#nav_superior").toggle();
    }

});
```

Código JQuery que oculta el botón hamburguesa de la versión móvil al superar un determinado tamaño de pantalla:

```
//cambio de css en función del tamaño de la pantalla
$(window).resize(function(){
    if ((screen.width>=1600) && (screen.height>=900))
    {

        $("#main").css("display","block");
        $("#mobile-menu").css("display","none");
    }
});
```

Anexo 3. Librerías/Código externo utilizado

Se ha utilizado el framework simple CSS grid propiedad de Zach Cole con una licencia de código abierto. El siguiente código se ha añadido al fichero Styles.CSS de la plantilla de WordPress modificada en el proyecto.

```
/**
*** SIMPLE GRID
*** (C) ZACH COLE 2016
**/

@import url(https://fonts.googleapis.com/css?family=Lato:400,300,300italic,400italic,700,700italic);

/* UNIVERSAL */

html,
body {
  height: 100%;
  width: 100%;
  margin: 0;
  padding: 0;
  left: 0;
  top: 0;
  font-size: 100%;
}

/* ROOT FONT STYLES */

* {
  font-family: 'Lato', Helvetica, sans-serif;
  color: #333447;
  line-height: 1.5;
}

/* TYPOGRAPHY */

h1 {
  font-size: 2.5rem;
}
```

```
h2 {
  font-size: 2rem;
}

h3 {
  font-size: 1.375rem;
}

h4 {
  font-size: 1.125rem;
}

h5 {
  font-size: 1rem;
}

h6 {
  font-size: 0.875rem;
}

p {
  font-size: 1.125rem;
  font-weight: 200;
  line-height: 1.8;
}

.font-light {
  font-weight: 300;
}

.font-regular {
  font-weight: 400;
}

.font-heavy {
  font-weight: 700;
}

/* POSITIONING */
```

```
.left {  
  text-align: left;  
}
```

```
.right {  
  text-align: right;  
}
```

```
.center {  
  text-align: center;  
  margin-left: auto;  
  margin-right: auto;  
}
```

```
.justify {  
  text-align: justify;  
}
```

```
/* ===== GRID SYSTEM ===== */
```

```
.container {  
  width: 90%;  
  margin-left: auto;  
  margin-right: auto;  
}
```

```
.row {  
  position: relative;  
  width: 100%;  
}
```

```
.row [class^="col"] {  
  float: left;  
  margin: 0.5rem 2%;  
  min-height: 0.125rem;  
}
```

```
.col-1,  
.col-2,
```

```
.col-3,  
.col-4,  
.col-5,  
.col-6,  
.col-7,  
.col-8,  
.col-9,  
.col-10,  
.col-11,  
.col-12 {  
  width: 96%;  
}  
  
.col-1-sm {  
  width: 4.33%;  
}  
  
.col-2-sm {  
  width: 12.66%;  
}  
  
.col-3-sm {  
  width: 21%;  
}  
  
.col-4-sm {  
  width: 29.33%;  
}  
  
.col-5-sm {  
  width: 37.66%;  
}  
  
.col-6-sm {  
  width: 46%;  
}  
  
.col-7-sm {  
  width: 54.33%;  
}
```

```
.col-8-sm {  
  width: 62.66%;  
}
```

```
.col-9-sm {  
  width: 71%;  
}
```

```
.col-10-sm {  
  width: 79.33%;  
}
```

```
.col-11-sm {  
  width: 87.66%;  
}
```

```
.col-12-sm {  
  width: 96%;  
}
```

```
.row::after {  
  content: "";  
  display: table;  
  clear: both;  
}
```

```
.hidden-sm {  
  display: none;  
}
```

```
@media only screen and (min-width: 33.75em) { /* 540px */  
  .container {  
    width: 80%;  
  }  
}
```

```
@media only screen and (min-width: 45em) { /* 720px */  
  .col-1 {  
    width: 4.33%;  
  }  
}
```



```
}
```

```
.col-2 {  
  width: 12.66%;  
}
```

```
.col-3 {  
  width: 21%;  
}
```

```
.col-4 {  
  width: 29.33%;  
}
```

```
.col-5 {  
  width: 37.66%;  
}
```

```
.col-6 {  
  width: 46%;  
}
```

```
.col-7 {  
  width: 54.33%;  
}
```

```
.col-8 {  
  width: 62.66%;  
}
```

```
.col-9 {  
  width: 71%;  
}
```

```
.col-10 {  
  width: 79.33%;  
}
```

```
.col-11 {  
  width: 87.66%;
```

```
}  
  
.col-12 {  
  width: 96%;  
}  
  
.hidden-sm {  
  display: block;  
}  
}  
  
@media only screen and (min-width: 60em) { /* 960px */  
  .container {  
    width: 75%;  
    max-width: 60rem;  
  }  
}
```

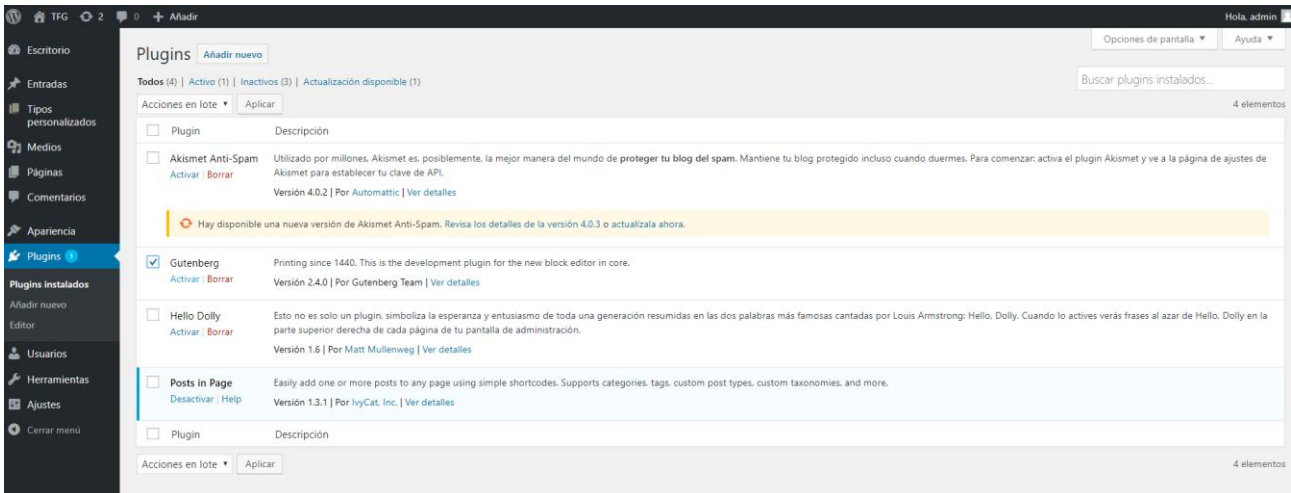
Para el menú “hamburguesa” de la versión móvil se ha utilizado como base el código de la siguiente pagina web:

<https://codepen.io/lbebbber/pen/LELBEO>

Anexo 4. Capturas de pantalla

Capturas de pantalla tanto del trabajo/servicio/aplicación realizado como del proceso de trabajo. Este anexo también puede utilizarse para recopilar las capturas mostradas en otras secciones, en mayor tamaño para su mejor visualización, o no ser necesario su uso por el tipo de trabajo realizado.

Instalación de plugin Guttenberg:



Programación en jQuery:

```
115  * You can remove this if you don't need it
116  */
117  loadGravatars();
118
119  /* EVENTOS MENU MOVIL OCULTAR TODO EL CONTENIDO SALVO EL NAV */
120  $("#hamburguesa").click( function() {
121    $("#nav#mobile-menu").toggle();
122    $("#main").toggle();
123
124    if ($(this).hasClass('checked')){
125      $("#label#hamburguesa").removeClass("checked");
126    }else{
127      $("#label#hamburguesa").addClass("checked");
128    }
129
130    /*$("#label#hamburguesa.checked").removeClass("checked");*/
131    /*$("#label#hamburguesa").addClass("checked");*/
132
133    /*$("#.checked").removeClass("checked");*/
134    $(this).addClass("checked");*/
135  });
136
137
138  //cambio de css en función del tamaño de la pantalla
139  $(window).resize(function(){
140    if ((screen.width>=1600) && (screen.height>=900))
141    {
142
143      $("#main").css("display","block");
144      $("#mobile-menu").css("display","none");
145    }
146  });
147
148  }); /* end of as page load scripts */
149
150
151
```

Anexo 5. Guía de usuario

El rediseño del blog se encuentra disponible en la siguiente web para su testeo:

<http://javiermurillotfg.xyz/>



1-Pulsando en el logo de la UOC o en el nombre del blog se navegará a la página principal.

2-Para acceder al contenido de una *card* en particular habrá que hacer clic en el nombre de la misma o en la leyenda “Leer más”.

3-Si se presiona en el icono de la lupa se desplegará el menú de buscar.

4-Para navegar por las distintas categorías bastara con seleccionarlás en el menú.

Anexo 6. Libro de estilo

- Logotipos.

Logo UOC:



Logotipo EIMT:



- Paleta de colores.
- Fondos

Utilización de fondo blanco configurado desde el menú de apariencia de WordPress.

- Botones

Icono lupa:



Icono menú hamburguesa:



Anexo 7. Glosario/Índice analítico

- PWA: Progressive web app
- WP: WordPress
- Mokups: Maqueta de una página web o app móvil.
- Wireframes: Maqueta de una página web o app móvil.
- Cards: Se trata de contenedores rectangulares o cuadrados, a la hora de ser programadas en html se realizarán mediante “divs”.

Anexo 8. Bibliografía

Israel Flores, (2017).Conceptos básicos del posicionamiento de rejilla :

https://developer.mozilla.org/es/docs/Web/CSS/CSS_Grid_Layout/Conceptos_B%C3%A1sicos_del_Posicionamiento_con_Rejillas

Manz. Grid CSS : <https://lenguajecss.com/p/css/propiedades/grid-css>

Gutenber Team, Gutenber plugin : <https://es.wordpress.org/plugins/gutenberg/#description>

César Córcoles ,Lenguajes y estándares web <http://multimedia.uoc.edu/blogs/htmlcss/#>

Laura Cuesta (2018) . ¿Qué son las PWA? : <https://www.tu-app.net/blog/progressive-web-app/>

Jose ;anuel Alarcón (2017), Qué son las Aplicaciones Web Progresivas o "Progressive Web Apps":

<https://www.campusmvp.es/recursos/post/Que-son-las-Aplicaciones-Web-Progresivas-o-Progressive-Web-Apps.aspx>

Iñaky Berzal (2017), Como acelerar WordPress: <http://iberzal.com/como-acelerar-wordpress-blog/>

Brian Jackson (2017), Cómo Configurar Notificaciones Push de WordPress Gratuitas:

<https://kinsta.com/es/blog/notificaciones-push-wordpress/>

Recursos vectoriales: <https://www.flaticon.com/>

Franciso Herrera, El blog como recurso educativo

http://roble.pntic.mec.es/jprp0006/didactica_internet/didactica_internet2/3_el_blog_como_recurso_educativo.html

Gian Oliveira ,(2018) Instalar WordPress en un servidor local [https://www.gianoliveira.com/instalar-](https://www.gianoliveira.com/instalar-wordpress-servidor-local.html)

[wordpress-servidor-local.html](https://www.gianoliveira.com/instalar-wordpress-servidor-local.html)

Anexo 9. Vita

Nací en Madrid en el año 1986 aunque he vivido la mayor parte de mi vida en un pueblo al norte de Madrid llamado Colmenar Viejo y soy el mediano de tres hermanos. Al finalizar el instituto comencé los estudios de técnico superior en Desarrollo de aplicaciones informáticas donde realicé las prácticas de empresa en la que hoy en día aun es mi actual empresa, han pasado ya casi 12 años. A día de hoy trabajo como administrador de software para ingeniería y DBA Oracle.

En el año 2011 comencé mis estudios en la UOC donde espero finalizar este año 2018 el grado Multimedia que estoy cursando.