



UNIVERSIDAD DE MÁLAGA



GRADO EN INGENIERÍA DEL SOFTWARE

Construcción de un sitio Web “hub” de Activos para
Proyectos Tecnológicos

Development of an Asset Website Hub for Technological
Projects

Realizado por
José Carlos Ruiz Díaz

Tutorizado por
Eduardo Medina Cano

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, septiembre 2021



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADUADO EN INGENIERÍA DEL SOFTWARE

**Construcción de un sitio Web “hub” de Activos para
Proyectos Tecnológicos**

**Development of an Asset Website Hub for Technological
Projects**

Realizado por
José Carlos Ruiz Díaz

Tutorizado por
Eduardo Medina Cano

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE DE 2021

Fecha defensa:



UNIVERSIDAD
DE MÁLAGA



Resumen

El Objetivo de este trabajo fin de grado es la construcción de un sitio Web que permita recoger una información amplia, categorizada, ordenada y útil de herramientas, tecnologías, lenguajes de programación, marcos de trabajo y software en general.

El sitio Web permite que los usuarios puedan registrar cualquiera de estos elementos, que denominaremos activos tecnológicos, así como su modificación y borrado, a través de un perfil de usuario específico denominado "EDITORACTIVOS" con permisos para realizar estas acciones.

Además, existirá un perfil de administrador capaz de configurar todos los aspectos relativos al sitio Web, como son la creación de nuevos usuarios, edición de los campos y tipos personalizados, publicación de nuevas páginas y contenidos de otro tipo.

Por último, el perfil usuario no registrado, que podrán consultar los activos, realizar búsquedas personalizadas, convirtiendo este sitio Web en un lugar de aprendizaje dónde poder comparar la inmensidad de activos de gran valor disponibles en la actualidad.

Palabras clave: Web, activos, recursos, tecnologías, buscador, aprendizaje, WordPress, PHP, JavaScript

Abstract

The aim of this final degree project is the implementation of a website that allows the collection of broad, categorized, orderly and useful information on tools, technologies, programming languages, frameworks and software in general.

The website allows users to register any of these elements, which we will call technological assets, as well as their modification and deletion, through a specific user profile called "EDITORACTIVOS" with permissions to perform these actions.

In addition, there will be an administrator profile capable of configuring all aspects of the website, such as creating new users, editing custom fields and types, publishing new pages and other types of content.

Finally, the unregistered user profile, who will be able to consult the assets, do custom searches, turning this website into a place of learning where you can compare the immensity of high-value assets available today.

Keywords: Web site, assets, resources, technologies, search engine, learning, WordPress, plugin, PHP, JavaScript

Índice

Resumen	1
Abstract	1
Índice	1
Introducción.....	1
1.1 Motivación	1
1.2 Objetivos.....	2
1.3 Estructura de la memoria	4
Tecnologías	7
2.1 Herramientas	8
2.1.1 WordPress.....	8
2.1.2 Laragon	10
2.1.3 Visual Studio Code	11
2.1.4 GitLab.....	12
2.1.5 Xdebug	13
2.1.6 Microsoft Teams	14
2.2 Entorno de desarrollo.....	15
2.2.1 Instalación y configuración del servidor Web.....	15
2.2.2 Instalación y configuración de WordPress.....	18
2.2.3 Configuración de Visual Studio Code	19
Metodología	21
3.1 Planificación.....	22
3.1.1 Sprint 0: Identificar la solución tecnológica.....	22

3.1.2 Sprint 1: Montaje inicial de la Web y configuración de usuarios	23
3.1.3 Sprint 2: Desarrollo de la funcionalidad registro de activos	23
3.1.4 Sprint 3: Diseño del interfaz de usuario	23
3.1.5 Sprint 4: Implementar funcionalidad para mostrar un activo.....	23
3.1.6 Sprint 5: Implementar funcionalidad de búsqueda de activos	24
3.1.7 Sprint 6: Incorporar filtros a la búsqueda de activos	24
3.1.8 Sprint 7: Implementar paginación de resultados de búsqueda	24
3.1.9 Sprint 8: Pruebas, detección y corrección de errores	25
3.1.10 Sprint 9: Elaboración de la memoria	25
3.2 Gestión	25
Especificación de requisitos	3
4.1 Comparativa con otros sitios Web	3
4.2 Tipos de usuario de la Web	6
4.3 Funcionalidades	7
4.3.1 Creación de un nuevo activo.....	7
4.3.2 Modificación de un activo.....	17
4.3.3 Borrado de un activo.....	18
4.3.4 Consulta de un activo.....	18
4.3.5 Búsqueda de activos	19
4.3.6 Creación nuevos lenguajes de programación	21
4.3.7 Creación nuevos sistemas operativos	21
4.4 Diagrama de casos de uso	22
Solución técnica.....	23
5.1 Análisis de las alternativas	23
5.2 Estructura de la información	28
5.3 Diseño técnico.....	31
Implementación	43

6.1 Desarrollo	43
6.2 Introducción de contenidos	55
Pruebas	59
7.1 Acceso a la Web	60
7.1.1 Acceso a web pública	60
7.1.2 Acceso a panel de administración	60
7.2 Creación de un activo	61
7.3 Ver un activo	62
7.4 Modificar un activo	63
7.5 Borrar un activo	65
7.6 Lenguajes programación.....	66
7.6.1 Alta de un lenguaje de programación	66
7.6.2 Edición de un lenguaje de programación.....	67
7.6.3 Borrar de un lenguaje de programación	68
7.7 Sistemas Operativos	69
7.7.1 Alta de un sistema operativo	69
7.7.2 Edición de un sistema operativo	70
7.7.3 Borrar de un sistema operativo	70
7.8 Visualizar páginas del sitio Web.....	71
7.8.1 Visualizar la página principal.....	71
7.8.2 Visualizar entrada del blog.....	72
7.8.3 Enlaces banda lateral derecha del blog	74
7.9 Realizar búsquedas de activos	75
7.9.1 Buscar todos los activos.....	75
7.9.2 Buscar activos filtrados por el campo qué es.....	75
7.9.3 Buscar activos filtrados por los campos qué es y precio.....	76
7.9.4 Buscar activos filtrados por el campo tipo de activo	77

7.9.5 Buscar activos filtrados por el campo tipo de proyecto.....	78
7.9.6 Buscar activos filtrados por el campo perfil profesional.....	79
7.9.7 Buscar activos filtrados por el campo actividad de proyecto	79
7.9.8 Buscar activos filtrados por el campo área de proceso CMMI.....	80
7.9.9 Buscar activos filtrados por el campo objetivos específicos CMMI	80
7.9.10 Buscar activos filtrados por el campo proceso PMBOK	81
7.9.11 Buscar activos filtrados por el campo gestión del servicio TI.....	82
7.10 Ver el contenido de un activo	82
7.10.1 Ver la ventana principal del activo.....	82
7.10.2 Ver la pestaña caracterización de un activo	83
7.10.3 Ver la pestaña comparable con de un activo.....	84
7.10.4 Ver la pestaña estándares de un activo	85
7.10.5 Ver la pestaña aprendizaje de un activo	86
7.11 Pruebas realizadas	87
Conclusiones.....	95
Bibliografía	97
Manual de Instalación	101
Requerimientos:	101
Instalación en servidor Web.....	101
Paso 1: Despliegue	101
Paso 2: Instalar la base de datos.....	103
Paso 3: Actulizar los datos	104
Paso 4: Acceder al panel de administrador.....	105

1

Introducción

1.1 Motivación

Desde hace años y de forma creciente, están surgiendo diferentes herramientas, tecnologías, lenguajes, marcos de trabajo, sistemas, productos, software en general, etc. para ser utilizados en proyectos tecnológicos (todos estos elementos los englobaremos bajo el término "activo", haciendo uso de su significado: "algo que aporta valor"). Principalmente estos activos son muy útiles en proyectos de desarrollo de software, pero también en proyectos de gestión de servicios en el ámbito de las tecnologías de la información y de las comunicaciones. El panorama es, por una parte, confuso, puesto que es difícil llegar a conocer todas las alternativas existentes, para qué sirven, dónde se pueden usar, etc. incluso entender las diferencias y beneficios que cada una de ellas puede aportar.

Por otra parte, es complicado estar al día y seguir los nuevos activos que aparecen cada año.

Sería de gran utilidad poder disponer de un sitio que ofrezca información sobre estos activos, de una manera amplia, organizada y estructurada. Este sitio Web lo llamaremos un "hub" de activos tecnológicos, utilizando un nombre que se usa actualmente con un significado similar al que deseamos tener con este proyecto (hub.docker.com, github, etc.). Dicho sitio Web se desea utilizar como recurso docente en asignaturas de los grados informáticos, para facilitar que el alumno disponga de una visión profunda de todos estos activos, de forma que se incorporen al mercado de trabajo

con un mayor conocimiento de las alternativas tecnológicas existentes.

Por otra parte, este sitio será en sí mismo útil para profesionales, dado que la información que proporciona es muy valiosa en el ámbito de la empresa, de cara a tomar decisiones y considerar el uso de diferentes activos en sus procesos.



Figura 1.1 Imagen del sitio Web <https://tiactivos.com/>

1.2 Objetivos

El objeto del trabajo es por tanto la construcción de un sitio Web que permita gestionar un *hub* de activos para proyectos tecnológicos. Será clave realizar un buen modelo conceptual de la BD, pues la riqueza de la información estará en proporcionar los campos apropiados, y en la clasificación múltiple en las que un activo debe ser registrado. De manera que esto facilite la búsqueda, la comparativa, y en definitiva su comprensión y su uso. Los activos quieren además relacionarse con estándares en el ámbito de las tecnologías de la información y comunicación (PMBOK¹, CMMI²,

¹ Project Management Body of Knowledge

² Capability Maturity Model Integration

ISO2000³, etc.), lo que además aportará una utilidad adicional para el entorno empresarial.

El objetivo del proyecto no es la introducción de la información de un alto volumen de activos, sino la de preparar el sitio Web, para que puede alojar dichos activos, con los requisitos que se desean. Lógicamente deberá tener los contenidos suficientes para poder mostrar el resultado final del trabajo fin de grado.

La solución tecnológica que se desea utilizar será WordPress.

En el trabajo se aplicará con rigor las actividades de la ingeniería del software.

Con todo ello, los objetivos que se persiguen con este trabajo fin de grado son:

- Diseñar, desarrollar y proporcionar un sitio Web "hub" con información de activos para el desarrollo de proyectos tecnológicos, con la finalidad de:
 - o Ser usado como recurso docente para que los alumnos de grados de informática tengan una visión lo más profunda y completa posible del universo de activos existentes.
 - o Pueda ser utilizado por empresas para facilitar el acceso a esta información, con la cual tomar decisiones de cara a mejorar sus propios procesos en proyectos de tecnología.

Las funcionalidades básicas que el sitio Web debe ofrecer son las siguientes:

- Creación, lectura, modificación y borrado de activos (CRUD). El activo deberá ser un elemento con un número alto de campos para poder registrar toda la información útil que debe contener.
- Visualización de un activo. Dado que se desea guardar bastante información asociada a un activo, se deberán implementar distintas vistas

³ Estándar internacional en gestión de servicios de TI ([Tecnologías de la Información](#))

- Búsqueda de activos. Deberá implementarse un potente motor de búsqueda, con diferentes filtros y condiciones.
- Funcionalidades de Administración del sitio Web.

1.3 Estructura de la memoria

La presente memoria se estructura en diez capítulos, cada uno de los cuales está dividido en subcapítulos donde se abordan distintos aspectos relacionados con el proyecto de forma organizada.

En su **primer capítulo**, este documento recoge la idea inicial que sirve como punto de partida para este trabajo, la necesidad de un sitio Web que sea el punto de encuentro de los diferentes activos tecnológicos, que permita buscar, encontrar y evaluar estos activos. También recoge la investigación de otros sitios web y su análisis en profundidad para comprobar qué información ofrecen y qué carencias presentan, para finalmente establecer cuáles son los objetivos a alcanzar con nuestro proyecto y definir un plan de trabajo para alcanzarlos.

A continuación, , en el **capítulo dos**, recogemos las tecnologías utilizadas en la realización del proyecto, enumerando las opciones analizadas para abordar cada una de las necesidades planteadas para la realización del trabajo, así como la motivación que nos ha llevado a la elección de unas determinadas herramientas en detrimento de otras opciones. Definimos las características del conjunto de procedimientos y herramientas que se utilizado para desarrollar el proyecto, es decir el entorno de desarrollo. Documentando su instalación y configuración a fin de poder replicar el entorno de desarrollo utilizado para escribir, generar, probar y depurar el sitio Web.

En el **capítulo tres**, se aborda la metodología aplicada a lo largo del proceso, recogiendo en primer lugar la planificación del trabajo y en segundo lugar la gestión del mismo.

En el **capítulo cuatro**, se recoge la especificación de los requisitos. Para ello, en primer lugar hemos analizado otros sitios Web existentes con el objetivo de determinar funcionalidades que podemos incorporar, así como las carencias que presentan con respecto el resultado deseado para nuestro proyecto. A continuación, se han definido los diferentes tipos de usuarios que deben poder acceder a la Web, así como los privilegios necesarios para cada uno de

ellos en base a su rol. Por último se han especificado las funcionalidades que debe ofrecer el sitio Web.

A lo largo del **capítulo cinco**, se explica como se ha orquestado la solución técnica. Ha sido necesario en primer lugar llevar a cabo un análisis de diversas alternativas, para a continuación, determinar la estructura de la información y finalmente realizar el diseño técnico del proyecto.

El **capítulo seis**, está dedicado a la implementación de la solución descrita en el capítulo anterior. Se recogen todas las fases de desarrollo llevadas a cabo para implementar los requisitos funcionales, abordando las dificultades que ha aparecido en cada una de las fases de desarrollo y cómo se han solventado las mismas, quedando documentado todo el proceso.

En el **capítulo siete**, se identifican y describen los casos de pruebas en relación a los casos de uso descritos.

En el **capítulo ocho**, se exponen las conclusiones obtenidas durante el desarrollo del trabajo, identificando los objetivos conseguidos y recogiendo posibles líneas futuras de trabajo.

Finalmente, se relacionan las referencias consultadas para la realización del trabajo, así como las fuentes citadas en la presente memoria.

2

Tecnologías

Para la construcción del sitio Web hemos optado por usar como tecnología base WordPress, un sistema de gestión de contenidos enfocado a la creación de páginas Web, cuyo uso está ampliamente extendido y que responde a la premisa de tratarse de un software libre.

Este gestor de contenidos permite la instalación de numerosos complementos denominados "*plugins*" que permiten implementar funcionalidades en el sitio web basado en este marco de trabajo, así como temas que facilitan la creación de páginas web y el trabajo de diseño de las mismas.

Gran parte del trabajo realizado consiste en el estudio, análisis y elección de la mejor combinación de estos elementos con la intención de conseguir el objetivo planteado inicialmente.

Para el entorno de desarrollo se ha optado por utilizar **Laragon** una herramienta gratuita, sencilla rápida que permite configurar entornos de desarrollo basados en el servidor Web Apache, lenguaje de programación PHP y base de datos MySQL. Se ha configurado la extensión PHP Xdebug que permite la depuración de código y errores.

Se ha incorporado al entorno de desarrollo el uso de tecnologías y herramientas **DevOps** (desarrollo y operaciones) como GitLab para la gestión de repositorios y el control de versiones, permitiendo la integración y el despliegue de las diferentes versiones de la aplicación. Adicionalmente se ha utilizado la funcionalidad de gestión de peticiones y tareas para el control del desarrollo y coordinación con el tutor del trabajo.

Como editor de código fuente se ha optado por la herramienta Microsoft Visual Studio Code, una herramienta gratuita y disponible para Windows, Linux y MacOS, que incluye depuración, control integrado con Git y resaltado de sintaxis entre otras características.

Para las reuniones y tutorías, que debido a la situación actual y restricciones de movilidad causadas por la pandemia COVID19 se han realizado de forma no presencial, se ha utilizado la herramienta Microsoft Teams, que permite la reuniones de video, dispone de chat y permite compartir pantalla. Además dispone de un sistema de alojamiento de fichero en la nube, permitiendo organizar gracias a esta utilidad los ficheros y editarlos de forma distribuida.

2.1 Herramientas

Para el desarrollo del presente trabajo fin de grado ha sido necesario el uso de diversas herramientas y tecnologías con el fin de alcanzar el objetivo propuesto al inicio del mismo.

La elección de esas herramientas se ha realizado en base a un estudio y análisis de las opciones existentes en la actualidad en el mercado, así como de las características de cada una de ellas, tratando de elegir las más adecuadas a las necesidades de nuestro proyecto.

Se han tomado en consideración aspectos tales como, la popularidad de la tecnología en el sector, su fiabilidad, facilidad de integración, la existencia de un mantenimiento por parte de los responsables de la herramienta, mediante la publicación de actualizaciones y por último se ha intentado basar todo el trabajo en herramientas que no supongan un coste en licencias y que por tanto se base en software libre.

2.1.1 WordPress

Para llevar cabo la realización del trabajo fin de grado propuesto ha sido necesario la elección de un soporte que nos permita desarrollar el sitio web www.tiactivos.com para poder crear y gestionar los activos tecnológicos conforme a las necesidades de almacenamiento, presentación y mantenimiento de los mismos.

Según consta en la Web www.wordpress.com a fecha de hoy el 42% de la "Web" se ha creado con WordPress, y según las estadísticas del sitio [1] más del 60% de los sitios Web creados con un gestor de contenidos Web se basan en WordPress.

Esto unido al número de complementos (*plugins*) desarrollados por programadores disponibles para poder ampliar las funcionalidades originales ofrecidas por este gestor de contenidos Web.

WordPress dispone también de un amplio número de temas disponibles que permiten ajustar el diseño del sitio Web dando un aspecto personalizado y fácilmente ajustable a la funcionalidad que se desea implementar en el sitio Web. Este factor unido al anterior implica una mayor extensibilidad que las que presentan otras alternativas como Joomla, o Drupal.

WordPress permite una instalación sencilla, dispone de actualizaciones periódicas que garantizan mantener la seguridad del sistema.

Otra de las características que incorpora es la posibilidad de generar URLs optimizadas para para posicionamiento SEO.

Adicionalmente, al disponer de un diseño adaptativo incorporado nos garantizará tener una visualización correcta del sitio Web desde ordenadores, tabletas y móviles.

Al tratarse de un software libre liberado sobre licencia GPLv2 no conlleva costes para su instalación e implantación, siendo uno de los factores a valorar positivamente para elegir esta herramienta.

Los requisitos necesarios para la instalación de WordPress son:

- PHP versión 7.4 o superior.
- MySQL versión 5.6 o superior o MariaDB versión 10.1 o superior.
- Se recomienda Apache o Nginx, al ser los servidores más robustos y flexibles para ejecutar WordPress, pero cualquier servidor que sea compatible con PHP y MySQL funcionará.

En conclusión, aunque hubiera sido factible el uso de otro gestor de contenidos Web, las características descritas anteriormente, unidas a que WordPress es la solución usada a nivel empresarial por la empresa donde trabaja el tutor del proyecto, dotándole de conocimientos y experiencia en el desarrollo de soluciones Web basadas en WordPress, han sido determinantes para que sea la opción elegida para implementar este trabajo fin de grado.

2.1.2 Laragon

Para configurar un entorno de desarrollo adecuado a los requisitos exigidos por WordPress, se han evaluado diferentes alternativas existentes que permiten configurar un espacio local de trabajo con soporte para PHP y MySQL en el que poder trabajar de forma local en el proyecto.

XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X, Apache, MariaDB/MySQL, PHP, Perl.

Sin duda es una de las opciones más extendidas para desarrollar con PHP, estando disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

WampServer es un paquete software para el sistema operativo Microsoft Windows, que consiste en el servidor web Apache, OpenSSL para soporte SSL, base de datos MySQL y lenguaje de programación PHP.

Docker ofrece la posibilidad de crear entornos de desarrollo con independencia del sistema operativo, al contrario que XAMPP y WampServer, ofreciendo todo lo necesario para poder configurar el servidor Web Apache, PHP y MySQL.

Laragon es una herramienta que permite crear bajo el sistema operativo Windows diferentes entornos de desarrollo con el conjunto de programas necesarios para desarrollar aplicaciones. Sirve para trabajar con PHP, pero también con otros lenguajes del lado del servidor, como Node, Python o Ruby. [2]

Algunas de las características de hacen muy interesante esta herramienta son la sencillez y rapidez con la que se pueden crear los entorno de trabajo, donde por ejemplo podemos crear Virtualhost de una forma automática.

Otra de las características que incorpora Laragon es que facilita el cambio, en caso de ser necesario, de las

versiones de Apache o MySQL / MariaDB.

Se trata de un entorno de desarrollo muy liviano que no requiere grandes prestaciones en el equipo donde se ejecuta.

Además y como todas las soluciones evaluadas para el entorno de desarrollo Laragon también es gratuito.

En conclusión, Laragon se postula como el entorno de desarrollo elegido para este trabajo porque resulta más fácil y automático de configurar que Xampp, WampServer y Docker, además ofrece un resultado de rendimiento mejor y menor dificultad para definir un entorno seguro de desarrollo con certificado SSL. Con Xampp requiere bastante más trabajo, y con Docker hay que usar alguna solución para facilitar crear un certificado SSL para cada instalación.

Por último, el Laragon resulta más fácil y rápida de configurar la extensión PHP Xdebug para la depuración del código.

2.1.3 Visual Studio Code

Visual Studio Code es un editor de código fuente multiplataforma desarrollado por Microsoft para el desarrollo con PHP, incorporando entre sus características soporte para la depuración, control integrado versiones, resaltado de sintaxis, concordancia de llaves en expresiones.

Sin embargo, una de sus principales ventajas frente a otros entornos integrados de desarrollo como PHPStorm o Eclipse es que podemos ampliar sus funcionalidades con las numerosas extensiones que encontraremos en la tienda de Visual Studio Code. [3]

Entre las características que presenta este editor se encuentra el resaltado de la sintaxis en colores, las marcas de errores y advertencias, concordancia de llaves, visualizador de estructura, búsqueda y remplazo, permite refactorizar código, integración con Git para la gestión de repositorios y depuración de código gracias a la extensión Xdebug.

PhpStorm es un entorno de desarrollo integrado o IDE especializado en lenguaje PHP, HTML y JavaScript. Presenta compatibilidad con los principales marcos de trabajo como son Symfony, Laravel, Drupal, WordPress y otros marcos de trabajo. Es una herramienta profesional muy potente, cuya licencia tiene un coste anual.

Se ha elegido Visual Studio Code como la herramienta de programación para la realización del presente trabajo fin de grado, principalmente por el aprendizaje que aporta al alumno en el uso de una herramienta cada vez más extendida entre los programadores a nivel profesional en los últimos tiempos, debido a su productividad, interfaz amigable, potencia de desarrollo y disponibilidad para distintos

sistemas operativos (Windows, Linux y MacOS)), así como, por estar disponible como software libre.

2.1.4 GitLab

Git es una herramienta de software libre creada por Linus Torvalds, para el control de versiones. Los controladores de versiones son aplicaciones que tienen como objetivo facilitar el proceso de desarrollo de software, particularmente en entornos donde intervienen múltiples programadores sobre proyectos relativamente complejos.

Para ello, estas herramientas realizan un seguimiento de todos los cambios realizados sobre los archivos de código fuente a lo largo del tiempo. Gracias a ello, cuando surgen errores, es mucho más sencillo comparar las diferentes versiones del software y volver a una versión anterior si es necesario. [4]

GitLab nació como un sistema de alojamiento de repositorios Git, es decir, un hosting para proyectos gestionados por el sistema de versiones Git. Sin embargo, alrededor de esta herramienta han surgido muchas otras herramientas muy interesantes para programadores y equipos de desarrollo, que envuelven todo el flujo del desarrollo y el despliegue de aplicaciones, test, etc.

La principal diferencia entre GitLab y sus competidores es que GitLab se ofrece como un software libre que puedes descargar e instalar en cualquier servidor. Esta posibilidad permite usar GitLab para una empresa, profesional u organización en sus propios servidores, sin ningún coste adicional. La otra alternativa es usar GitLab directamente de GitLab.com, que se ofrece sin necesidad de invertir tiempo en configuración y aprovechando sus ventajas desde el primer minuto y sin coste para publicar repositorios de software libre, igual que su competidor GitHub, aunque GitLab dispone de algunos servicios extra. [5]

Un proyecto en GitLab corresponde con un repositorio Git, que será gestionado por unos usuarios autorizados y con permisos de escritura sobre el repositorio.

GitLab abarca todo el ciclo de vida DevOps, desarrollo y operaciones, metodología de desarrollo de software que se desea aplicar a lo largo de este trabajo y que se en la comunicación, colaboración e integración entre desarrolladores de software y los profesionales de sistemas en las tecnologías de la información. Su objetivo es ayudar a una organización a producir productos y servicios software más rápidamente, de mejor calidad y a un coste menor. Las

empresas con entregas muy frecuentes podrían requerir conocimientos de DevOps. [6]

La gestión de repositorios GitLab dispone de un mecanismo para dar soporte a la creación y gestión de ramas dentro de un proyecto. Esta funcionalidad de los repositorios Git nos permite crear ramificaciones de desarrollo sin tener que seguir la rama principal. Las ramas se pueden crear y unir entre sí de forma fácil y rápida. [7]

Las ramas son espacios independientes de desarrollo, para que distintos equipos de programadores puedan trabajar de forma simultánea sobre distintos puntos o funcionalidades de un mismo proyecto, conservando inalterados los archivos originales del proyecto, ofreciendo flexibilidad en el desarrollo de forma organizada.

Finalmente las ramas de desarrollo independientes se pueden fusionar de forma que el trabajo realizado en diferentes funcionalidades del proyecto se pueden integrar de forma segura y rápida, permitiendo solucionar los conflictos en código que se hayan podido producir durante el trabajo en cada una de las ramas.

GitLab permite la planificación de los proyectos mediante tareas (*issues*) y su organización mediante tableros (*boards*) e hitos.

Entre los motivos que nos han llevado a elegir esta herramienta para la gestión de repositorios están precisamente la organización y el seguimiento de las tareas de desarrollo, registrar y documentar los cambios realizados durante las distintas fases de desarrollo o *sprints* y facilitar las tareas de mantenimiento y corrección de errores del proyecto.

2.1.5 Xdebug

Xdebug es una extensión para PHP que proporciona una serie de características adicionales de depuración de código y errores, encaminadas a mejorar el desarrollo en PHP. Esta extensión mejora la depuración tradicional del lenguaje de programación PHP que utiliza únicamente las funciones `var_dump()` y `print_r()`.

Xdebug permite analizar el contenido de las variables y también realizar el seguimiento del flujo de ejecución, pudiendo introducir puntos de ruptura en el código, ver las variables en ese punto y trazar la ejecución de la programación en PHP.

Entre las características de esta extensión se encuentra la posibilidad de realizar un rastreo de las llamadas a cada función, argumentos, ubicación de la invocación en el disco y opcionalmente incluye cada asignación de variable y valor de retorno para cada función.

El objetivo final es integrar en el editor o entorno integrado de desarrollo Xdebug en sus herramientas de depuración, siendo compatible con PHPstorm, Netbeans y Visual Studio Code entre otros.

2.1.6 Microsoft Teams

Microsoft Teams es una aplicación de colaboración que ayuda a los equipos de personas a organizarse y mantener conversaciones centralizadas, todo en el mismo lugar, con Teams tendremos la posibilidad de crear equipos de trabajo, de gestionar proyectos, tareas y contenido con el resto del equipo. Dispone de áreas y funcionalidades para compartir archivos, desarrollar conversaciones o mensajería, chats individuales, videoconferencias, etc.

Con Teams tendremos la posibilidad de crear equipos de trabajo, de gestionar proyectos, tareas y contenido con el resto del equipo. Dispone de áreas y funcionalidades para compartir archivos, desarrollar conversaciones o mensajería, chats individuales, videoconferencias, etc.

Microsoft Teams es un *hub* digital que reúne conversaciones, reuniones, archivos y aplicaciones en un solo lugar. Debido a que se basa en Office 365, las escuelas se benefician de la integración con sus aplicaciones y servicios Office. Ofrece seguridad y cumplimiento de nivel empresarial que es extensible y personalizable. [8]

Una de las funcionalidades que incorpora Teams es la posibilidad de compartir pantalla durante una reunión, gracias a la cual es posible mostrar en pantalla el escritorio o cualquier aplicación del mismo a modo de presentación, resultando muy útil para reforzar la explicación verbal.

Debido a que esta aplicación se encuentra incluida en Office 365 y en virtud al acuerdo entre la Universidad de Málaga y Microsoft, está disponible para su uso por parte de los alumnos y el profesorado sin coste adicional en licencias. Esto unido a que tanto el tutor del trabajo, como el alumno están familiarizados con el manejo de la aplicación, la convierten en la opción elegida para realizar las tutorías de forma telemática.

2.2 Entorno de desarrollo

Para poder trabajar en el proyecto una de las primeras tareas que debemos enfrentar en la configuración de un entorno de desarrollo local compatible con las características y requisitos de nuestro proyecto.

Las principales ventajas a la hora de utilizar un entorno local de desarrollo son la capacidad de poder trabajar sin necesidad de disponer de conexión a internet, la velocidad a la hora de guardar y ejecutar cambios y por último la posibilidad de realizar desarrollos sin que se vea afectado el entorno de producción.

El desarrollo para este proyecto se basará en gestor de contenidos Web WordPress, por lo que debemos configurar en nuestro equipo local un servidor web Apache con certificado de seguridad SSL, interprete de para lenguaje PHP y base de datos MySQL.

El proceso de configuración del entorno de desarrollo lo hemos organizado en distintas fases o tareas que se detallan a continuación.

2.2.1 Instalación y configuración del servidor Web

Para este proceso se ha optado por la descarga del instalador Laragon Full para Windows 64 bits⁴. La instalación finaliza dejando configurados en nuestro equipo local los siguientes productos y versiones: Apache 2.4, Nginx, MySQL 5.7, PHP 7.4, Redis, Memcached, Node.js 14, npm y git.

El siguiente paso es la configuración de nuestro archivo hosts. El archivo hosts se usa por el sistema operativo para guardar la correspondencia entre los nombres de dominios y direcciones IP. Este es uno de los diferentes métodos que usa el sistema operativo para resolver nombres de dominio.

La creación de *hosts* virtuales (*virtualhosts*) permite a un mismo ordenador resolver múltiple nombres de dominio. Los *host* virtuales son útiles en diversos escenarios, en nuestro caso, lo usaremos para configurar nuestro entorno de desarrollo local, de forma que el servidor Web Apache resuelva ciertas direcciones a través de una dirección IP local.

⁴ <https://laragon.org/download/index.html>

Para ello, necesitaremos editar el archivo "hosts", que en el sistema operativo Windows se encuentra en la siguiente ruta "C:\Windows\System32\drivers\etc\hosts" para mapear los nombres de los hosts con las direcciones IP. Es muy importante que el nombre de dominio a usar no exista a fin de evitar colisiones. Para nuestro entorno de desarrollo vamos a usar el dominio "tiactivos.local". Por este motivo incluiremos una línea en el archivo con la siguiente configuración:

127.0.0.1 tiactivos.local

Para configurar la extensión Xdebug en Laragon, necesitaremos la información correspondiente al resultado de la función phpinfo() de PHP, que muestra gran cantidad de información sobre el estado actual de PHP. Incluyendo información sobre las opciones de compilación y extensiones de PHP, versión de PHP, información del servidor y entorno (si se compiló como módulo), entorno PHP, versión del OS, rutas, valor de las opciones de configuración locales y generales, cabeceras HTTP y licencia de PHP. Este resultado podemos obtenerlo de dos formas distintas, la primera es creando un fichero PHP en nuestro servidor que llame a la función phpinfo() y luego invocarlo desde el navegador Web y el segundo es llamando a nuestro servidor local con la siguiente URL: <http://localhost/?q=info>

Usaremos el asistente para instalar Xdebug, disponible en la URL <https://xdebug.org/wizard.php> y siguiendo las instrucciones pegaremos el resultado de invocar la función phpinfo() y analizaremos la salida ofrecida. El propio asistente de configuración nos facilita las instrucciones a seguir para poder configurar la extensión.

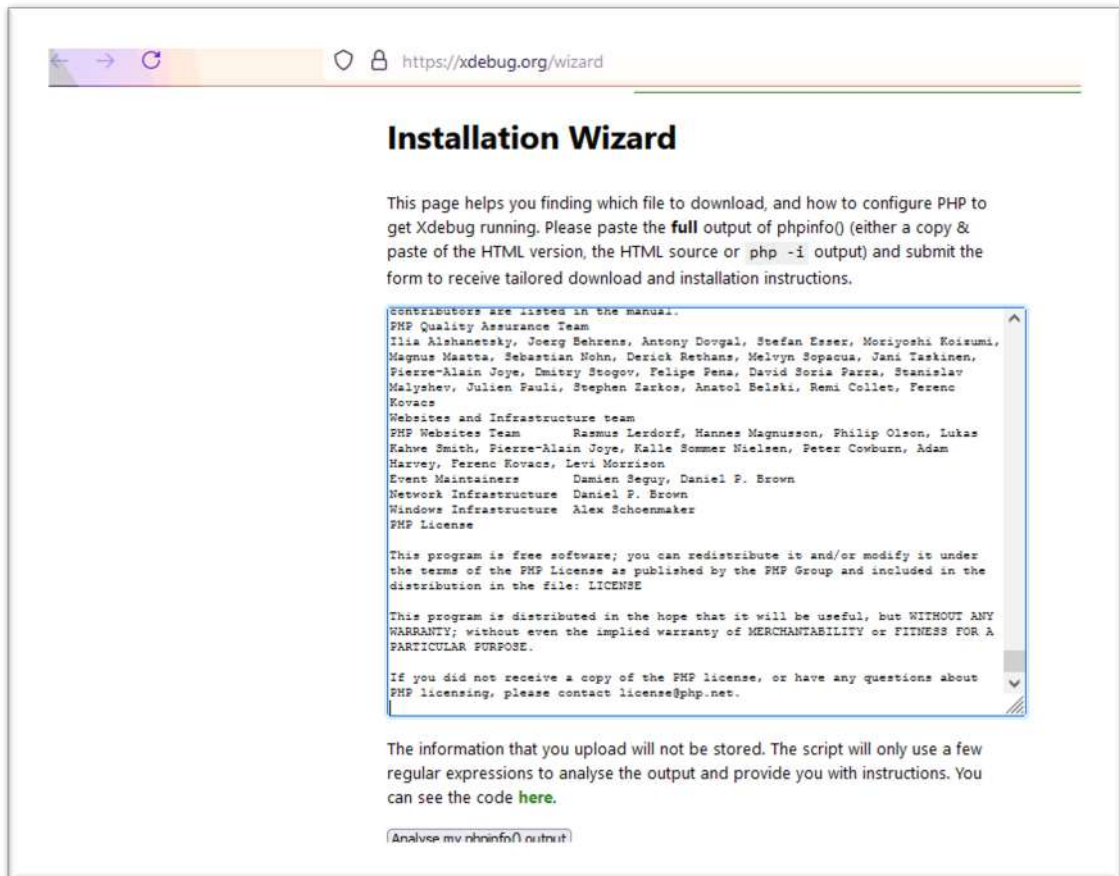


Figura 2.1 Xdebug Installation Wizard

Finalmente y como resultado es necesario modificar el fichero "php.ini" debiendo quedar configurado como se muestra a continuación:

```
zend_extension = C:\laragon\bin\php\php-7.2.19-Win32-VC15-x64\ext\php_xdebug-3.0.3-7.2-vc15-x86_64.dll
```

```
xdebug.remote_enable=1
```

```
xdebug.remote_host = localhost
```

```
xdebug.remote_port = 9003
```

```
xdebug.remote_connect_back = 1
```

```
xdebug.remote_autostart=1
```

```
xdebug.mode = debug
```

```
xdebug.start_with_request = yes
```

2.2.2 Instalación y configuración de WordPress

Una vez configurado nuestro servidor Web local correctamente, es necesario realizar una instalación del sistema de gestión de contenidos WordPress. Para este trabajo se ha, descargado el paquete de instalación disponible en la Web:

<https://es.wordpress.org/download/#download-install>

Tras descargar el fichero, será necesario descomprimir todos los archivos in carpetas en el directorio usado por el servidor Apache como como documento raíz.

Antes de comenzar el despliegue de la instalación, es necesario crear en el sistema de gestión de base de datos MySQL la base de datos que será usada por el instalador de WordPress. A continuación, ya podemos acceder desde el navegador Web usando la host virtual que hemos creado al script de instalación.

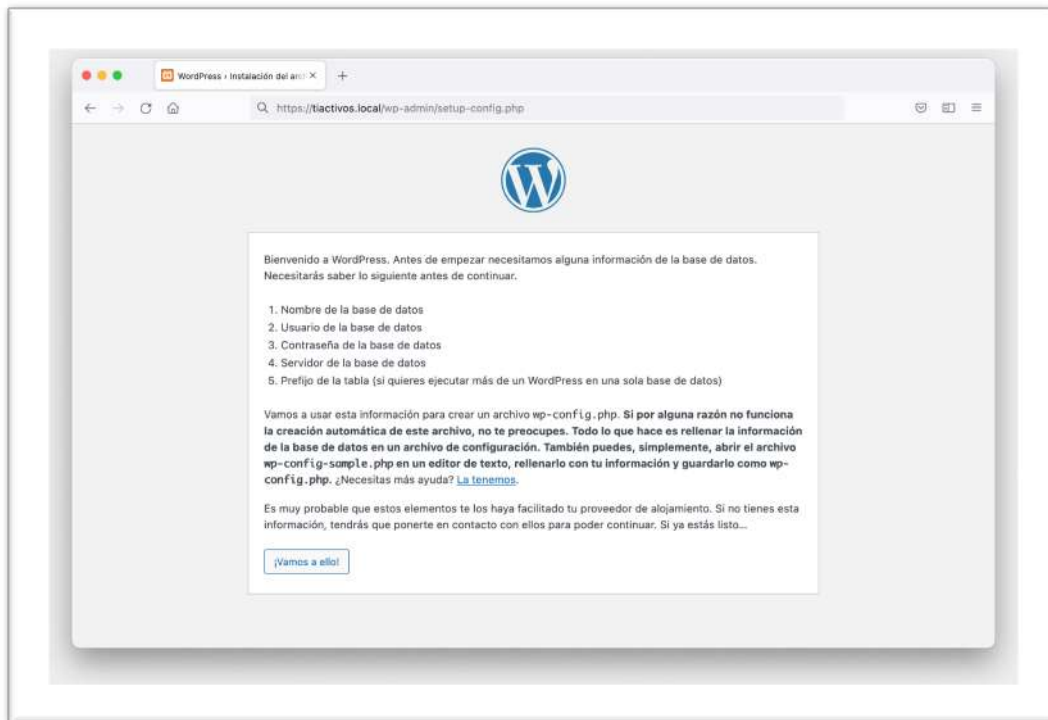


Figura 2.2 Ventana de instalación de WordPress

Tras seguir los pasos de script de instalación de WordPress, elegir un usuario administrador y establecer un contraseña de acceso, tendremos correctamente configurado y totalmente operativo nuestro entorno de desarrollo local.

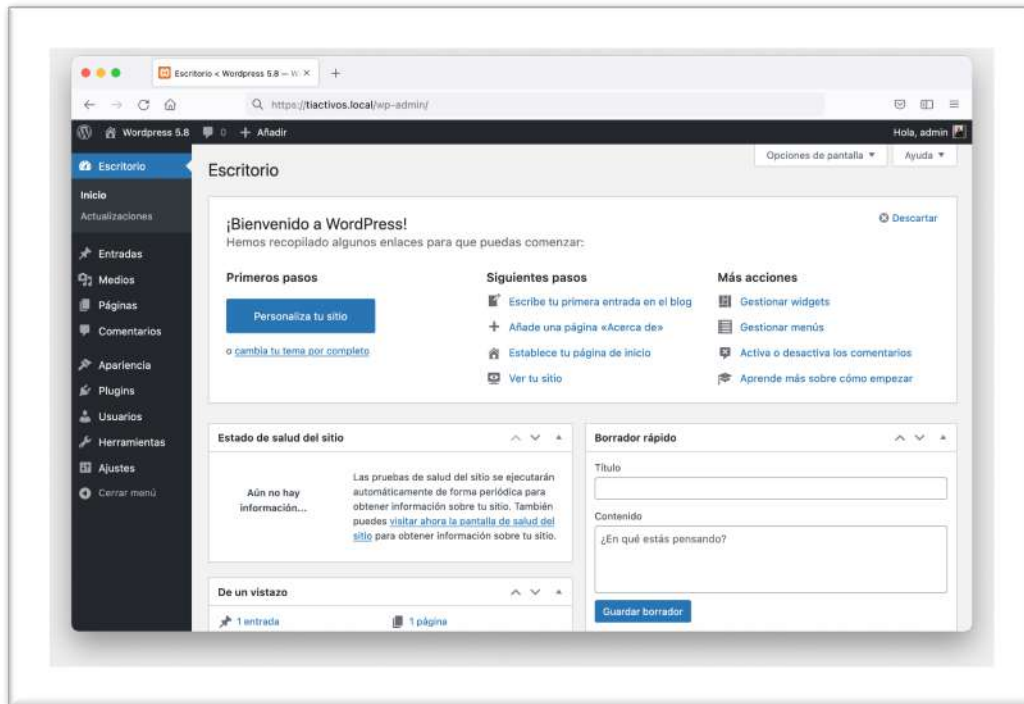


Figura 2.3 Escritorio de administración WordPress

2.2.3 Configuración de Visual Studio Code

Para poder realizar el desarrollo en lenguaje PHP, ha sido necesaria la instalación y configuración del entorno de desarrollo integrado Visual Studio Code. Dado que la descarga e instalación del software es trivial y no presenta ninguna incidencia o dificultad, nos centraremos en la configuración de los complementos necesarios, utilizados a lo largo del proceso de desarrollo, para facilitar las tareas de control de versiones y depuración.

Para el control de versiones vamos a utilizar Git y aunque Visual Studio Code integra una interfaz gráfica para la gestión de los comando, existe una extensión denominada "GitLens" que permite ampliar las características de Git, permitiendo visualizar el autor del código, navegar y explorar los repositorios Git. También instalaremos "Git History" que nos proporciona una vista profunda del historial Git.

Para depurar el código PHP a través de Visual Studio Code, utilizaremos la extensión Xdebug que previamente hemos activado y configurado en nuestro servidor Web. Para poder usar la depuración desde el editor, primero debemos instalar la extensión "PHP Debug". Una vez instalada la extensión es necesario configurarla editando el fichero "launch.json" para incluir una configuración denominada "Listen for XDebug". A tener en cuenta que en la versión utilizada en

este trabajo hemos tenido que cambiar el puerto que se escucha por defecto que es el 9000, por el 9003 que es para el que funciona correctamente para la última versión de Xdebug. [9]

Quedando la configuración del fichero como sigue:

launch.json

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Listen for XDebug",
      "type": "php",
      "request": "launch",
      "cwd": "${workspaceRoot}",
      "port": 9003,
      "stopOnEntry": false,
      "pathMappings": {
        "C:/laragon/www/tiactivos": "C:/laragon/www/tiactivos"
      }
    },
    {
      "name": "Launch currently open script",
      "type": "php",
      "request": "launch",
      "program": "${file}",
      "cwd": "${fileDirname}",
      "port": 9000
    }
  ]
}
```

La segunda configuración la crea por defecto la extensión, y no la usaremos. Está pensada para depurar un script de php directamente.

3

Metodología

Para la planificación de este proyecto se usará la metodología SCRUM, un modelo de metodología ágil en el que se aplican un conjunto de buenas prácticas para trabajar colaborativamente en equipo con el fin de obtener el mejor resultado posible de un proyecto. En SCRUM se realizan tareas parciales del producto final, realizando un trabajo de forma incremental, priorizando el beneficio que aportan al proyecto final. Resultando especialmente indicado para proyectos en los que necesitan resultados inmediatos y los requisitos y prioridades pueden ser cambiantes. [10]

Si bien la ejecución del proyecto por parte de una única persona no permitirá aplicar todos aquellos aspectos de una metodología ágil que implica a un equipo, sí será posible organizar el trabajo en *sprints*, definir el rol de *product owner* y del *scrum master*, seguir un esquema de priorización de las funcionalidades, correspondiendo al alumno el rol de equipo de desarrollo. [11]

Se ha aplicado gestión de proyectos para un adecuado control del trabajo, tanto para conseguir una ejecución eficiente en plazos, como la calidad esperada para los entregables finales.

Para el desarrollo del software se ha aplicado la metodología DevOps, ligada al desarrollo software ágil, en ella los equipos de desarrollo comienzan a colaborar y trabajar conjuntamente para cubrir todo el ciclo del desarrollo de forma que los procesos son mucho más ágiles y seguros,

garantizando un producto final de mayor calidad y fiabilidad. [12]

3.1 Planificación

Las fases en que se han distribuido las tareas del trabajo son las siguientes:

Sprint	Tarea	Temporalización
0	Identificar la solución tecnológica	Enero 2021
1	Montaje inicial de la Web y configuración de usuarios	Febrero 2021
2	Desarrollo de la funcionalidad registro de activos	Febrero 2021
3	Diseño del interfaz de usuario	Abril 2021
4	Implementar funcionalidad para mostrar un activo	Mayo 2021
5	Implementar funcionalidad de búsqueda de activos	Mayo 2021
6	Incorporar filtros a la búsqueda de activos	Junio 2021
7	Implementar paginación de resultados de búsqueda	Junio 2021
8	Pruebas, detección y corrección de errores	Julio 2021
9	Elaboración de la memoria	Julio 2021

Tabla 3.1 Fases del trabajo por Sprints y mes de realización

3.1.1 Sprint 0: Identificar la solución tecnológica

- Concretar la estructura de la Información
- Concretar el tema a utilizar en WordPress
- Concretar los complementos o *plugins* necesarios
- Concretar forma en la que los usuarios alimentan la Web

- Definir según los estándares de WordPress la estructura de carpetas debajo del tema hijo o en general para un plugin a medida

3.1.2 Sprint 1: Montaje inicial de la Web y configuración de usuarios

- Instalar servidor Web local Laragon
- Instalar Visual Studio Code
- Mostrar campos personalizados en interfaz del usuario (*frontend*)
- Crear tipo personalizado ACTIVO
- Crear un usuario de tipo EDITORACTIVOS
- Instalar el complemento para la administración de roles
- Configurar el rol de usuario EDITORACTIVOS para alumno

3.1.3 Sprint 2: Desarrollo de la funcionalidad registro de activos

- Crear campos personalizados para el tipo personalizado ACTIVO
- Actualizar la configuración de campos personalizados con .JSON
- Formateado de los campos personalizados
- Crear plugin personalizado activos-mng (*activos management*)

3.1.4 Sprint 3: Diseño del interfaz de usuario

- Introducción de los activos
- Hacer el boceto de la página para mostrar la ficha de un activo
- Hacer el boceto gráfico de la página de búsqueda de activo
- Definir las clases necesaria para manejar la información de los campos en un activo
- Formatear el contenido con estilos en base al diseño realizado usando el tema AVADA de WordPress
- Replicar los diseños realizados en el servidor remoto al entorno de desarrollo

3.1.5 Sprint 4: Implementar funcionalidad para mostrar un activo

- Diseñar una plantilla para activos con la interfaz gráfica.
- Crear una rama o *branch* para desarrollar en local

- Construir página de un activo con los campos conforme al diseño de la interfaz gráfica.
- Implementar las casillas de verificación de los campos de Lenguajes de Programación y Sistemas Operativos
- Actualizar el modelo de datos para mostrar referencias
- Revisar visualización del activo, para que si no existe un campo o pestaña no se muestre

3.1.6 Sprint 5: Implementar funcionalidad de búsqueda de activos

- Implementar función de búsqueda de resultados
- Implementar clase que dibuje el activo en pantalla.
- Implementar la vista del resultado de la búsqueda de activos
- Implementar una barra lateral para realizar las búsquedas
- Crear un formulario de búsqueda para activos con condiciones

3.1.7 Sprint 6: Incorporar filtros a la búsqueda de activos

- Crear clase para generar los argumentos de la búsqueda personalizada.
- Mantener las condiciones seleccionadas en la barra lateral
- Añadir botón borrar para resetear las condiciones en la barra lateral
- Cambiar enlace en nombre de la ficha de activo por su slug
- Incluir información total resultados en los resultados de búsqueda

3.1.8 Sprint 7: Implementar paginación de resultados de búsqueda

- Mostrar las páginas para poder navegar por lo resultados
- Implementar botón página anterior
- Implementar botón página siguiente
- Mantener las condiciones seleccionadas en el formulario de búsqueda al cambiar de página.

3.1.9 Sprint 8: Pruebas, detección y corrección de errores

- Realizar búsquedas y revisar los resultados obtenidos
- Los resultados de búsqueda de los activos no se visualizan correctamente
- La búsqueda por el campo TIPO ACTIVO no funciona
- Las búsqueda por campo PRECIO no funciona correctamente
- Mapear los nombres de los campos en el formulario de búsqueda y en el activo

3.1.10 Sprint 9: Elaboración de la memoria

- Redacción de la memoria
- Revisión
- Correcciones y mejoras
- Presentación y entrega

3.2 Gestión

La carga de los activos que podemos encontrar en el sitio Web www.tiactivos.com ha sido realizada por los alumnos de la asignatura impartida por el tutor de este trabajo fin de grado.

Recordemos que uno de los objetivos que pretendíamos alcanzar al iniciar este trabajo, era proporcionar un espacio que pudiera ser utilizado no solo por profesionales de las tecnologías de la información para consultar y comparar recursos tecnológicos, sino también para que pudiera ser utilizado como recurso docente de cara a la formación de los alumnos. En este sentido, una de las tareas prácticas que ha formado parte de la asignatura Calidad del Software, es la de analizar distintos activos tecnológicos disponibles en la actualidad, para después registrar el resultado de dicho análisis en el sitio Web.

Por este motivo a partir del sprint 2, la funcionalidad que permite insertar, modificar y eliminar un activo debía estar finalizada de forma que los alumnos pudiesen registrar los activos y posteriormente completar los campos con el resultado de su estudio y análisis del activo.

El resultado final es de un total de 243 activos registrado en el sitio Web, que dotan no solo de valor a nivel de contenido, sino que además permitían probar y utilizar todas las funciones implementadas a lo largo del trabajo.

En la siguiente tabla se recogen los activos cargados en el sitio Web por los alumnos de la asignatura Calidad de Software.

ID	Activo
1	Koding
2	XAMPP
3	Dart
4	Flutter
5	SQLite
6	Moqups
7	MoonModeler
8	Enterprise Architect
9	StarUML
10	PythonAnywhere
11	sqlmap
12	UXPin
13	C#
14	Inkscape
15	IntelliJ
16	Matlab
17	PyCharm
18	PyTorch
19	Sci-kit
20	Sketch
21	SublimeText
22	TensorFlow
23	Bitbucket
24	Java
25	MariaDB
26	Maven
27	Jenkins
28	Slack
29	Jira
30	JPA

31	Quarkus
32	Ansible
33	BackboneJS
34	Gradle
35	Capsule
36	Ajax
37	Android Studio
38	Angular
39	Cameo Requirements Modeler
40	Figma
41	Kotlin
42	Swift
43	Tailwind CSS
44	KanbanFlow
45	GitKraken Boards & Timeline
46	Kiuwan
47	Kubernetes
48	Lamp
49	Nagios
50	Perl
51	Spyder
52	Struts
53	Flask
54	SoapUI
55	Redis
56	Redmine
57	GitHub
58	XML
59	Wireshark
60	TeX
61	WildFly

62	Vue.js
63	Erlang
64	JSF (Java Server Faces)
65	GitLab
66	PostgreSQL
67	Notepad++
68	R
69	Trello
70	Nozbe
71	Netlify
72	Google Analytics
73	Cloud9 IDE
74	Apache Cassandra
75	Apache Subversion
76	Axure
77	BlueJ
78	Puppeteer
79	Confluence
80	DigitalOcean
81	Docker
82	CodeIgniter
83	ionic
84	OpenShift
85	Microsoft Azure
86	Delphi
87	Drupal
88	Unity
89	Google Cloud Platform
90	Heroku
91	JavaScript
92	Json
93	TypeScript
94	Paymo

95	Prolog
96	Rapid Miner
97	Code::Blocks
98	Haskell
99	IFMLEdit
100	Monday.com
101	MySQL
102	Postman
103	Wordpress
104	Cucumber
105	Gatsby
106	Hudson
107	Basecamp
108	Chanty
109	Datadog
110	Fluid UI
111	JMeter
112	Netsparker
113	PHPStorm
114	Monotone
115	Polymer
116	TestRail
117	Skype
118	Visual Paradigm
119	Sphinx
120	Vercel
121	Zabbix
122	Zoho Projects
123	Zoho Creator
124	Nessus
125	Browserstack
126	BlazeMeter
127	NinjaMock

128	Balsamiq
129	Dynatrace
130	Grafana
131	Travis
132	PHP
133	Laravel
134	Gatling
135	Taiga
136	MagicDraw
137	MantisBT
138	Discord
139	Github Desktop
140	Microsft Teams
141	Gitkraken Git GUI
142	Evernote
143	Simplenote
144	NextCloud
145	Perfex
146	SQL Server
147	React
148	Npm
149	Yarn
150	Visual Studio code
151	Node Js
152	Git
153	Selenium
154	Python
155	Sass
156	SonarLint
157	Dropbox
158	Mattermost
159	Symfony
160	Spring

161	Marvel
162	Ruby
163	Rational Doors
164	Asana
165	Sensu
166	Nanobox
167	Katalon
168	Avast Antivirus
169	AVG
170	Atom
171	BitDefender
172	Brackets
173	Eiffel
174	COBOL
175	Ghost
176	Eclipse
177	Mercurial
178	Papyrus
179	Ranorex
180	Kaspersky Antivirus
181	SonarQuBe
182	Tableau
183	Mockito
184	NetBeans
185	Django
186	Firebird
187	Acunetix
188	Evosuite
189	Express.js
190	Appium
191	Sails.js
192	Meteor.js
193	Koa.js

194	Invisionapp
195	JustInMind
196	MongoDB
197	pytest
198	Swagger
199	Chaos Monkey
200	Microsoft Project
201	Notion
202	Prometheus
203	Caliber
204	Couchbase
205	Kentico
206	Bootstrap
207	Tricentis Tosca
208	Hapi Framework
209	Xamarin
210	Microsoft Visual Studio
211	OCL
212	Go
213	USE
214	Scala
215	Microsoft Visio
216	Team Foundation Server
217	ArangoDB
218	Redux

219	Cypress
220	Amazon Web Services
221	Avaza
222	Bulma
223	Vaadin
224	Chef
225	Geany
226	JUnit
227	Linx
228	Expression Studio
229	Strapi
230	Oracle Database
231	Dokku
232	jQuery
233	Next.js
234	Hugo
235	Cassandra
236	Oracle
237	Pivotal Tracker
238	Testlink
239	Saltstack
240	Puppet
241	TortoiseSVN
242	Watson IBM
243	Resource Description Network

Tabla 3.2 Listado de activos cargado en el sitio www.tiactivos.com

4

Especificación de requisitos

Para llevar a cabo la especificación de los requisitos que debe cumplir el sitio Web, en primer lugar hemos comparado con otras alternativas existentes en la actualidad con el objetivo de analizar sus características y determinar en qué se diferencian del resultado final y los objetivos planteados al inicio de este trabajo.

Una vez se haya realizado un estudio preliminar de los sitios existentes, será necesario definir qué tipos de usuarios van existir en nuestra Web y cómo deben interactuar con ella, definiendo por tanto cuál será su rol y qué permisos debe tener cada uno de ellos.

Finalmente, estaremos en condiciones de especificar las funcionalidades que debe implementar el sitio Web, para cumplir con las necesidades de los distintos usuarios que van a hacer uso de la misma, así como con los objetivos marcados al inicio del proyecto.

4.1 Comparativa con otros sitios Web

Se ha analizado el sitio www.capterra.es que está pensado para ayudar a los usuarios a encontrar software adecuado. Para ello dispone de un buscador que busca y compara el

software. También dispone de la opción de navegar por distintas categorías de software.

Aunque en principio se aproxima a lo deseado para nuestro sitio Web, no se ajusta, ya que está orientado a mostrar el software existente en un espectro muy amplio, más allá de los proyectos de tecnología. Por otra parte, este sitio sólo contempla software, dejando de un lado las tecnologías, librerías, frameworks, herramientas especializadas, etc. que se pueden usar en un proyecto de desarrollo de software en primer lugar, o de tecnología en general, como es el objetivo de nuestro sitio. Es decir, no entra en el nivel de profundidad que se persigue en el proyecto.

Tampoco muestra la relación del software con los estándares CMMI⁵ y PMBOK⁶, e incluso con procesos de gestión de servicios TI (estándares de referencia aquí son ITIL⁷ e ISO20000⁸) uno de los principales valores que va a ofrecer nuestro sitio Web, y que será de mucho interés para los alumnos y profesionales de las tecnologías de la información a la hora de identificar una solución tecnológica candidata de ser usada en un proyecto, o como herramienta estándar dentro de una organización.

Otro aspecto también contemplado en nuestro proyecto es el de mostrar los perfiles profesionales que podrían usar cada uno de los activos, información útil para empresas y profesionales, algo que no incluye el sitio de capterra.

Así mismo, entre las carencias que hemos detectado está el hecho de que no ofrece recursos docentes para la tecnología. Este trabajo tiene entre sus objetivos principales ser una herramienta de ayuda a la docencia y por tanto debe incorporar información relevante a recursos docentes relacionados con el activo.

El buscador implementado presenta similitudes con las necesidades de nuestro proyecto, ya que incorpora filtros de búsqueda que facilitan poder localizar el software.

En cuanto a la página de resultados tras realizar una búsqueda es muy visual, presentando una breve descripción y resumen de las principales funcionalidades del software. Cada recurso software está enlazado a la URL del proveedor

⁵ *Capability Maturity Model Integration*

⁶ *Project Management Body of Knowledge*

⁷ *Information Technology Infrastructure Library*

⁸ Norma ISO 20000 de calidad en los servicios y procesos que realiza una empresa de gestión de servicios TI

del mismo. Para nuestro proyecto, sería adecuado que pudiéramos acceder a una ficha del activo donde se recoja una información mucho más detallada de las características del recurso.

Otro de los sitios que hemos analizado es www.softonic.com un sitio Web que ofrece la descarga de programas gratuitos y de pago. Los contenidos descargables de esta Web se organizan en primer lugar por sistema operativos, aunque también están categorizados por la funcionalidad que cubren.

También incorpora un buscador que permite realizar búsquedas por el nombre del software y posteriormente aplicar filtros a los resultados por licencia, idioma, plataforma y sistema operativo.

Los resultados de la búsqueda se presentan en un formato compacto en el que se muestra una información mínima. Básicamente si es gratuito o de pago, idioma, sistema operativo y breve descripción. Además de una puntuación, con el número de valoraciones y descargas realizadas del software. Haciendo clic sobre la aplicación en que estamos interesados, podemos acceder a una ficha con una información más completa acerca del mismo.

Al igual que ocurre con www.capterra.es el sitio www.softonic.com no se ajusta al planteamiento propuesto en este trabajo, ya que el enfoque del sitio es que el usuario pueda localizar y descargar de la forma más fácil posible una aplicación o un software, pero no que se pueda utilizar la información registrada para un uso docente o profesional, en los que se prima conocer las características del activo y su relación con los estándares.

Por último, hemos analizado el sitio www.hub.docker.com donde es posible buscar imágenes de aplicaciones para desplegarlas dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

Este sitio dispone de una opción para explorar las imágenes disponibles para descarga en el sitio Web, incorporando unos filtros que permiten elegir el tipo de publicador, las categorías, el sistema operativo y la arquitectura de procesador. Si bien, este sitio no se ajusta por el contenido que ofrece a las necesidades de nuestro sitio Web para activos, la forma en que se presentan las imágenes disponibles y se pueden filtrar, sí se aproximan al resultado que necesitamos ofrecer en nuestro sitio. Y de hecho nos ha

servido de referencia e inspiración, para realizar el diseño de los resultados de una búsqueda.

En conclusión, analizadas distintas opciones existentes en la actualidad que permiten buscar y filtrar aplicaciones o soluciones software, no podemos encontrar ninguna que cumpla con las necesidades requeridas para nuestro proyecto, motivo que justifica el valor y la aportación de de un sitio Web dirigido al registro, mantenimiento y consulta de los activos tecnológicos más importantes disponibles en la actualidad y de interés a nivel docente y profesional para usuarios con perfil de ingeniero informático, o especializaciones similares.

4.2 Tipos de usuario de la Web

Este sitio Web contará con distintos tipos de usuario, en función del interés que les motiva a usar esta solución, de cómo acceden a la misma y de los permisos para realizar acciones de búsqueda y consulta.

El primero de esos tipos de usuarios es el de **alumno** que desea conocer que activos existen y cuáles son sus características. Esa consulta debe poder hacerse desde la interfaz de usuario disponible para usuarios sin necesidad de estar registrados e identificados en la Web. Otra de las motivaciones para este tipo de usuario es la formación en diferentes tecnologías.

Estos usuarios de tipo alumno, también deben poder acceder como parte de su formación de forma identificada a la zona de administración, usando una cuenta con el rol de EDITORACTIVOS, dónde podrán crear nuevos activos, de forma que puedan introducir la información referida al activo. Cada uno de estos usuarios podrá ver todos los activos del sistema, pero únicamente podrá editar aquellos activos que el mismo haya creado, pudiendo incluso borrar el activo.

El segundo tipo de usuario que puede visitar la Web es un **profesional** de las tecnologías de la información que desea formarse o conocer las diferentes tecnologías existentes.

Una de las preguntas a las que debe dar respuesta nuestro sitio es, dado el puesto que desempeña un profesional en la empresa, ¿qué tipo de herramientas y tecnologías, es decir activos, debería conocer?

Suponiendo que quiero evolucionar desde un puesto de programador a analista, ¿qué activos existen y debería conocer para poder dar ese paso?

Por otra parte, este tipo de usuario, el profesional de las tecnologías de la información, es muy posible que tenga que buscar una solución técnica para un proyecto en el que debe tomar una decisión con respecto a las tecnologías y herramientas a utilizar, y nuestro sitio Web debe poder facilitar las alternativas que existen de forma que sirva de ayuda a la toma de la decisión.

El tercer tipo de usuario es el **docente**, que hará uso de la Web para hacer un análisis de los activos que existen en el mercado, permitiendo incorporar a sus clases estas herramientas y tecnologías.

Por cuarto, el tipo de usuario **empresa**, que en pro de mejorar su capacidad de desarrollo y producción software, puede hacer uso de la Web para encontrar los activos existentes, que le permita cumplir con determinadas prácticas o estándares.

Por último, tendremos un usuario **administrador**, que será el encargado de gestionar todo el sistema. Este usuario tendrá la capacidad de crear nuevas cuentas de usuario de tipo EDITORACTIVOS, administrar los privilegios y tendrá control absoluto sobre todos los activos registrados en el sistema.

4.3 Funcionalidades

Tras realizar un análisis de los requisitos descritos en el anteproyecto de trabajo fin de grado y la estructura de la información necesarias para el *hub* de activos tecnológicos, así como las interfaces necesarias para implementar dichas funcionalidades, se han extraído los siguientes requisitos funcionales y casos de uso.

4.3.1 Creación de un nuevo activo

La creación de un nuevo activo solo estará disponible para usuarios identificados en el sitio Web que tengan concedidos permisos para gestionar los activos.

Desde el menú de administración deberán poder acceder a un formulario que les permita registrar toda la información correspondiente a un activo. En dicho formulario encontraremos unos campos clave que obligatoriamente deben ser completados por el usuario y otros campos que podrá rellenar de forma opcional, pudiendo completarlos más adelante.

La estructura de datos y campos necesarios para almacenar la información de un activo será la siguiente:

1. **Nombre del activo**, denominación del activo campo obligatorio
2. **Logotipo**, imagen identificativa del activo, campo obligatorio.
3. **Descripción corta**, breve descripción del activo, campo de carácter obligatorio.
4. **Descripción larga**, permite recoger de forma extensa una descripción del activo.
5. **Qué es**, campo que nos permite elegir desde un desplegable categorías, recogidas en la tabla 4.1, indicando el nivel de ajuste entre 1 y 5 existente entre el activo y la categoría, así como un texto con la justificación del nivel de ajuste. Es obligatorio indicar entre 1 y 3 categorías para este campo del activo.
6. **Tipos de activos**, este campo que nos permite elegir desde un desplegable tipos de activos, recogidos en la tabla 4.2, indicando el nivel de ajuste entre 1 y 5 existente entre el activo y el tipo de activo, así como un texto con la justificación del nivel de ajuste. Es obligatorio indicar entre 1 y 3 tipos de activo para este campo del activo.
7. **Lenguajes de programación**, este campo permitirá seleccionar tantos lenguajes de programación como sean necesarios con los que está relacionado el activo. Es decir el activo en cuestión se puede usar en proyectos que se estén desarrollando con el/los lenguaje/s de programación que se indiquen. No es obligatorio completar este campo.
8. **Es comparable con**, este campo permite elegir otros activos existentes en el sitio, indicando el valor de ajuste entre 1 y 5 y una justificación del nivel de ajuste. El campo no es obligatorio y permite indicar hasta 3 activos con los que es comparable.
9. **Tipos de proyecto**, este campo que nos permite elegir desde un desplegable los tipos de proyectos para los que se puede usar este activo de entre los recogidos en la tabla 4.3, indicando el nivel de ajuste entre 1 y 5 existente entre el activo y el tipo de proyecto, así como un texto con la justificación del nivel de ajuste. Es obligatorio indicar entre 1 y 3 tipos de proyecto para este campo del activo.
10. **Sistemas operativos**, este campo debe permitir seleccionar todos los sistemas operativos para los que está disponible el activo. No es un campo obligatorio.
11. **Actividades de proyecto**, este campo nos permite elegir entre las actividades disponibles en la tabla 4.4, indicando el nivel de ajuste entre 1 y 5 y la

justificación de dicho ajuste. Es obligatorio indicar entre 1 y 3 actividades de proyecto.

12. **Perfiles profesionales**, este campo permite indicar a qué perfiles profesionales se ajusta el activo, conforme a los recogidos por la tabla 4.5, indicando un nivel de ajuste entre 1 y 5, y pudiendo justificar ese grado de ajuste. Es decir, qué perfiles profesionales deberían conocerlo, y/o podría tener más sentido usar el activo en el tipo de trabajo que realiza dicho perfil. El campo es obligatorio y el usuario podrá indicar entre 1 y 3 perfiles profesionales.
13. **Precio**, campo que recoge si el activo es gratuito, si es una versión de pago o si dispone de ambas posibilidades. Es un campo obligatorio.
14. **Área de procesos CMMI**⁹, campo para elegir las áreas de proceso de este estándar, en las que tiene más sentido usar el activo en cuestión, indicando el nivel de ajuste con el área entre 1 y 5, junto con la justificación del ajuste. Las áreas de procesos CMMI, están recogidas en la tabla 4.6. Este campo no es obligatorio y por tanto se podrán indicar entre 0 y 3 áreas.
15. **Objetivos específicos CMMI**, campo para elegir los objetivos específicos de este estándar, a los que el activo podría estar ayudando a lograr, indicando el nivel de ajuste con el objetivo entre 1 y 5, junto con la justificación del ajuste. Las objetivos específicos CMMI, están recogidas en la tabla 4.7. Este campo no es obligatorio y por tanto se podrán indicar entre 0 y 3 valores para el campo.
16. **Procesos PMBOK**¹⁰, campo para indicar los procesos PMBOK asociados a un activo, es decir, en qué procesos tiene más sentido utilizar el activo en cuestión, a elegir entre los disponibles en la tabla 4.8. Es posible indicar el nivel de ajuste con el proceso entre 1 y 5, indicando la justificación de dicho ajuste. El campo no es obligatorio, por tanto se deben poder introducir entre 0 y 3 valores.
17. **Proceso servicios ITIL**¹¹, campo para indicar los procesos de servicios ITIL asociados a un activo, de

⁹ Integración de Modelos de Madurez de Capacidades

¹⁰ Guía de los fundamentos para la dirección de proyectos

¹¹ Biblioteca de Infraestructura de Tecnologías de Información

forma similar, en qué procesos tendría más sentido usar el activo en cuestión, a elegir entre los disponibles en la tabla 4.9. Es posible indicar el nivel de ajuste con el proceso entre 1 y 5, indicando la justificación de dicho ajuste. El campo no es obligatorio, por tanto se deben poder introducir entre 0 y 3 valores.

18. **Funcionalidades - Características**, campo de texto donde es posible describir las funcionalidades y características del activo. No es un campo obligatorio.
19. **Referencias de descripción**, campo para poder añadir enlaces a páginas Web con información descriptiva sobre el activo. El campo es opcional y debe permitir añadir cuantas referencias sean necesarias.
20. **Referencias de formación**, campo para añadir enlaces a páginas Web dónde se pueden encontrar recursos de formación y aprendizaje relacionados con el activo.
21. **Comentarios**, este campo permite a los usuario incluir comentarios, opiniones y testimonios relacionados con el activo.

Id	Categoría de activo
1	Herramienta de chequeo de código
2	Lenguaje de programación
3	Framework desarrollo
4	Stack de software
5	Base de Datos
6	Lenguaje de programación
7	Framework CSS
8	Gestión de Proyecto
9	DevOps
10	Entorno desarrollo
11	Diseño de IU
12	CRM
13	Desarrollo de Pruebas Unitarias
14	Desarrollo y prueba de APIs
15	Construir proyectos y resolver dependencias
16	Clientes GIT

Tabla 4.1 Categorías de activos disponibles para el campo Qué es

Id	Tipo de activo
1	Herramienta
2	Tecnología
3	Librería
4	Producto
5	Sistema
6	Software
7	Componente
8	Framework
9	Arquitectura
10	Solución
11	Plataforma
12	Lenguaje
13	BD
14	Otro

Tabla 4.2 Tipos de activos

Id	Tipo de proyecto
1	Producción de software
2	Gestión del servicio TI
3	Gobierno TI
4	Consultoría
5	Inteligencia artificial
6	Big data
7	Marketing digital
8	Otro

Tabla 4.3 Tipos de proyectos

Id	Actividad
1	Captura de Requisitos
2	Especificación

3	Análisis
4	Diseño Técnico
5	Arquitectura Software
6	Diseño Gráfico
7	Programación Software
8	Pruebas
9	Calidad
10	Entorno de desarrollo
11	DevOps
12	Integración Continua
13	Entrega Continua
14	Despliegue Continuo
15	Entorno de Preproducción
16	Entorno de Producción
17	Operaciones
18	Gestión de Servicio TI
19	Planificación
20	Medición

Tabla 4.4 Actividades de proyecto

ID	Perfil profesional
1	Product Owner
2	JP
3	Scrum Master
4	Analista
5	Arquitecto
6	Diseñador técnico
7	Especialista de Pruebas
8	Analista-Programador
9	Programador Front-End
10	Programador Back-End
11	Full-stack developer
12	Especialista UX

13	Especialista IU
14	Diseñador Gráfico
15	Tester
16	Especialista DevOps
17	Site Reliability Engineer (SRE)
18	Administrador de Sistemas

Tabla 4.5 Perfiles profesionales

ID	Área de proceso CMMI
2-PP	Planificación del Proyecto
2-PMC	Monitorización y control del proyecto
2-RM	Gestión de requisitos
2-SAM	Gestión de acuerdos con proveedores
2-CM	Gestión de la configuración
2-PPQA	Aseguramiento de la calidad del producto y el proceso
2-MA	Medición y análisis
3-IPM	Gestión de proyectos integrada
3-RM	Gestión de riesgos
3-VAL	Validación
3-VER	Verificación
3-RD	Desarrollo de requisitos
3-ST	Solución técnica
3-PI	Integración del producto
3-OT	Formación organizacional
3-OPD	Definición organizacional de procesos
3-OPF	Enfoque organizacional de procesos
3-DAR	Análisis de decisiones y resolución
4-OPP	Desempeño organizacional de procesos
4-QPM	Gestión cuantitativa de proyectos
5-CAR	Análisis de causas y resolución
5-OPM	Gestión organizacional del desempeño

Tabla 4.6 Áreas de procesos CMMI

ID	Objetivo específico CMMI
2-PP	Realizar estimaciones
2-PP	Elaborar un Plan de Proyecto
2-PP	Obtener compromiso con el Plan
2-PMC	Monitorizar el proyecto según el Plan
2-PMC	Gestionar Acciones Correctivas hasta cerrarlas
2-RM	Gestionar los Requisitos
2-SAM	Realizar acuerdos con proveedores
2-CM	Establecer Líneas Base
2-CM	Seguir y controlar los cambios
2-CM	Establecer Integridad
2-PPQA	Evaluar objetivamente los procesos y los productos generados
2-PPQA	Proporcionar información objetiva
2-MA	Alinear medidas y actividades de análisis
2-MA	Proporcionar resultados de medidas
3-IPM	Usar procesos previamente definidos en los proyectos
3-IPM	Coordinar y colaborar con grupos de interés relevantes
3-RM	Preparación para la gestión de riesgos
3-RM	Identificar y analizar riesgos
3-RM	Mitigar los riesgos
3-VAL	Preparación para la Validación
3-VAL	Validar productos o componentes de productos
3-VER	Preparación para verificación
3-VER	Realizar revisiones entre pares
3-VER	Verificar los productos generados que se seleccionen
3-RD	Describir y concretar los requisitos de clientes
3-RD	Describir y concretar los requisitos de productos

3-RD	Analizar y validar los requisitos
3-TS	Seleccionar soluciones para los elementos a construir
3-TS	Describir y concretar el Diseño
3-TS	Implementar el Diseño de los elementos a construir
3-PI	Preparación para la Integración de los elementos del Producto
3-PI	Asegurar la compatibilidad de los interfaces
3-PI	Ensamblar los elementos generados y entregar el producto
3-OT	Crear y mantener una capacidad de formación organizacional
3-OT	Proporcionar Formación
3-OPD	Crear y mantener Activos de Procesos en la Organización
3-OPF	Determinar Oportunidades de Mejoras en los Procesos
3-OPF	Planificar e implementar Acciones en los Procesos
3-OPF	Desplegar Activos de Procesos en la Organización e Incorporar Experiencias
3-DAR	Evaluar alternativas
4-OPP	Crear y mantener líneas base y modelos del rendimiento en la Organización
4-QPM	Preparación para Gestión Cuantitativa
4-QPM	Gestión cuantitativa de los proyectos
5-CAR	Determinar las causas de los Resultados que se seleccionen
5-CAR	Abordar las causas de los Resultados que se seleccionen
5-OPM	Gestionar el Rendimiento del negocio
5-OPM	Seleccionar Mejoras
5-OPM	Desplegar Mejoras

Tabla 4.7 Objetivos específicos CMMI

ID	Procesos PMBOK
1	Definir el Proyecto
2	Planificar el Proyecto
3	Capturar y describir requisitos
4	Definir el Alcance
5	Crear y manejar Esquemas de Desglose de Trabajo (EDT)
6	Crear y manejar Planificaciones
7	Realizar Estimaciones de trabajo para planificar
8	Gestionar Costes y Presupuestos de Proyecto
9	Gestionar y Planificar la Calidad del Proyecto
10	Gestionar las Personas en el Proyecto
11	Gestionar las Comunicaciones en el ámbito del Proyecto
12	Gestionar los Riesgos
13	Gestionar las Compras y Subcontratación
14	Monitorizar y seguir el Proyecto

Tabla 4.8 Procesos PMBOK

ID	Proceso de servicio ITIL
1	Estrategia del Servicio. Gestión Financiera
2	Estrategia del Servicio. Gestión del Portafolio
3	Estrategia del Servicio. Gestión de la Demanda
4	Estrategia del Servicio. Gestión de Relaciones con el Cliente
5	Diseño del Servicio. Gestión del Catálogo de Servicios
6	Diseño del Servicio. Gestión de Niveles de Servicios
7	Diseño del Servicio. Gestión de la Disponibilidad
8	Diseño del Servicio. Gestión de la Capacidad
9	Diseño del Servicio. Gestión de la Continuidad de los Servicios de TI
10	Diseño del Servicio. Gestión de Proveedores
11	Diseño del Servicio. Gestión de la Seguridad de Información
12	Diseño del Servicio. Coordinación del Diseño

13	Transición del Servicio. Gestión de la Configuración y Activos
14	Transición del Servicio. Gestión del Cambio
15	Transición del Servicio. Gestión del Conocimiento
16	Transición del Servicio. Planificación y Apoyo a la Transición
17	Transición del Servicio. Gestión de Release y Despliegue
18	Transición del Servicio. Gestión Validación y Pruebas
19	Transición del Servicio. Evaluación (Evaluación del cambio)
20	Operación del Servicio. Gestión de Incidencias
21	Operación del Servicio. Gestión de Problemas
22	Operación del Servicio. Gestión de Peticiones
23	Operación del Servicio. Gestión de Eventos
24	Operación del Servicio. Gestión de Accesos
25	Mejora Continua del Servicio.
26	Medida del Servicio
27	Informes del Servicio

Tabla 4.9 Procesos de servicio ITIL

4.3.2 Modificación de un activo

Una vez creados los activos, estos deben poder ser editados por un usuario identificado con permisos de edición. Para ello será necesario acceder al listado de todos los activos existentes y permitir su modificación.

Es necesario diferenciar entre dos niveles de usuario, el administrador podrá editar cualquier activo existente en el sistema, pero los usuarios con rol EDITORACTIVOS sólo deberán poder modificar aquellos activos que han creado ellos. De esta forma se evita que un usuario cambie por error los activos que son responsabilidad de otro usuario.

4.3.3 Borrado de un activo

El borrado de un activo es una funcionalidad que debe estar disponible para usuarios identificados con rol de administrador o EDITORACTIVOS.

Los usuarios con rol EDITORACTIVOS sólo podrán borrar aquellos activos de los que son propietarios. Sin embargo, un administrador debe poder borrar cualquier activo.

4.3.4 Consulta de un activo

La estructura de información de un Activo en el sitio Web está compuesta de 6 bloques.

1. **Cabecera**, en la que se incluye una descripción breve y si existen versiones de software libre y/o de pago.
2. **Descripción**, se muestra información general del activo, las funcionalidades que ofrece, etc.
3. **Caracterización**. Uno de los objetivos del sitio es ofrecer una clasificación de los diferentes activos. Aquí se muestra la categorización de un activo atendiendo a diferentes parámetros: Qué es, su tipo, en qué tipo de proyectos principalmente se podría usar, para qué actividades, cuáles serían los perfiles profesionales que lo utilizan, etc. Finalmente, y si tiene sentido, para qué lenguajes de programación y/o sistemas operativos está preparado el activo.
4. **Comparable con**. Es una información que se considera muy valiosa, y se incluye en varios activos. Con el gran número de soluciones y herramientas disponibles actualmente, es normal que existan diferentes alternativas que puedan usarse de forma equivalente con la misma finalidad. En este apartado se muestra una comparación de los elementos más relevantes dependiendo del tipo de activo, con otros activos equivalentes.
5. **Estándares**. La finalidad de este apartado responde a otro de los objetivos importantes de TIActivos, que consiste en mostrar especialmente para profesionales y organizaciones TI, en qué proceso de un estándar tiene más sentido usar el activo correspondiente. Es decir, para las organizaciones y departamentos TI hoy en día es necesario seguir estándares dentro de los servicios TI que están bajo su responsabilidad (desarrollo de software, gestión de servicios, seguridad, etc.). Para tener optimizados y bien desplegados sus procesos, es imprescindible utilizar soluciones y herramientas que ayuden a trabajar con control, efectividad y calidad.
6. **Aprendizaje**. En esta sección se mostrarán referencias en Internet que ayuden a conocer de manera más amplia sobre el activo en cuestión (referencias de

descripción), o por otra parte, a formarse y aprender sobre el mismo (referencias de formación).

La siguiente es una imagen del boceto que fue creado para el contenido de un activo en la Web, antes de realizar su implementación.

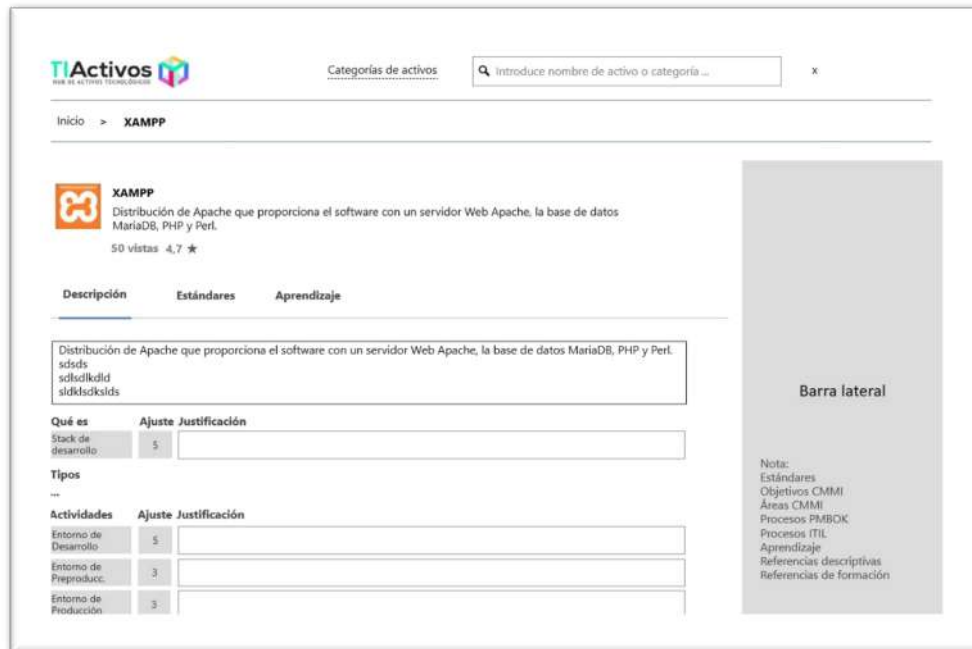


Figura 4.1 Boceto de la ficha de activo

4.3.5 Búsqueda de activos

Una de las principales funcionalidades de este sistema es la búsqueda de activos. Para ello debemos implementar un sistema que permita aplicar condiciones de búsqueda sobre distintos campos de interés del activo, de forma que el usuario pueda filtrar los resultados para afinar la búsqueda a sus necesidades. Los campos por los que debe ser posible buscar y filtrar los resultados son los siguientes:

- Qué es
- Precio
- Tipo de activo
- Tipo de proyecto
- Perfil profesional
- Actividad de proyecto
- Área de procesos CMMI
- Objetivos específicos CMMI
- Procesos PMBOK
- Gestión de servicios TI

Cada uno de estos filtros se convierten en una condición con carácter lógico Y, de forma que cada condición que incluimos se suma a las anteriormente añadidas, debiéndose cumplir todas ellas para que el activo aparezca en los resultados de la búsqueda.

Además, es necesario diseñar un formato al mostrar los resultados en el que además del nombre del activo aparezca otra información de interés de una forma compacta, pero con suficiente información para que el usuario pueda identificar en una primera visualización las principales características de cada activo mostrado en el listado de resultados. La información del activo que debe mostrarse es la siguiente:

- Logotipo
- Nombre del activo
- Descripción breve
- Precio
- Qué es
- Tipos
- Proyectos
- Actividades
- Lenguajes
- Perfiles

De manera similar a la vista de un Activo, la siguiente imagen muestra el boceto de pantalla que fue construido previamente a la implementación de la pantalla de Búsqueda de activos.

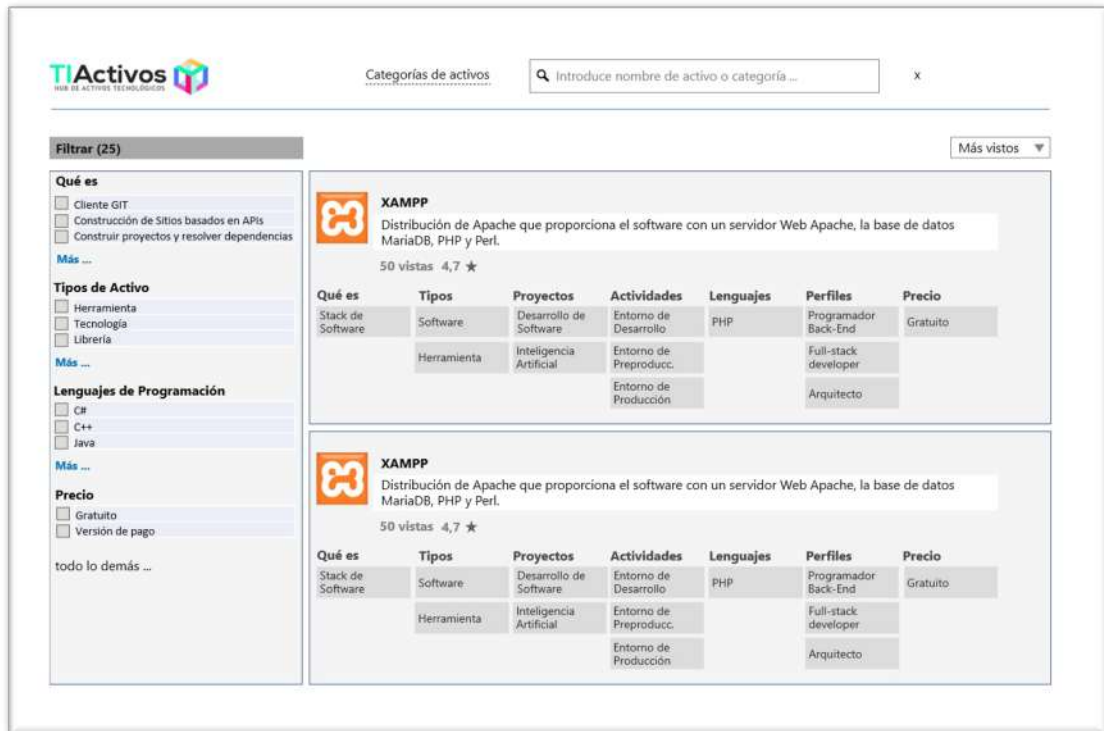


Figura 4.2 Boceto de filtro de búsqueda y resultados

4.3.6 Creación nuevos lenguajes de programación

Para el usuario administrador y el usuario con rol de EDITORACTIVOS es necesario implementar una funcionalidad que permita la creación nuevos lenguajes de programación, debido a la constante aparición de nuevos lenguajes de programación, ya de lo contrario los valores disponibles en este campo del activo quedarían obsoletos al no poder asociar el activo a los nuevos lenguajes de programación.

La funcionalidad además de permitir la creación de un nuevo lenguaje de programación, debe contemplar que ese nuevo lenguaje aparezca disponible en el campo designado a tal efecto en el activo.

4.3.7 Creación nuevos sistemas operativos

Para el usuario administrador y el usuario con rol de EDITORACTIVOS es necesario implementar una funcionalidad que permita la creación nuevos sistemas operativos, debido a la constante evolución de los mismos y a la aparición de nuevas versiones, de forma que las opciones disponibles para este campo del activo puedan mantenerse actualizadas.

Por este motivo, es necesario que además de facilitar al usuario la posibilidad de crear un nuevo sistema operativo, este quede incorporado a la lista de opciones disponible para ese campo designado en el activo.

4.4 Diagrama de casos de uso

A continuación se recogen los diagramas con los casos de uso con los diferentes perfiles de usuario que van a utilizar nuestro sitio Web, para cada una de las funcionalidades especificadas como requisitos del sistema.

Los perfiles de usuario son los siguientes:

1. Usuario no identificado
2. Editor
3. Administrador

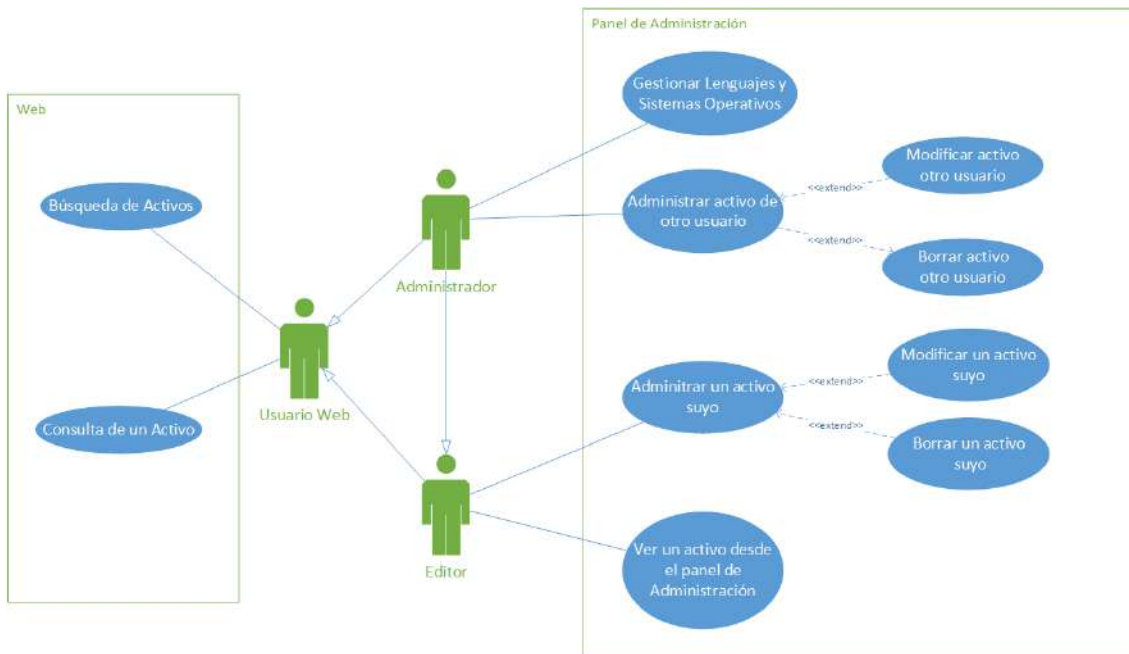


Figura 4.4 Diagrama de caso de uso

5

Solución técnica

5.1 Análisis de las alternativas

Se ha realizado un análisis de los temas y complementos disponibles para WordPress, con el objeto de determinar aquellos que mejor se ajusten a los requisitos funcionales del sitio Web descritos en el punto anterior.

El planteamiento inicial es analizar los complementos existentes y sus características para intentar identificar los que necesitamos para poder implementar las funcionalidades y requerimientos definidos para el sitio Web a desarrollar.

También se han analizado distintos temas que además de incorporar funcionalidades ofrecen una interfaz gráfica que nos permita ofrecer una apariencia que se ajuste al borrador del diseño.

En un primer momento hemos identificado distintos temas de WordPress que incluyen un buscador y también hemos valorado la posibilidad de aplicar filtros a los resultados para poder refinar los resultados de la búsqueda. También, se ha valorado el aspecto como se muestran los resultados de la búsqueda y la información que se muestra al usuario.

En la tabla 5.1 se recogen un resumen de los temas que se han valorado para desarrollar este trabajo. En ella se evalúan aspectos como que dispongan de un buscador integrado, que permitan ordenar los resultados, filtrar y refinar la búsqueda y por último que disponga de la posibilidad de valorar el cada elemento encontrado.

Nombre	Buscador	Ordenación	Filtro	Valoración ítems
Listingo ¹²	Sí	Sí	No	Sí
Alliance ¹³	No	No	No	No
Aardvark ¹⁴	Sí	Sí	Sí	Sí
Reviews ¹⁵	Sí	Sí	Sí	Sí
BlockBuster ¹⁶	Sí	Sí	No	Sí
Sequex ¹⁷	Sí	No	No	Sí

¹² <https://themographics.com/wordpress/listingo/job-listing/>

¹³ <https://themeforest.net/item/alliance-intranet-extranet-wordpress-theme/12087162>

¹⁴ <https://themeforest.net/item/aardvark-buddypress-membership-community-theme/21281062>

¹⁵ <https://themeforest.net/item/reviews-products-and-services-review-wp-theme/13004739>

¹⁶ <https://themeforest.net/item/blockbuster-film-review-movie-database-html-template/20011814>

¹⁷ <https://themeforest.net/item/sequex-wordpress-movie-database-theme/20845377>

Gauge ¹⁸	Sí	Sí	No	Sí
TopNews ¹⁹	Sí	No	No	Sí
Woodmart ²⁰	Sí	Sí	Sí	Sí

Tabla 5.1 Temas de WordPress analizados

Además de estas características, también se ha analizado el encaje de la ficha de los elementos con las información del activo que necesitamos almacenar y mostrar cuando un usuario consulta un activo.

Uno de los objetivos a lograr es conseguir el mejor ajuste posible entre el tema utilizado para la implementación en WordPress del sitio Web y el resultado deseado, tanto a nivel funcional como en la presentación y apariencia gráfica. La importancia de la elección del tema radica en que el elegido condicionará el desarrollo posterior de las funcionalidades.

Como veremos más adelante, se pretende aislar en lo posible la dependencia del desarrollo realizado para este trabajo del tema utilizado, para evitar que el funcionamiento se vea condicionado por las actualizaciones y modificaciones que puedan producirse en el tema elegido. En este sentido, hemos valorado además de las características técnicas, que el proveedor del mismo sea de contrastada reputación y que publique actualizaciones del tema con cierta regularidad.

Para implementar la funcionalidad de búsqueda de activos y filtrado de los resultados, se han estudiado los complementos que se enumeran en la tabla 5.2 teniendo en cuenta determinados aspectos de interés como son: Búsqueda por relevancia, buscar en categorías, buscar en etiquetas, buscar en taxonomías personalizadas, excluir entradas en

¹⁸ <https://themeforest.net/item/gauge-multipurpose-review-theme/8676079>

¹⁹ <https://themeforest.net/item/topnews-news-magazine-newspaper-blog-viral-buzz-wordpress-theme/16171130>

²⁰ <https://themeforest.net/item/woodmart-woocommerce-wordpress-theme/20264492>

los resultados de búsqueda y búsqueda en campos personalizados.

Complemento	Búsqueda por relevancia	Buscar en categorías	Buscar en etiquetas	Buscar en taxonomías personalizadas	Excluir entradas de la búsqueda	Campos personalizados
WP Extended Search	Sí	Sí	Sí	Sí	Sí	No
WP Google Search	Sí	No	No	No	No	No
Better Search	Sí	No	No	No	No	No
Relevanssi	Sí	Sí	Sí	No	No	Sí
Search Everything	No	No	Sí	No	No	Sí
Search & Filter	Sí	Sí	Sí	Sí	No	Sí

Tabla 5.2 Temas de WordPress analizados

Finalmente se ha concluido que ninguno de los temas y complementos anteriormente analizados, eran apropiados para implementar los requisitos de este trabajo. Los motivos son los siguientes:

- Ninguna de las alternativas proporciona una forma directa de implementar los interfaces gráficos y funcionalidades requeridas.
- Estos temas o soluciones aportan mucha funcionalidad añadida que no sería utilizada en nuestro sitio, cargándolo por tanto de un software adicional

innecesario, que complica la aplicación, además de añadir posibles riesgos de seguridad.

- Habría que estar actualizando el tema y complementos elegidos, que además seguirían añadiendo más código a funcionalidades que no se estarían utilizando.
- En cualquier caso, habría que trabajar bastante para adaptar el tema o complementos elegidos para obtener la funcionalidad requerida para nuestro sitio web.

En definitiva la conclusión fue partir de un tema de carácter general, y usar los mínimos complementos posibles para construir el sitio. Esto, además de eliminar las desventajas anteriores nos proporcionaría las ventajas siguientes:

- Libertad para construir exactamente lo que se desea.
- Depender lo mínimo de otros programas que también van evolucionando, y que requerirían además un mantenimiento.
- Al haberse construido a medida, facilitar la evolución futura del sitio, dado que se conoce con detalle lo que se ha hecho, y esto permite construir nuevas funcionalidades de forma más directa, fácil y sin dependencias.
- Aprender con la programación a medida que se va a realizar, que lógicamente es importante en un TFG como es el caso que nos ocupa.

Una vez tomada esta decisión optamos por utilizar Avada, un tema muy válido y extendido de carácter general, idóneo para el desarrollo de este trabajo.

En primer lugar, este tema es uno de los más vendidos del mundo y de los más utilizados, más de 100.000 usuarios en todo el mundo, esto nos garantiza un soporte continuado en el tiempo, con actualizaciones periódicas del tema, además de numerosas publicaciones de usuarios con información de ayuda sobre cómo usar las herramientas de que dispone este tema de WordPress. [13]

A nivel de diseño este tema nos ofrece una gran versatilidad y flexibilidad, permitiendo crear de una forma rápida una Web personalizada, con estética excepcional y con total funcionalidad en muy poco tiempo. Este tema ya incorpora un diseño responsable que facilita que pueda ser utilizado desde cualquier dispositivo, ya sea un teléfono móvil, una tableta o un ordenador de sobremesa.

El tema Avada dispone de un diseñador de páginas con formato visual, *fusion builder*, que acorta el tiempo de diseño, al poder arrastrar y soltar los elementos directamente sobre el diseño para componer la página Web.

A nivel funcional, Avada incorpora la gestión de tipos de personalizados de *posts*²¹, a través del complemento que nos va a permitir diseñar de forma gráfica y en muy poco tiempo un tipo de publicación personalizado que se ajuste a las características de un activo tecnológico. Junto con este complemento, el tema incorpora otro complemento para la gestión de campos personalizados, que nos será de gran utilidad a la hora de diseñar los campos que se usarán para crear y modificar los activos desde el formulario Web.

Por último, la amplia experiencia y conocimientos adquiridos en el manejo y el desarrollo de sitios Web con este tema por el profesor, ha sido otro de los factores determinantes en la elección de Avada como el tema base para el desarrollo de este trabajo fin de grado.

Adicionalmente, usaremos el complemento "User Role Editor" que permite cambiar los roles de los usuarios y privilegios de forma sencilla, permitiéndonos añadir nuevos roles de usuario y personalizar sus privilegios. [14]

En nuestro proyecto necesitaremos crear el rol correspondiente al usuario editor de activos, de acuerdo con las necesidades definidas en las especificaciones de forma que únicamente tenga acceso en el panel de administración al menú de "activos TI" e impidiendo que pueda editar y borrar aquellos activos que no ha creado el mismo, pero permitiendo su edición y borrado para los que él ha creado.

En resumen, se usará el tema Avada de WordPress, junto con los complementos que incorpora, además el complemento "User Role Editor".

5.2 Estructura de la información

Para recoger la información de los activos se va a utilizar un tipo contenido personalizado denominado "**activo TI**", que crearemos gracias al complemento para WordPress "*Custom Post Type UI*", que proporciona una interfaz fácil de usar para

²¹ Un "*post*" es la unidad de información en WordPress para el "blog", que se puede traducir por entrada, publicación, o artículo, por ejemplo. Aunque posteriormente este elemento de WordPress se ha extendido para manejar muchos tipos diferentes de datos.

registrar y administrar tipos de contenidos personalizados y taxonomías para tu sitio WordPress. [15]

Este nuevo tipo de contenido a medida será el encargado de gestionar los activos tecnológicos y dará cabida a toda la información que queremos recoger referente a un activo.

Para crear ese tipo de contenido tendremos a nuestra disposición la posibilidad de definir campos personalizados. Los campos personalizados son metadatos de una entrada de contenido creado en WordPress que nos permiten almacenar y recuperar información adicional sobre ese contenido y que por tanto nos permitirá crear los campos necesarios para almacenar todas la características de nuestro activo tecnológico.

Otra de las funcionalidades que nos aporta el uso de campos personalizados es la posibilidad de filtrar los contenidos en base a esos campos. Esta funcionalidad nos será de gran utilidad en el desarrollo de nuestro proyecto, ya que uno de los objetivos principales del mismo es la posibilidad de poder buscar activos que cumplan unas determinadas condiciones.

Aunque tanto los tipos personalizados, como los campos personalizados se pueden crear directamente desde código PHP, nosotros vamos a utilizar los complementos existentes para WordPress, el ya mencionado WordPress "*Custom Post Type UI*" y "*Advanced Custom Fields PRO*" (dos complementos muy extendidos actualmente que es usado incluso por especialistas de desarrollo en Wordpress que aconsejan usar complementos lo mínimo posible) para la creación y configuración de los campos personalizados. Ambos complementos vienen incluidos en el tema Avada y permiten una gestión simplificada y más eficiente de los tipos y campos personalizados

La estructura de datos necesaria para albergar la información de un activo tecnológico se recoge en la tabla 5.3, y deberá crearse a través de campos personalizados asociados al tipo de contenido personalizado "**activo TI**".

Una de las dificultades que tendremos que afrontar es mostrar en el contenido personalizado la información de los campos personalizados, pero esto podemos hacerlo a través de programación gracias a la interfaz de funciones que ofrece el complemento.

El complemento "*Custom Post Type UI*" nos permite crear también además de tipos de contenidos específicos taxonomías que pueden ser de dos tipos, la primera de ellas son las

categorías que jerarquizadas y la segunda son las etiquetas que no son jerarquizadas.

Para nuestro proyecto necesitaremos crear dos taxonomías jerarquizadas que son las siguientes:

1. Lenguajes de Programación
2. Sistemas Operativos

Campo Personalizado
Nombre del activo
Logotipo
Descripción corta
Descripción larga
Qué es
Tipos de activos
Lenguajes de programación
Es comparable con
Tipos de proyecto
Sistemas operativos
Actividades de proyecto
Perfiles profesionales
Precio
Área de procesos CMMI
Objetivos específicos CMMI
Procesos PMBOK
Proceso servicios ITIL
Funcionalidades - Características
Referencias de descripción
Referencias de formación
Comentarios

Tabla 5.3 Estructura de datos del activo tecnológico

5.3 Diseño técnico

En nuestro planteamiento hemos considerado la posibilidad de utilizar un modelo vista controlador (MVC), para separar la interfaz de usuario, de los datos y la lógica de control.

Aunque WordPress no sigue una arquitectura modelo vista controlador propiamente dicha, lo que hace su código es una separación y reutilización de las responsabilidades. En WordPress el **modelo** se correspondería con la programación incorporada en los complementos, la **vista** se correspondería con el tema y el **controlador** se correspondería con el cargador de plantillas que se ejecuta a través del fichero "index.php" y que carga las peticiones de páginas conforme a la jerarquía de las plantillas. [16]

Aunque existen los siguientes complementos para WordPress que permiten una implementación de la arquitectura modelo vista controlador:

- WP MVC
- Churro
- Tina MVC

Hemos tenido muy presente para la realización de este trabajo lo recogido en el apartado de buenas prácticas del manual de WordPress referente a los patrones de diseño en WordPress. En este apartado podemos encontrar información acerca de cómo implementar el patrón MVC²² en un complemento para WordPress [17]

Para implementar un complemento para WordPress que se ajuste a esta arquitectura, es necesario separar en tres partes las funciones para que una se encarguen de la vista, es decir de la interfaz de usuario, otra del modelo para negociar la capa lógica y por último otra para el controlador encargada de aplicar la capa lógica. Con ello conseguiremos una mejor organización y legibilidad, una mayor flexibilidad y reutilización del código, un mantenimiento más fácil del complemento y una programación profesional. [18]

²² Modelo Vista Controlador

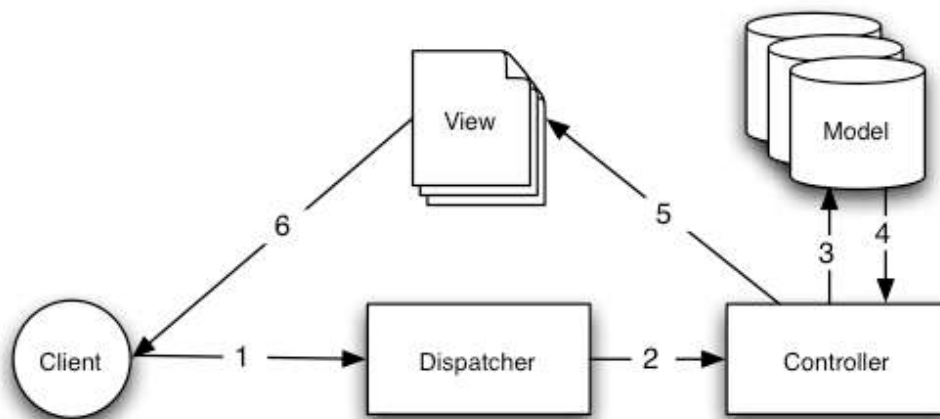


Figura 5.1 Esquema modelo vista controlador CakePHP [19]

Aunque estas implementaciones existentes permiten implementar un planteamiento más riguroso para modelo vista controlador, nosotros no hemos querido seguirlas debido a que el **marco de trabajo (framework) de WordPress está muy asentado**, y el núcleo y los complementos evolucionan respetándolo. Nos parecía un tanto arriesgado no utilizarlo tal cual, intentando ser más fiel a un planteamiento modelo vista controlador, pero en el que posteriormente tuviéramos conflictos en las siguientes versiones de WordPress y de sus complementos.

Por tanto, aunque hay posibilidades de realizar un planteamiento MVC puro, no parece que merezca la pena para este proyecto, ya que WordPress tiene todo lo que necesitamos y plantear una forma distinta de arquitectura, podría llevar a conflictos y problemas con las futuras actualizaciones del núcleo (*core*) de este sistema o de sus complementos.

Así pues hemos determinado la necesidad de desarrollar un complemento para WordPress al que denominaremos "**activos-mng**", tratando de encapsular dentro de este complemento todas las funcionalidades necesarias para la gestión de los activos.

Para su diseño vamos a utilizar siguiendo las indicaciones descritas en el apartado de buenas prácticas del manual de desarrollo de complementos para WordPress el complemento "**WordPress Plugin Boilerplate**" [20] que proporciona una base para crear un complemento bien organizado y orientado a objetos. Este complemento que nos servirá como punto de partida para el desarrollo de nuestro trabajo se basa en la interfaz de programación de aplicaciones de complementos para WordPress, en los estándares de codificación y documentación.

La estructura de carpetas de nuestro complemento respetará lo recogido en la sección de buenas prácticas del manual de desarrollo para complementos de WordPress:

```
/activos-mng
  activos-mng.php
  uninstall.php
  /languages
  /includes
  /admin
    /js
    /css
    /images
  /public
    /js
    /css
    /images
```

En nuestro trabajo hemos separado la parte de modelo de la parte de la vista, actuando como controlador el propio tema de WordPress utilizado para la implementación.

Para poder llevar a cabo esta separación hemos definido una clase "Activo_Model" que será la encargada de implementar todas las funciones relacionadas con el modelo de datos y una clase "Mng_Activo_Single_Activo_View" que será la encargada de gestionar la parte de la vista (en este caso para mostrar la página HTML de un activo simple) y hacer uso de las características que tiene el tema Avada con su propia codificación para aislar y concentrar en ella la gestión de la interfaz de usuario. Esto nos permite hacerlo de forma independiente del acceso a los datos que se lleva a cabo a través del modelo.

Las peticiones del usuario serán controladas por el tema que hará uso de las clases definidas para la vista con el objetivo de representar los datos que se obtendrán utilizando las clases definidas para el modelo.

Para el modelo de datos hemos necesitado definir dos clases, en primer lugar la antes mencionada "**Activo_Model**" para la gestión del modelo de datos de un activo simple y en segundo lugar la clase "**Mng_Search_Cond_Model**" para el manejo del modelo de datos en las búsquedas de activos.

La clase "**Activo_Model**" consta de las siguientes propiedades públicas:

- idActivo, identificador del activo.
- Nombre, nombre del activo.

- PostSlug, parte de la dirección URL que identifica al activo.

A su vez proporciona los siguientes métodos:

- loadActivo(\$argid), carga las propiedades del activo cuyo identificador se pasa como argumento.
- getTitulo(), retorna el nombre de un activo.
- getSlug(), retorna la parte de dirección URL del activo que va a continuación del nombre de dominio.
- getLabel(\$fieldName), retorna la etiqueta para el campo personalizado que se ha pasado como argumento.
- getNumCampos(\$fieldName), retorna el número de campos con ese nombre existentes en el activo.
- getValor(\$fieldName), retorna el valor del campo cuyo nombre se pasa como argumento.
- getGrupo(\$fieldName), retorna una matriz de 3 elementos con los valores del campo pasado como argumento, de forma que podrá ser utilizado por el controlador vista para representar el contenido.
- getGrupoRef(\$fieldName), retorna una matriz de 3 elementos que contiene la URL del campo referencia, el orden de esa URL y la descripción, de forma que podrá ser utilizado por el controlador vista para representar los campos de este tipo.
- getImagen(\$fieldName), retorna la URL donde se ubica la imagen del logotipo del activo.
- getTaxonomy(\$taxonomy), retorna una matriz con todos los valores para la taxonomía pasada como argumento.

La clase **"Mng_Search_Cond_Model"** únicamente tiene los siguiente métodos:

- set_fields(\$que_es = null, \$precio = null, \$tipo_activo = null, \$tipo_proyecto = null, \$perfil_prof = null, \$actividades = null, \$area_cmmi = null, \$obj_cmmi = null, \$pmbok = null, \$procesosITIL = null), construye un array con los valores introducidos por el usuario en el formulario de búsqueda y con los nombres de campos mapeados al nombre en el activo.
- get_Args(\$num_posts_per_pag, \$act_page, \$search_fields = array()), retorna una matriz con los criterios de búsqueda procesados para ser compatible con el formato requerido por clase WP_Query de WordPress.

La clase **"Mng_Activo_Single_Activo_View"** tiene una única propiedad privada:

- `$color_cabecera`, color a usar en la cabecera de un activo.

A su vez tiene los siguiente métodos:

- `abre_contenedor_1()`, retorna el HTML necesario para crear el primer contenedor
- `abre_fila_contenedor_1()`, retorna el HTML necesario crear una fila en el contenedor 1
- `abre_columna_1_contenedor_1()`, retorna el HTML necesario abrir una columna en el contenedor 1.
- `imagen($url = null)`, retorna el código HTML para mostrar la imagen cuya dirección se pasa como argumento.
- `precio($label_precio, $gratis, $pago)`, retorna el código HTML para mostrar el precio en pantalla.
- `cierra_columna()`, retorna el código HTML para cerrar una columna.
- `cierra_fila()`, retorna el código HTML para cerrar una fila.
- `cierra_contenedor()`, retorna el código HTML para cerrar un contenedor.
- `abre_columna_2_contenedor_1()`, retorna el código HTML para crear la segunda columna en el contenedor 1.
- `titulo($titulo)`, retorna el código HTML con el título de activo.
- `descripcion_corta($descripcion_corta)`, retorna el código HTML con la descripción corta.
- `abre_columna_contenedor_2()`, retorna el código HTML de apertura de una columna en el contenedor 2.
- `abre_tabs()`, retorna el código HTML para abrir una sección de pestañas.
- `abre_un_tab($titulo)`, retorna el código HTML de apertura de una pestaña con el título pasado como argumento.
- `cierra_un_tab()`, retorna el código HTML de cierre de una pestaña.
- `cierra_tabs()`, retorna el código necesario para cerrar una sección de pestañas.
- `print_field_descripcion($descripcion)`, retorna el código para mostrar la descripción con el formato deseado.
- `print_field_desc_comparable($etiqueta, $descripcion)`, genera el código para mostrar la descripción del campo comparable con.
- `print_field_caracteristicas($caracteristicas)`, retorna el código necesario para mostrar el campo características con formato.

- `abre_fila_interna()`, retorna el código para abrir una fila interna.
- `abre_columna_interna($type, $layout, $background_color = "", $first = "false", $cab = "false")`, retorna el código necesario para abrir una columna interna.
- `abre_text()`, retorna el código para abrir un espacio para texto.
- `cierra_text()`, retorna el código para cerrar un espacio para texto.
- `cierra_fila_interna()`, retorna el código para cerrar una fila interna.
- `cierra_columna_interna()`, retorna el código para cerrar una columna interna.
- `print_cabecera($clave)`, retorna el código para imprimir la cabecera de un bloque.
- `print_bloque_lista($lista)`, retorna el código que permite imprimir un bloque de hasta 3 elementos pasados como argumentos en la matriz `$lista`.
- `print_cabecera_referencias($clave)`, retorna el código para mostrar la cabecera del bloque de referencias.
- `print_bloque_lista_referencias($lista)`, retorna el código que muestra las referencias pasadas como argumento.
- `abre_checklist($campo, $text_icon, $size_font)`, retorna el código iniciar un bloque con casillas de verificación.
- `print_checklist_item($text_item)`, retorna el código para mostrar una casilla de verificación el elemento pasado como argumento.
- `print_checklist_list($lista)`, retorna el código para mostrar una casilla de verificación para cada elemento de la lista pasada como argumento.
- `cierra_checklist() ($campo, $text_icon, $size_font)`, retorna el código cerrar un bloque con casillas de verificación.

La clase **"Mng_Gen_View"** será la encargada de gestionar la vista de un activo durante la búsqueda , para ello dispondrá de una propiedad privada:

- `$color_cabecera = '#5271ff'`, que es la encargada de establecer el color de la cabecera.

Ofrece también los siguientes métodos:

- `print_contents()`, retorna el código necesario para crear el contendor.
- `abre_contenedor()`, retorna el código de apertura de un contenedor.

- `abre_fila()`, retorna el código de apertura de una fila.
- `abre_columna($dimension)`, retorna el código de apertura de una columna, con la dimensión pasada como argumento.
- `cierra_columna()`, retorna el código de cierre de una columna.
- `cierra_fila()`, retorna el código de cierre de una fila.
- `cierra_contenedor()`, retorna el código de cierre del contenedor.
- `abre_columna_2_contenedor_1()`, retorna el código de apertura de la columna2 en el contenedor 1.
- `titulo($titulo)`, retorna el código para imprimir el título con formato.
- `descripcion_corta($descripcion_corta)`, retorna el código de impresión de la descripción corta.
- `abre_columna_contenedor_2()`, retorna el código de apertura de la columna en el contenedor 2.
- `abre_tabs()`, retorna el código HTML para abrir una sección de pestañas.
- `abre_un_tab($titulo)`, retorna el código HTML de apertura de una pestaña con el título pasado como argumento.
- `cierra_un_tab()`, retorna el código HTML de cierre de una pestaña.
- `cierra_tabs()`, retorna el código necesario para cerrar una sección de pestañas.
- `print_tab_descripcion($descripcion, $caracteristicas)`, retorna el código de la pestaña descripción con los valores pasados como argumentos.
- `print_field_descripcion($descripcion)`, retorna el código para imprimir el campo descripción.
- `print_field_desc_comparable($etiqueta, $descripcion)`, genera el código para mostrar la descripción del campo comparable con.
- `print_field_caracteristicas($caracteristicas)`, retorna el código necesario para mostrar el campo características con formato.
- `abre_fila_interna()`, retorna el código para abrir una fila interna.
- `abre_columna_interna($type, $layout, $background_color = "", $first = "false", $cab = "false")`, retorna el código necesario para abrir una columna interna.
- `abre_text()`, retorna el código para abrir un espacio para texto.
- `cierra_text()`, retorna el código para cerrar un espacio para texto.
- `cierra_fila_interna()`, retorna el código para cerrar una fila interna.

- `cierra_columna_interna()`, retorna el código para cerrar una columna interna.
- `print_cabecera($clave)`, retorna el código para imprimir la cabecera de un bloque.
- `print_bloque_lista($lista)`, retorna el código que permite imprimir un bloque de hasta 3 elementos pasados como argumentos en la matriz `$lista`.
- `print_cabecera_referencias($clave)`, retorna el código para mostrar la cabecera del bloque de referencias.
- `print_bloque_lista_referencias($lista)`, retorna el código que muestra las referencias pasadas como argumento.
- `abre_checklist($campo, $text_icon, $size_font)`, retorna el código iniciar un bloque con casillas de verificación.
- `print_checklist_item($text_item)`, retorna el código para mostrar una casilla de verificación el elemento pasado como argumento.
- `print_checklist_list($lista)`, retorna el código para mostrar una casilla de verificación para cada elemento de la lista pasada como argumento.
- `cierra_checklist() ($campo, $text_icon, $size_font)`, retorna el código cerrar un bloque con casillas de verificación.

La clase **"Mng_Search_Cond_View"** es la encargada de gestionar la vista del formulario de condiciones para la búsqueda personalizada. Dispone de un único método:

- `banda_lateral()`, retorna el código necesario para mostrar el formulario de búsqueda personalizada que se muestra en la banda lateral de la página.

La clase **"Mng_Search_Results_View"** es la encargada visualizar los resultados de una búsqueda. Dispone de 2 métodos:

- `label_box($label)`, retorna el código para visualizar una etiqueta de un campo de búsqueda.
- `field_box($value)`, retorna el código para visualizar el valor de un campo de búsqueda.

Siguiendo las prácticas recomendadas se ha intentado concentrar la mayor parte de las funcionalidades desarrolladas dentro del complemento **"activos_mng"** con el objetivo de hacerlo lo más independiente posible del tema usado para nuestro trabajo, pero existen algunas tareas de desarrollo que deben realizarse sobre el propio tema debido a que es el encargado de presentar el contenido, gestionando las peticiones del usuario y actuando como controlador.

Para evitar que las actualizaciones del tema Avada puedan afectar a los desarrollos realizados sobre el mismo, se creará un tema hijo denominado "Avada-Child-Theme" sobre el que se va a realizar la tarea de desarrollo. Esto nos va permitir conservar todos los cambios realizados sobre el tema original y mantener actualizado el tema Avada de manera segura. El tema hijo hereda las funcionalidades del tema principal, pero nos permitirá ampliarlas con nuestro desarrollo.

El archivo "**functions.php**" del tema se comporta como un complemento de WordPress y es donde se pueden añadir funcionalidades, llamando a funciones de WordPress o definiendo nuestras propias funciones. Un tema hijo puede tener su propio "funtions.php" [21]

Los temas WordPress utilizan lo que se denominan plantillas, las plantillas que existen dentro de un tema expresan cómo se visualiza el sitio Web. Las plantillas de página son aquellas que se aplican solo a las páginas para cambiar su apariencia. Una plantilla de página se puede aplicar a una sola página, una sección de página o una clase de páginas. La jerarquía de plantillas es la lógica que usa WordPress para decidir qué archivo de plantilla del tema usar, según el contenido que se solicita. [22]

Una plantilla parcial es una parte de una plantilla que se incluye como parte de otra plantilla, como el encabezado de un sitio. Los parciales de plantilla se pueden incrustar en varias plantillas, lo que simplifica la creación de temas.

Para poder visualizar correctamente los contenidos de tipo activo hemos creado dentro del tema hijo los ficheros functions.php, style.css (diseño propio del hijo) y send-post-fields.php, así como las siguientes plantillas:

1. archive-activo.php
2. header-activo.php
3. page-search.php
4. single-activo.php
5. template-parts
 - 5.1. content-activo.php
 - 5.2. search-activo.php

Desde los archivos de las platillas parciales hacemos uso tanto de las clases del modelo, como de las clases de la vista para poder representar de la forma deseada los activos.



Figura 5.2 Diseño de clases

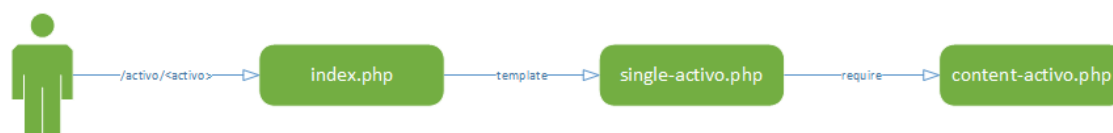


Figura 5.3 Visualizar un activo

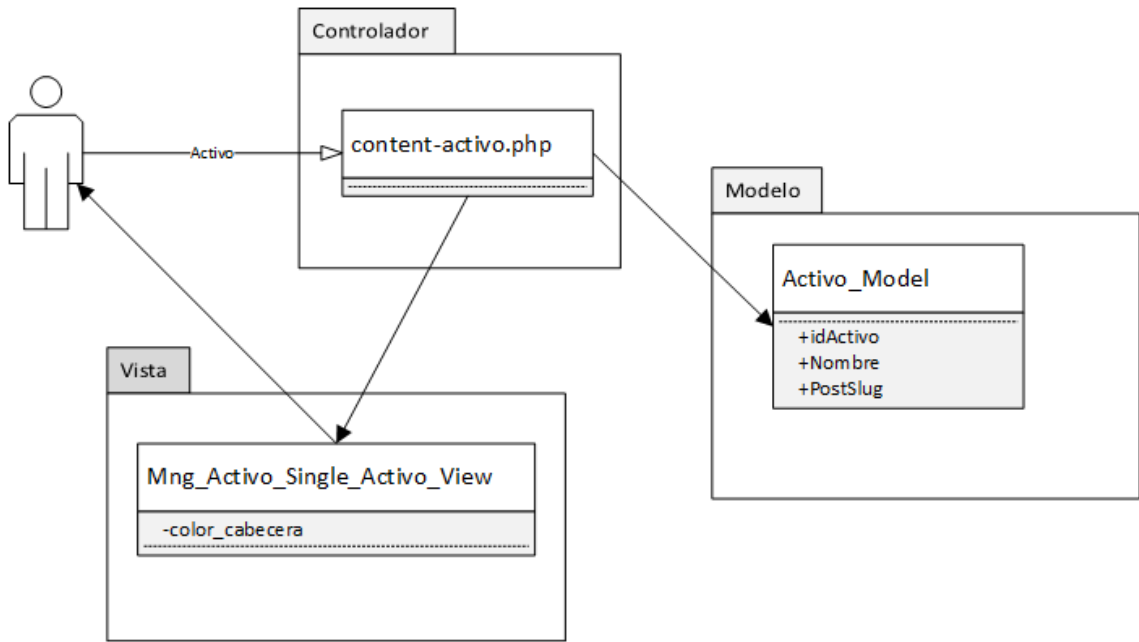


Figura 5.4 Diagrama de clases de diseño para visualizar un activo

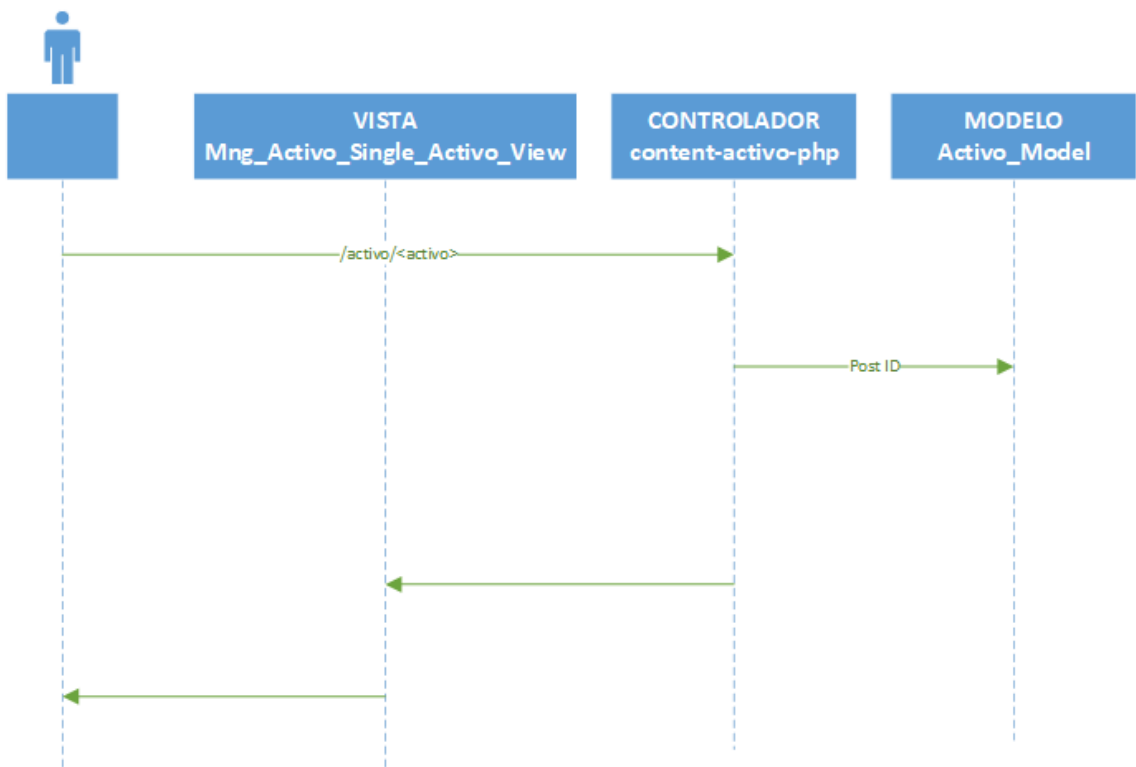


Figura 5.4 Diagrama de secuencia para visualizar un activo

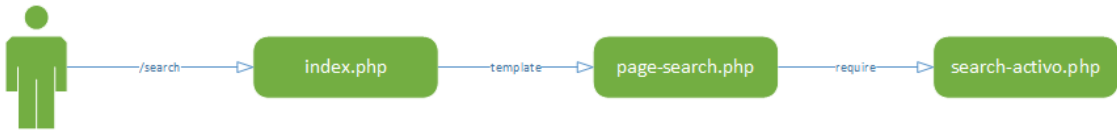


Figura 5.5 Buscar activos

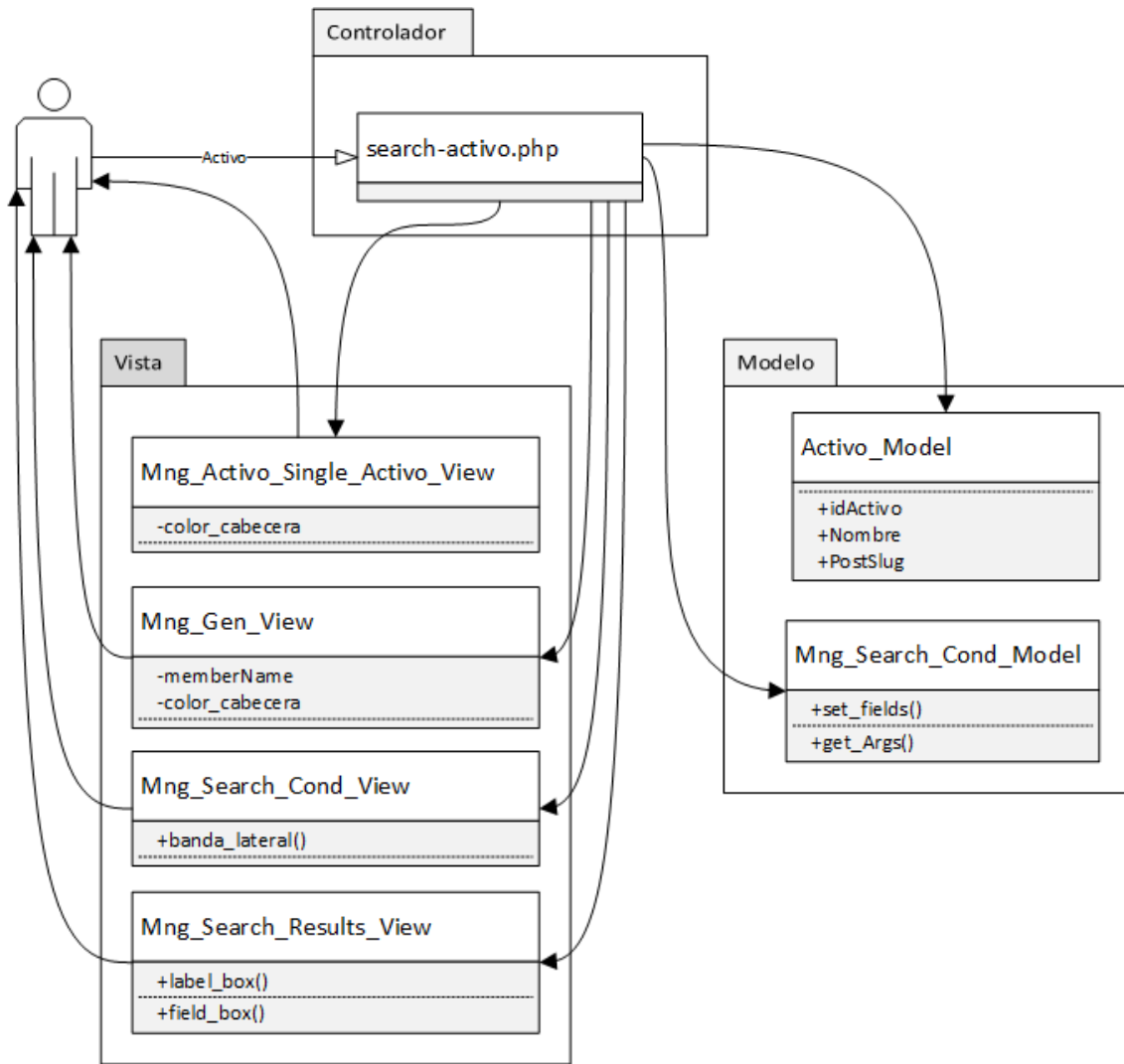


Figura 5.6 Diagrama de clases de diseño para la página de búsqueda /search

6

Implementación

En este capítulo se recogen las tareas propias de programación llevadas a cabo con el objeto de cumplir con los requisitos funcionales propuestos y el diseño técnico generado para la realización del hub de activos.

Por un lado se describe cómo se ha dividido el trabajo y cómo se ha abordado la solución de las dificultades encontradas a lo largo del desarrollo, ejemplos de código utilizados que pueden facilitar la comprensión de esta fase del proyecto y por último algunas pantallas con el resultado final.

Por último se recoge cómo se han introducido los contenidos en el sitio Web.

6.1 Desarrollo

Para este sitio se ha creado una página principal utilizando el diseñador de páginas con formato visual, *fusion builder*, incluido dentro del tema Avada que facilita la tarea de diseño debido a que permite arrastrar elementos dentro del espacio de diseño. En esta página principal, cuyo aspecto y diseño podemos ver en la figura 6.1, se describe qué es un activo tecnológico y por qué este sitio es un *hub* de esos activos.



Figura 6.1 Página principal de www.tiactivos.com



Figura 6.2 Qué es un Activo tecnológico

También en esta página principal se recogen los distintos perfiles a los que está dirigido este sitio e información acerca de los tipos de proyectos para los que puede resultar útil este *hub* de activos.



Figura 6.3 Tipos de usuarios del sitio

Dentro del sitio, hemos incluido un blog en el que poder publicar entradas de interés referidas a los activos, siendo los contenidos que existen en la actualidad el punto de partida, para que una vez hecho público el sitio se pueda ir completando.



Figura 6.4 Blog de www.tiactivos.com



Figura 6.5 Entrada en el Blog de www.tiactivos.com

Si bien aún están pendientes de revisión los contenidos, hemos alcanzado el objetivo de crear una herramienta plenamente funcional en las asignaturas impartidas por el profesor.

Una vez instalado y configurado el entorno de desarrollo, la primera funcionalidad que debíamos implementar es el alta y la modificación de un activo, ya que debía estar disponible y operativa en marzo de 2021, de forma que fuese posible la introducción de los activos por parte de los alumnos de la asignatura Calidad del Software impartida por el profesor. Para llevar a cabo la implementación del alta, modificación y borrado de activos se han utilizado los complementos de WordPress Custom Post Type UI y Advanced Custom Fields PRO. Durante esta fase nos hemos centrado en crear los campos personalizados del tipo "Activo" conforme a las especificaciones proporcionadas.

Para poder conseguir la interfaz de introducción de los campos personalizados que componen el activo conforme a lo reflejado en la figura 6.3, se ha incluido dentro de cada campo personalizado un valor para el atributo ***class*** que nos permite asociar un estilo CSS y de esta forma controlar el aspecto que presenta el campo personalizado dentro del formulario de creación y edición de un activo.

Los estilos CSS para los campos personalizados del activo están definidos dentro del fichero "activos-mng\admin\css\activos-mng-admin.css" del plugin activos-mng.

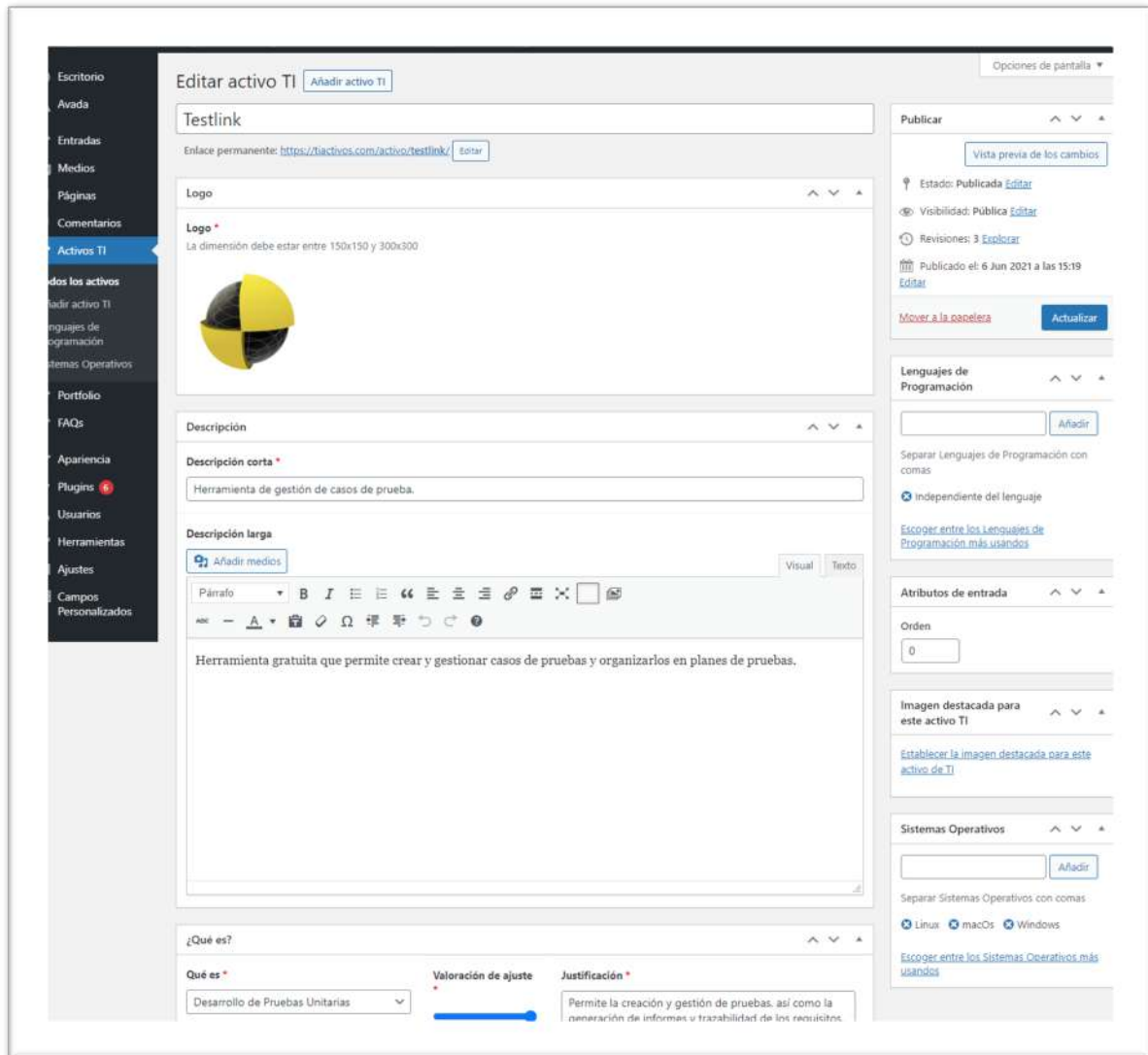


Figura 6.3 Formulario de creación/edición de un activo

Adicionalmente, durante la implementación de esta funcionalidad se hizo necesaria la creación de un rol específico denominado **EDITORACTIVOS** que solo tuviese acceso a la creación de nuevos activos y a la modificación y eliminación de aquellos activos creados por ese usuario. Esta configuración se ha realizado con el complemento User Role Editor.

El siguiente aspecto a desarrollar durante la implementación consistió en diseñar la interfaz de usuario. Para abordar esta fase se utilizó el complemento del tema Avada para realizar el diseño deseado y analizando los códigos cortos (*shortcodes*) creados por el tema para incorporarlos a los métodos de la clase encargada de la vista del plugin.

En el ejemplo de código fuente que podemos ver a continuación se muestra cómo el método "**titulo**" de la clase `Mng_Activo_Single_Activo_View` retorna el código necesario

para representar el título del activo con el formato y diseño deseado.

```
public function titulo($titulo)
{
    $html_codigo = '
        [fusion_title title_type="text" rotation_effect="bounceIn" display_time="1200"
highlight_effect="circle" loop_animation="off" highlight_width="9"
highlight_top_margin="0" before_text="" rotation_text="" highlight_text="" after_text=""
hide_on_mobile="small-visibility,medium-visibility,large-visibility"
sticky_display="normal,sticky" class="" id="" content_align_medium=""
content_align_small="" content_align="left" size="3" font_size="" animated_font_size=""
fusion_font_family_title_font="" fusion_font_variant_title_font="" line_height=""
letter_spacing="" text_shadow="no" text_shadow_vertical="" text_shadow_horizontal=""
text_shadow_blur="0" text_shadow_color="" margin_top_medium=""
margin_bottom_medium="" margin_top_small="" margin_bottom_small=""
margin_top="0" margin_bottom="" margin_top_mobile="" margin_bottom_mobile=""
text_color="" animated_text_color="" highlight_color="" style_type="none" sep_color=""
animation_type="" animation_direction="left" animation_speed="0.3"
animation_offset="" ]<p>' . $titulo .
        '</p>[/fusion_title]';
    return $html_codigo;
}
```

Usar esta codificación propia de Avada puede parecer compleja para el proceso de construcción de la Web, pero todo lo contrario. Para obtener este código el elemento gráfico deseado simplemente se diseña con "fusion builder" de forma visual, con un editor del tipo WYSIWYG (*what you see is what you get*). Cuando el elemento tiene el formato y aspecto deseado, simplemente se copia la codificación que la herramienta proporciona, y se utiliza directamente en el código fuente como se muestra arriba.

Por otra parte, y gracias al diseño técnico que se ha creado, separando la Vista del Modelo y del Controlador, en el caso de no existir el tema Avada es posible sustituir estos códigos cortos directamente por HTML, sin con ello afectar al resto de las partes de la arquitectura.

Una vez desarrollada la vista encargada de representar el diseño gráfico con el aspecto deseado, el siguiente paso consiste en mostrar la información almacenada para un activo. Para poder acceder a esa información se desarrolló la clase encargada del modelo de datos **Activo_Model** que permite acceder a través de sus métodos a los campos

personalizados de un activo y retornar la información contenida en él.

En este ejemplo podemos ver cómo se obtiene el valor almacenado en un campo personalizado del activo:

```
function getValor($fieldName)
{
    // Retorna el valor de contenido en el campo
    $value = get_field("$fieldName");
    if ($value) {
        return $value;
    } else {
        return "";
    }
}
```

Para poder conseguir una presentación completa dónde se integren la vista y el modelo es necesario desarrollar las funciones de controlador que en nuestro caso y gracias a las plantillas parciales hemos ubicado en el fichero "**content-activo.php**". Este fichero es el encargado de mostrar el contenido de un activo y por tanto utilizamos la clase encargada de la vista y la encargada del modelo, para representar los datos con el formato deseado, consiguiendo el aspecto que podemos ver en la figura 6.4.

Para desarrollar un buscador personalizado de activos, hemos tenido que afrontar varios retos debido a que el sistema de búsqueda implementado en WordPress no nos ofrece la flexibilidad necesaria.

En primer lugar, hemos tenido que crear una página Web dentro del sitio denominada "**search**" para implementar las búsquedas personalizadas de activos. Eso nos ha permitido crear una plantilla parcial denominada "**search-activo.php**" que será la encargada de acoger el código necesario para poder mostrar y gestionar las búsquedas.

A continuación, y haciendo uso de la clase incorporada en **WP_Query** podemos controlar y personalizar los resultados de la búsqueda. Para mostrar los resultados de la búsqueda de activos hemos desarrollado una clase denominada **Mng_Search_Results_View** y para gestionar el modelo de datos

encargado de filtrar los resultados en función de las condiciones de búsqueda establecidas por el usuario hemos desarrollado la clase **Mng_Search_Cond_Model**.

En esta clase hemos tenido que afrontar algunos desafíos, como son el mapeo de los campos de búsqueda debido a que el nombre de los campos en el formulario de búsqueda no coincide con el nombre de campo en el activo, así pues gracias al método `set_fields($que_es = null, $precio = null, $tipo_activo = null, $tipo_proyecto = null, $perfil_prof = null, $actividades = null, $area_cmmi = null, $obj_cmmi = null, $pmbok = null, $procesosITIL = null)` se genera una matriz con los nombres de campo en el activo y el valor introducido por el usuario para el campo en el formulario de búsqueda.

El método `get_Args($num_posts_per_pag, $act_page, $search_fields = array())` es el encargado de generar una matriz de argumento compatible con la clase `WP_QUERY` de WordPress.

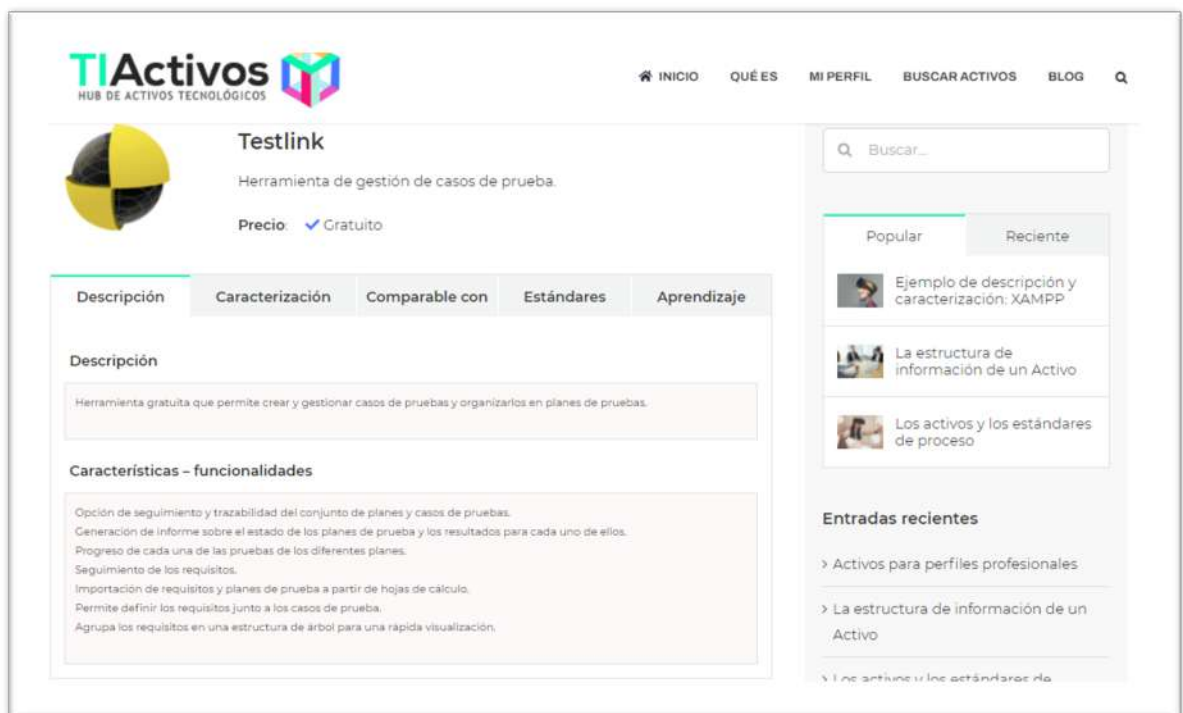


Figura 6.4 Presentación del contenido de un activo

Para mostrar las condiciones y filtros personalizados de búsqueda se ha creado un formulario, en forma de barra de lateral, que se incluye dentro de la página de búsqueda. Para gestionar la vista de esta barra hemos desarrollado la clase **Mng_Search_Cond_View**.

En relación a la barra de búsqueda hemos tenido que solventar algunas dificultades. Dado que el contenido del formulario de búsqueda se envía a través de `$_POST`, al mostrar los resultados de la búsqueda y generar la pantalla que podemos ver en la figura 6.5.

Dado que debemos mantener los criterios de búsqueda seleccionados por el usuario en el formulario de esa barra lateral, se ha desarrollado un fichero PHP denominado "**send-post-fields.php**" encargado de pasar los valores que hay en `$_POST`, desde el escritorio hasta el interfaz de usuario, pasando a estar disponibles en JavaScript.

Esto lo hacemos a través de un script que carga en la variable JavaScript "**ti_arr_cond_field**" una matriz con los valores de los campos del formulario de búsqueda. Adicionalmente, en el fichero "**activos-search.js**" se ha desarrollado una función que se ejecuta durante la carga de la página de búsqueda de activos y recorre la matriz con todos los campos detectando aquellos que tienen un valor y que por tanto el usuario consignó en el formulario de búsqueda de la barra lateral y trasladando el valor al formulario actual.

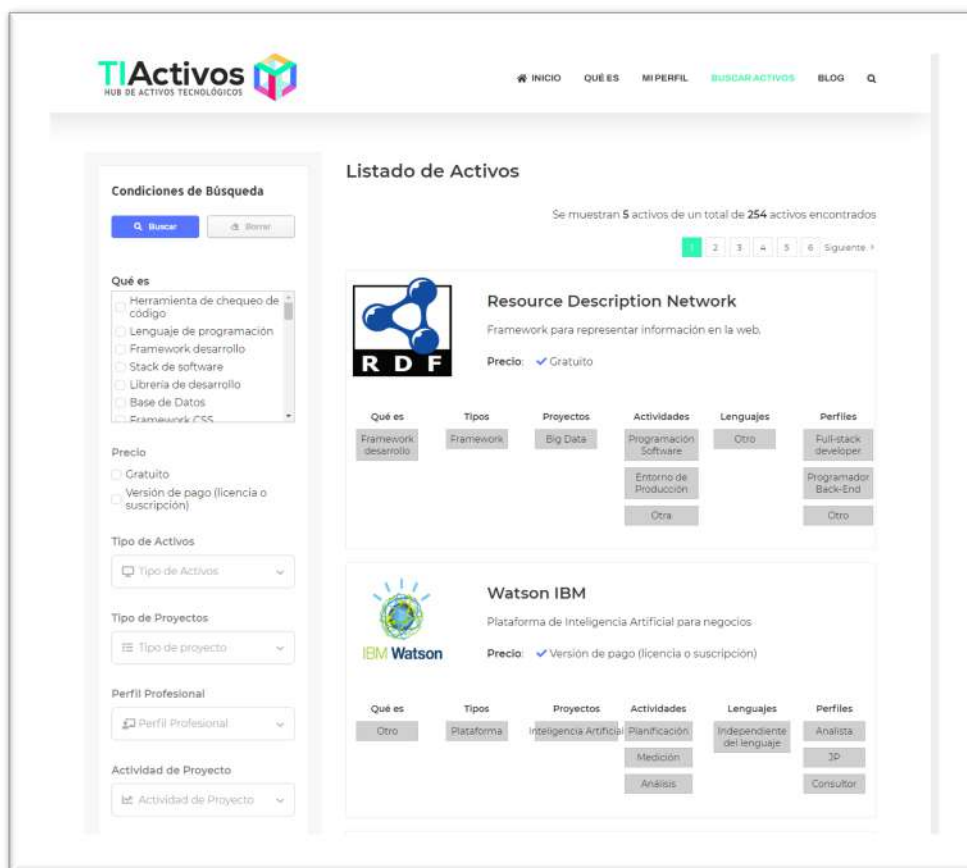


Figura 6.5 Pantalla de búsqueda

Los resultados de búsqueda se muestran paginados a razón de 5 activos por página para facilitar la usabilidad al usuario. Para poder ofrecer esa funcionalidad es necesario mostrar los botones de paginación.

Aunque la paginación es una característica propia del tema del WordPress, para nuestro caso su implementación no es trivial, debido a que deben enviarse los valores del formulario de condiciones de búsqueda ubicado en la barra lateral vía post.

Para solucionar esta situación hemos desarrollado una función JavaScript "**ti_pagination(pageNum)**" que se encarga de actualizar el valor del parámetro "search_page" con la página en la que ha hecho clic el usuario y después envía en formulario de condiciones de búsqueda usando el método post.

TIActivos
HUB DE ACTIVOS TECNOLÓGICOS

INICIO QUE ES MI PERFIL **BUSCAR ACTIVOS** BLOG

Condiciones de Búsqueda

Buscar Borrar

Qué es

- Herramienta de chequeo de código
- Lenguaje de programación
- Framework desarrollo
- Stack de software
- Librería de desarrollo
- Base de Datos
- Framework CSS

Precio

- Gratuito
- Versión de pago (licencia o suscripción)

Tipo de Activos

Software

Tipo de Proyectos

Tipo de proyecto

Perfil Profesional

Perfil Profesional

Actividad de Proyecto

Actividad de Proyecto

Área de Proceso CMMI

Área de Proceso CMMI

Objetivos Específicos CMMI

Objetivos específicos CM

Proceso PMBOK

Proceso PMBOK

Gestión del Servicio TI

Gestión del servicio TI

Listado de Activos

Se muestran 5 activos de un total de 38 activos encontrados

Anterior 2 3 4 5 6 7 8 Siguiente

Avast Antivirus

Antivirus con una opción totalmente gratuita que también presenta versiones de pago

Precio: Gratuito Versión de pago (licencia o suscripción)

Qué es	Tipos	Proyectos	Actividades	Lenguajes	Perfiles
Herramienta de Seguridad	Software	Gestión del Servicio	Gestión de Servicio TI	C++	Otro

Katalon

Solución de testing automatizado.

Precio: Gratuito Versión de pago (licencia o suscripción)

Qué es	Tipos	Proyectos	Actividades	Lenguajes	Perfiles
Desarrollo y prueba de APIs	Software	Producción de Software	Pruebas	Groovy	Tester
Pruebas de Software				Java	Especialista de Pruebas

Mattermost

Servicio de chat en línea en código abierto y auto hospedable.

Precio: Gratuito Versión de pago (licencia o suscripción)

Qué es	Tipos	Proyectos	Actividades	Lenguajes	Perfiles
Herramienta de comunicación	Software	Producción de Software	Gestión de Servicio TI	Cualquiera	Otro
	Solución	Marketing Digital	Gestión de Proyecto		
			Planificación		

Dropbox

Servicio de alojamiento de archivos multiplataforma en la nube.

Precio: Gratuito Versión de pago (licencia o suscripción)

Qué es	Tipos	Proyectos	Actividades	Lenguajes	Perfiles
Herramienta de comunicación	Software	Otro	Operaciones	Cualquiera	Especialista DevOps
Entorno de trabajo colaborativo	Solución		DevOps		Gerente
			Otro		Otro

Git

Git es un software de control de versiones

Precio: Gratuito Versión de pago (licencia o suscripción)

Qué es	Tipos	Proyectos	Actividades	Lenguajes	Perfiles
Sistema de control de versiones	Software	Producción de Software	Integración Continua		Full-stack developer
			Entrega Continua		Especialista DevOps
			Despliegue Continuo		

Anterior 2 3 4 5 6 7 8 Siguiente

Figura 6.6 Búsqueda con varias condiciones y páginas de resultados.

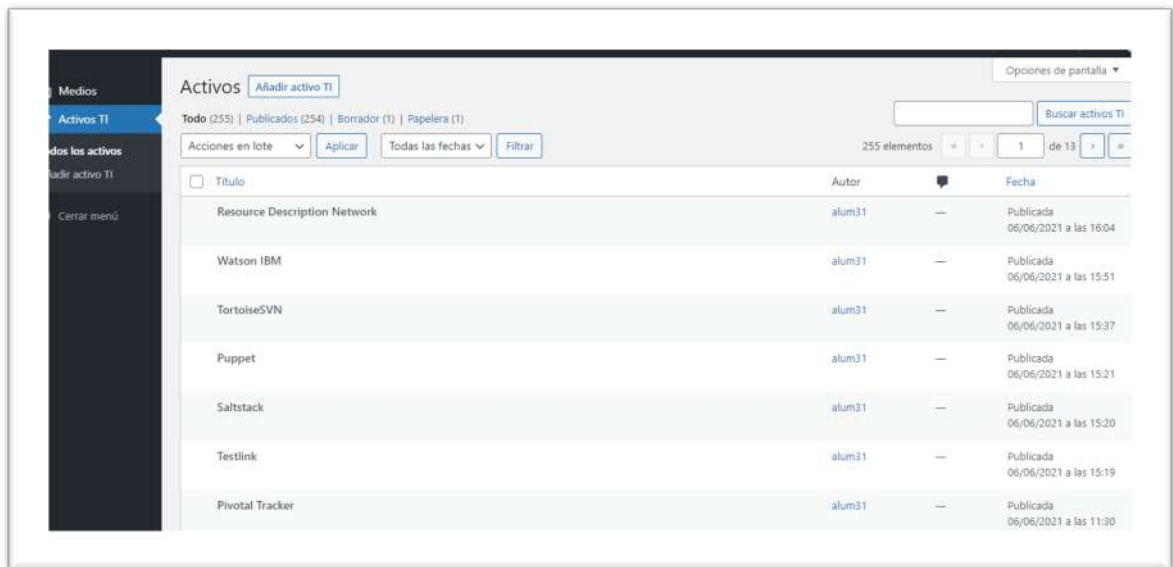
6.2 Introducción de contenidos

La introducción de activos en el sitio Web se ha llevado a cabo por los alumnos de la asignatura calidad del software, como una parte práctica de la misma.

El profesor les ha facilitado un listado de activos a los alumnos, se les ha explicado la estructura de un activo y cada uno de ellos ha seleccionado hasta 10 activos para buscar información e introducirla en la Web.

Para ello se ha creado una cuenta de usuario para cada alumno con el rol de **EDITORACTIVOS** que les ha dado acceso al interfaz de administración de activos, una vez identificados en el sitio Web.

Tras la identificación los alumnos han podido ver todos los activos creados por otros alumnos, pero no editar ni borrar aquellos activos que no han creado ellos mismos. La única opción disponible en estos casos es poder ver el activo.



The screenshot shows a web interface for managing assets. The main heading is 'Activos' with a sub-link 'Añadir activo TI'. Below this, there are filters for 'Todo (255)', 'Publicados (254)', 'Borrador (1)', and 'Papelera (1)'. There are also buttons for 'Acciones en lote', 'Aplicar', 'Todas las fechas', and 'Filtrar'. A search bar is present with the text 'Buscar activos TI'. The table below lists several assets with columns for 'Título', 'Autor', and 'Fecha'.

<input type="checkbox"/>	Título	Autor	Fecha
<input type="checkbox"/>	Resource Description Network	alum31	Publicada 06/06/2021 a las 16:04
<input type="checkbox"/>	Watson IBM	alum31	Publicada 06/06/2021 a las 15:51
<input type="checkbox"/>	TortoiseSVN	alum31	Publicada 06/06/2021 a las 15:37
<input type="checkbox"/>	Puppet	alum31	Publicada 06/06/2021 a las 15:21
<input type="checkbox"/>	Saltstack	alum31	Publicada 06/06/2021 a las 15:20
<input type="checkbox"/>	Testlink	alum31	Publicada 06/06/2021 a las 15:19
<input type="checkbox"/>	Pivotal Tracker	alum31	Publicada 06/06/2021 a las 11:30

Figura 6.6 Pantalla todos los activos

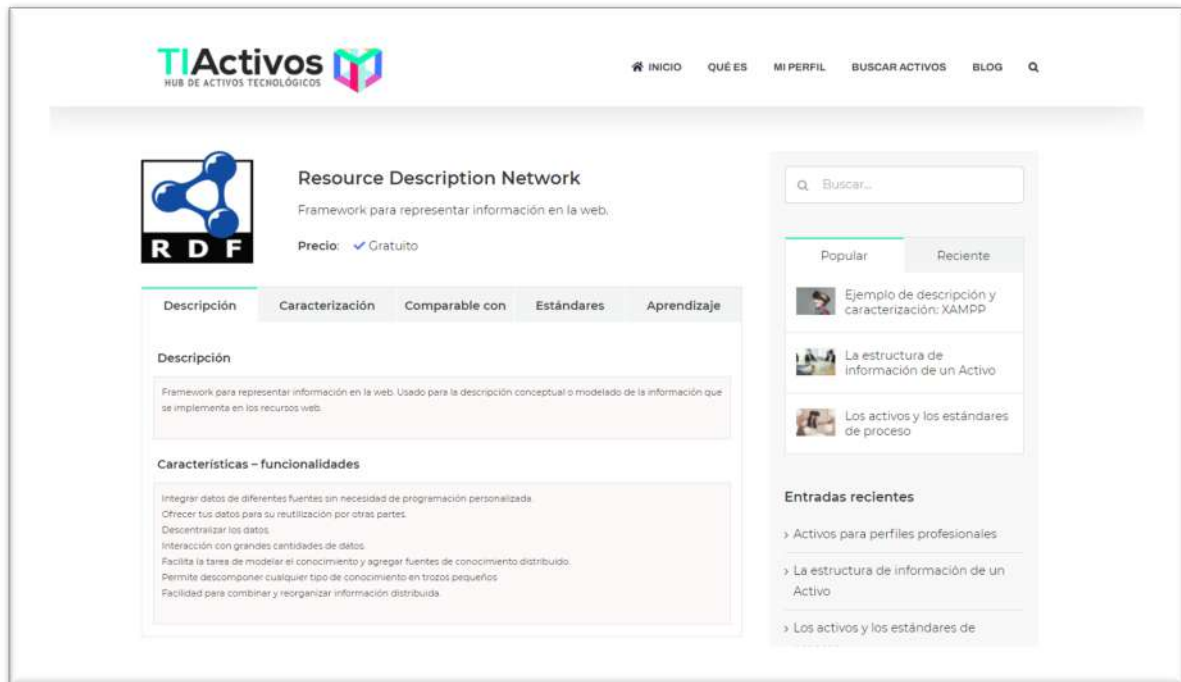


Figura 6.7 Pantalla ver un activo

Los activos actualmente introducidos en la Web se repartieron entre los alumnos, de forma que cada uno de los alumnos creó a través del formulario de alta de nuevos activos, figura 6.8, todos los activos asignados, introduciendo inicialmente únicamente los campos obligatorios y quedando como tarea realizar una documentación previa a completar el resto de campos del activo.

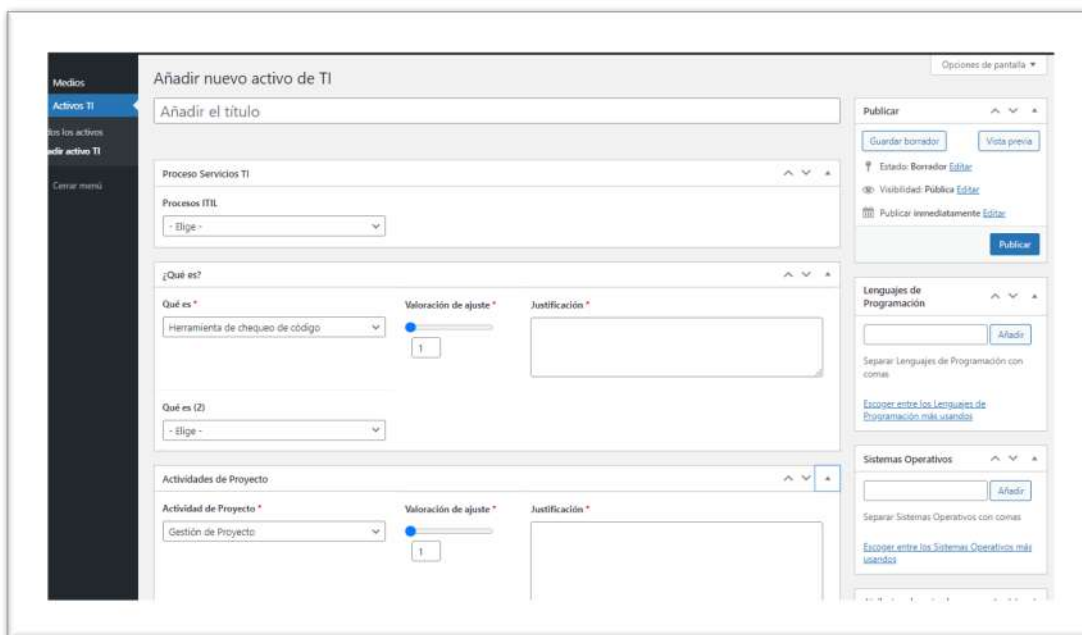


Figura 6.8 Pantalla añadir nuevo activo TI

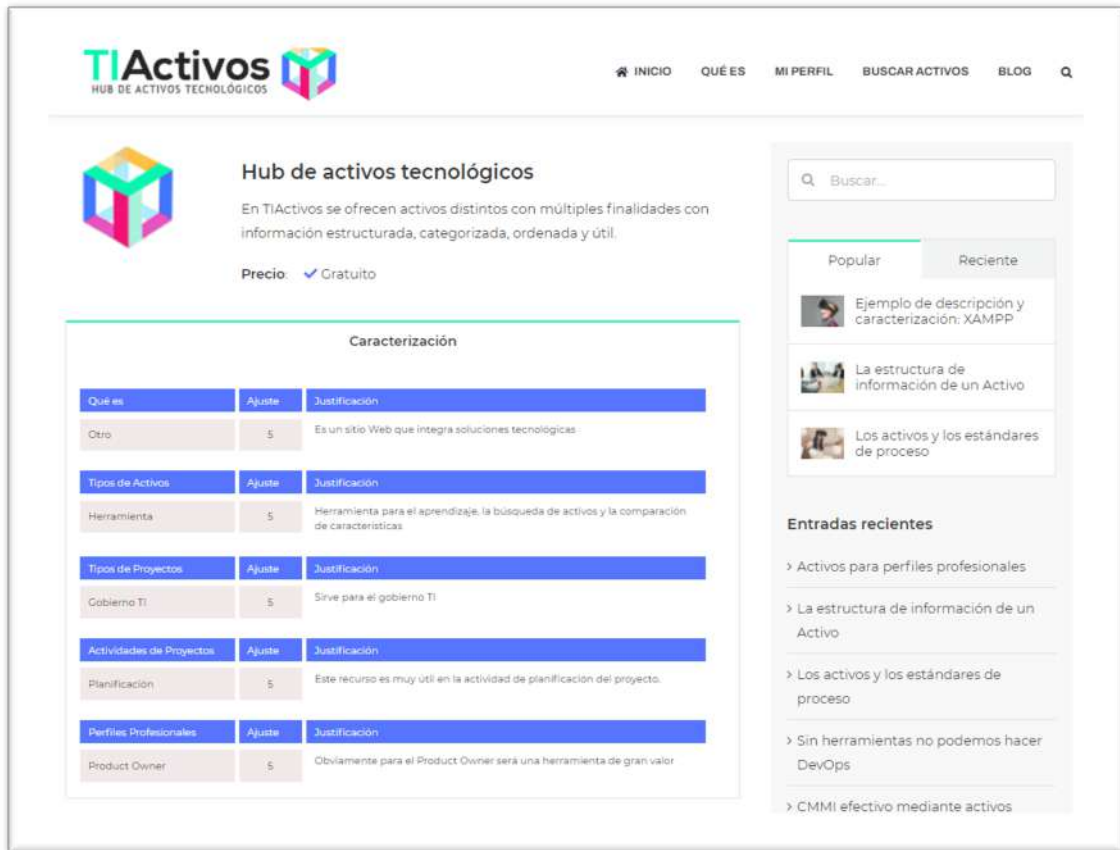


Figura 6.9 Pantalla de activo con los campos requeridos

Los activos creados por el usuario pueden listarse en la página de administración en el apartado "Mío" y ofrecerán, además de la opción ver, la opción editar y enviar a la papelera.

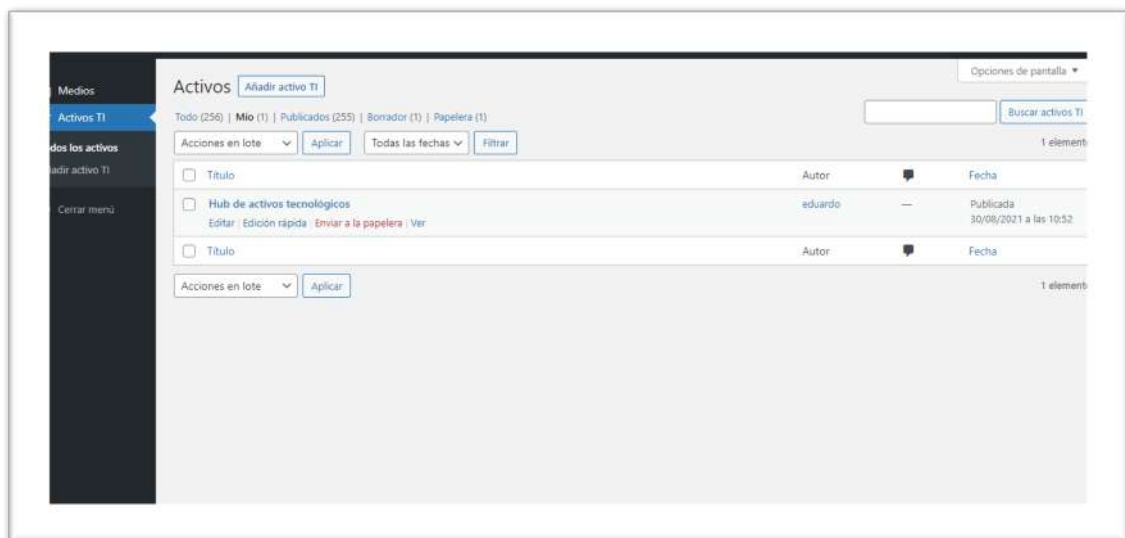


Figura 6.10 Pantalla activos creados por un USUARIOEDITOR



Pruebas

Una vez finalizada la implementación las funcionalidades y especificaciones técnicas descritas para este trabajo fin de grado, se han realizado los casos de pruebas a fin de comprobar el correcto funcionamiento del sitio Web, la detección y en su caso depuración de errores con el objetivo de garantizar la calidad del desarrollo.

Se hace la observación de que el objeto del trabajo fin de grado era disponer de una primera versión de la herramienta donde la funcionalidad estuviera implementada, y pudiera ser ya usada en asignaturas del profesor, como se ha hecho ya en el curso 2020-21. En los planes del tutor de este trabajo fin de grado está revisar el contenido del Hub de activos, y mejorar algunos aspectos de diseño y ajustes en los formatos que están pendientes de realizar, antes de que la Web sea publicada.

En cualquier caso, en este trabajo se recogen las pruebas básicas que se han realizado, para de esta forma, haber recorrido todas las fases de la ingeniería del software para la plataforma construida.

7.1 Acceso a la Web

Este caso contempla tres posibilidades que se describen a continuación, en función del actor.

7.1.1 Acceso a web pública

Objetivo:	Visualizar una página pública del sitio Web
Precondiciones	Se pide al servidor una página existente.
Postcondiciones	Se muestra la página pedida al servidor
Actor principal	Usuario no registrado, usuario registrado con rol de administrador, usuario registrado con rol de EDITORACTIVOS

Escenario de éxito principal:

1. El usuario puede visualizar la página solicitada.

7.1.2 Acceso a panel de administración

Objetivo:	Acceder al panel de administrador de la Web
Precondiciones	Solicitud al servidor de un página de para la que es requisito ser usuario registrado y tener privilegios para ver el panel.
Postcondiciones	Se muestra el panel conforme al rol del usuario identificado
Actor principal	Usuario registrado

Escenario de éxito principal:

1. El usuario se identifica de forma correcta con una cuenta registrada.
2. El usuario tiene perfil EDITORACTIVOS y se muestra el panel de administración de activos

3. El usuario tiene perfil de administrador y se muestra el panel de administración.

Escenario de fracaso:

1. El usuario no se identifica de forma correcta, se le informa del error y tiene opción a volver a realizar la identificación.

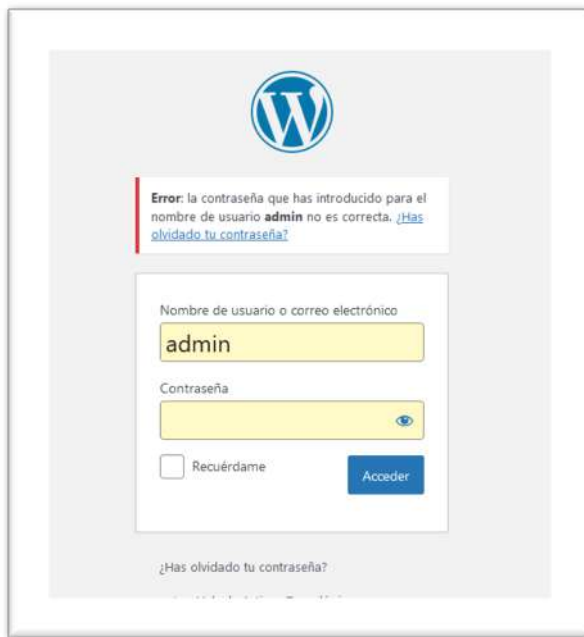


Figura 7.1 Pantalla error de identificación

7.2 Creación de un activo

Objetivo:	Crear un nuevo activo
Precondiciones	El usuario está registrado con perfil de administrador o EDITORACTIVOS
Postcondiciones	El listado de activos se refresca y muestra el nuevo activo
Actor principal	Usuario registrado

Escenario de éxito principal:

1. El usuario hace clic en Añadir activo TI.
2. El usuario completa los campos obligatorios del activo
3. El usuario hace clic en el botón publicar
4. El Activo queda registrado en el sitio Web.
5. Se muestra el listado de activos existentes

Escenario de fracaso:

1. El usuario no ha completado todos los campos obligatorios del activo.
2. El activo no se guarda.
3. Se informa al usuario de cuántos campos obligatorios faltan por completar y de destacan en color rojo.

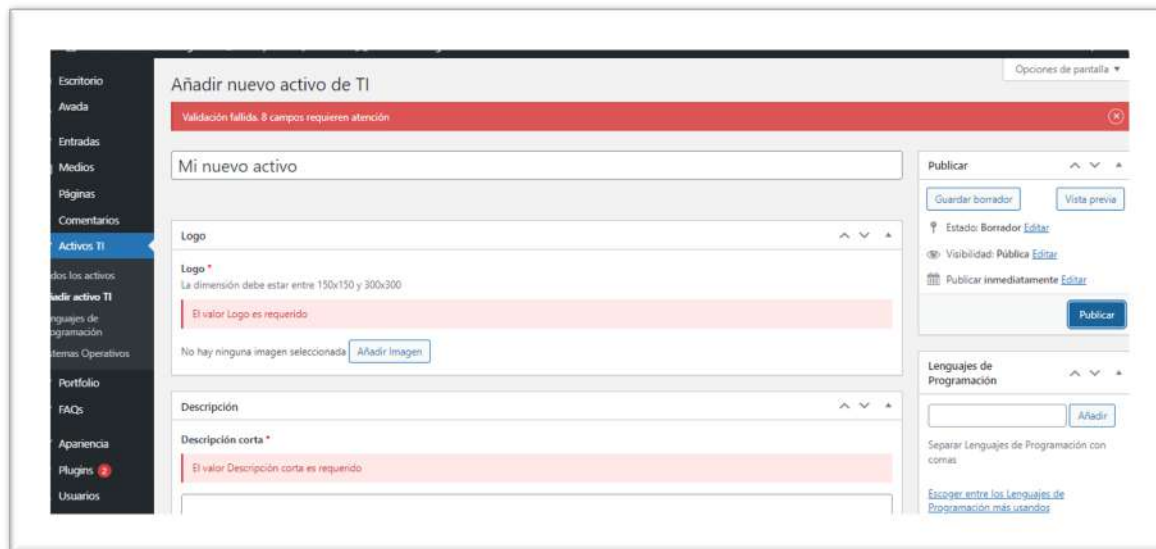


Figura 7.2 Pantalla error al crear activo

7.3 Ver un activo

Objetivo:	Visualizar un nuevo activo desde el panel del administración
Precondiciones	El usuario está registrado con perfil de administrador o EDITORACTIVOS
Postcondiciones	El actor principal puede ver también los activos de otros usuarios.
Actor principal	Usuario registrado

Escenario de éxito principal:

6. El usuario hace clic en todos los activos.
7. El usuario hace clic en **"Ver"** en la fila de un activo.
8. Se muestra el activo con el formato y diseño definido.

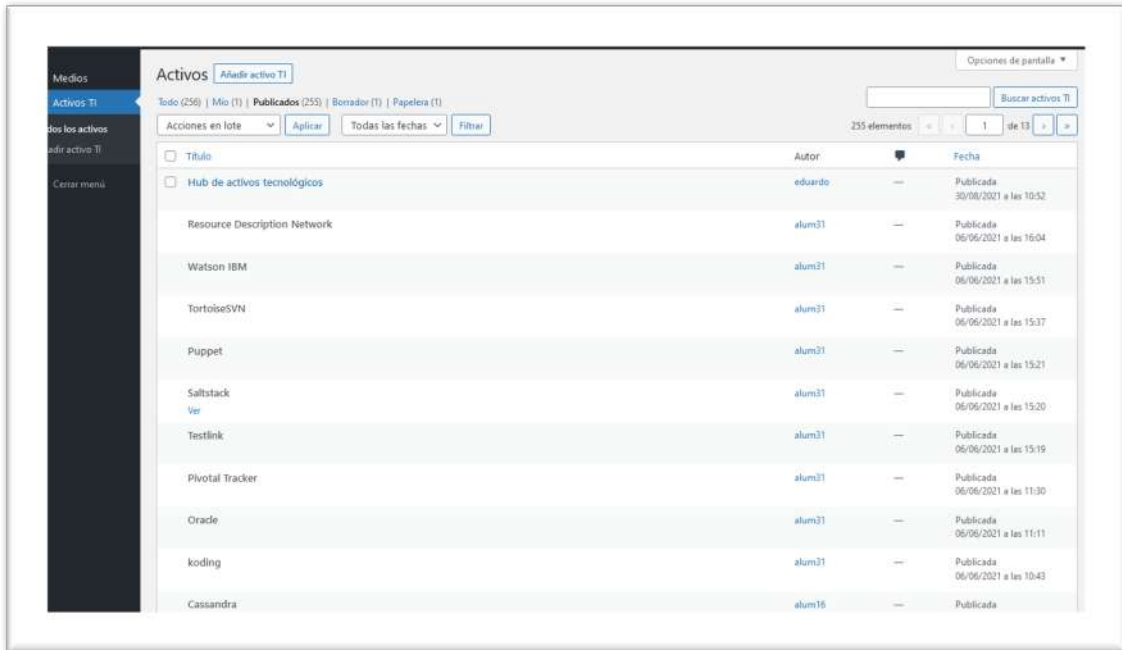


Figura 7.3 Pantalla todos los activos

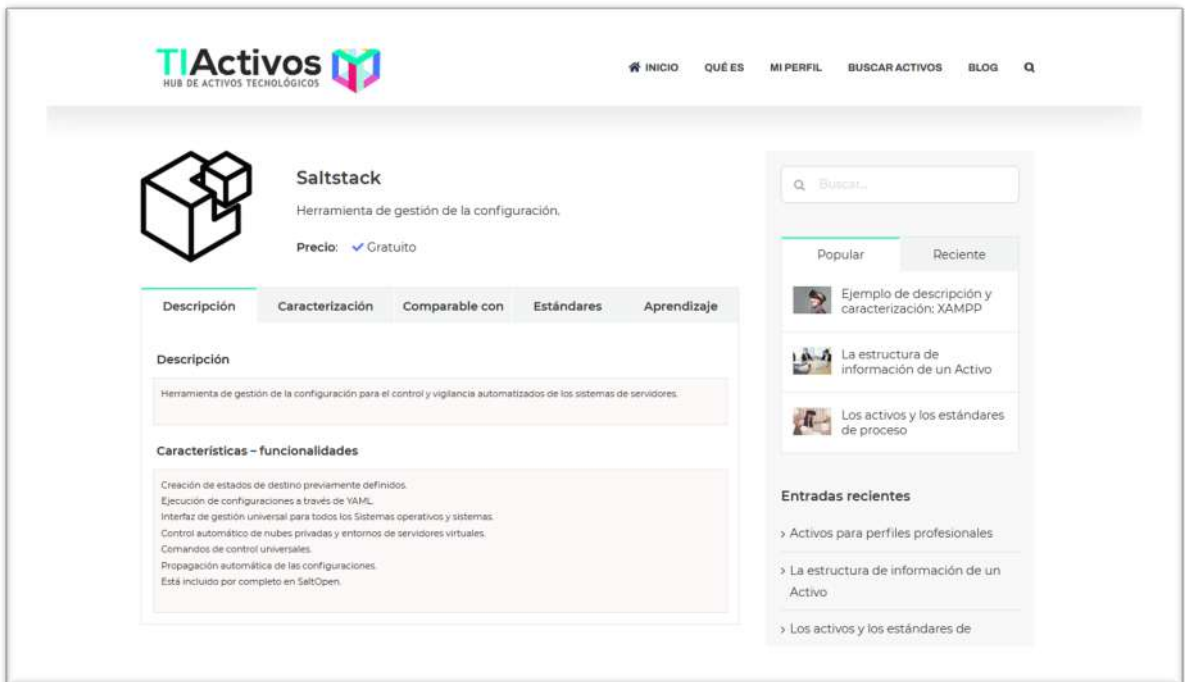


Figura 7.4 Pantalla ver activo

7.4 Modificar un activo

Objetivo:	Modificar un activo desde el panel del administración
-----------	---

Precondiciones	El usuario está registrado con perfil de administrador o EDITORACTIVOS
Postcondiciones	Si el actor principal tiene perfil EDITORACTIVOS sólo puede modificar los activos que ha creado.
Actor principal	Usuario registrado

Escenario de éxito principal:

1. El usuario hace clic en todos los activos.
2. El usuario hace clic en **"Editar"** en la fila de un activo del que es propietario o tiene perfil de administrador.
3. Se muestra la ventana de edición del activo.

Escenario de fracaso:

1. El usuario se sitúa sobre un activo propiedad de otro usuario y tiene perfil EDITORACTIVOS.
2. No aparece la opción **"Editar"**, sólo aparece la opción ver.

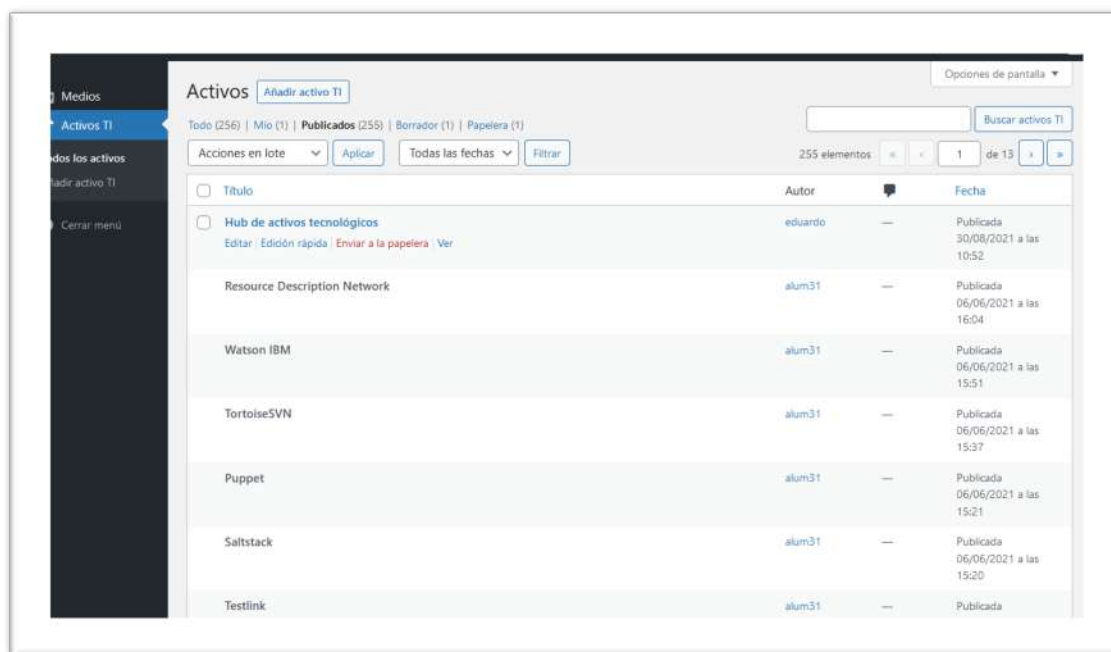


Figura 7.5 Pantalla todos los activos

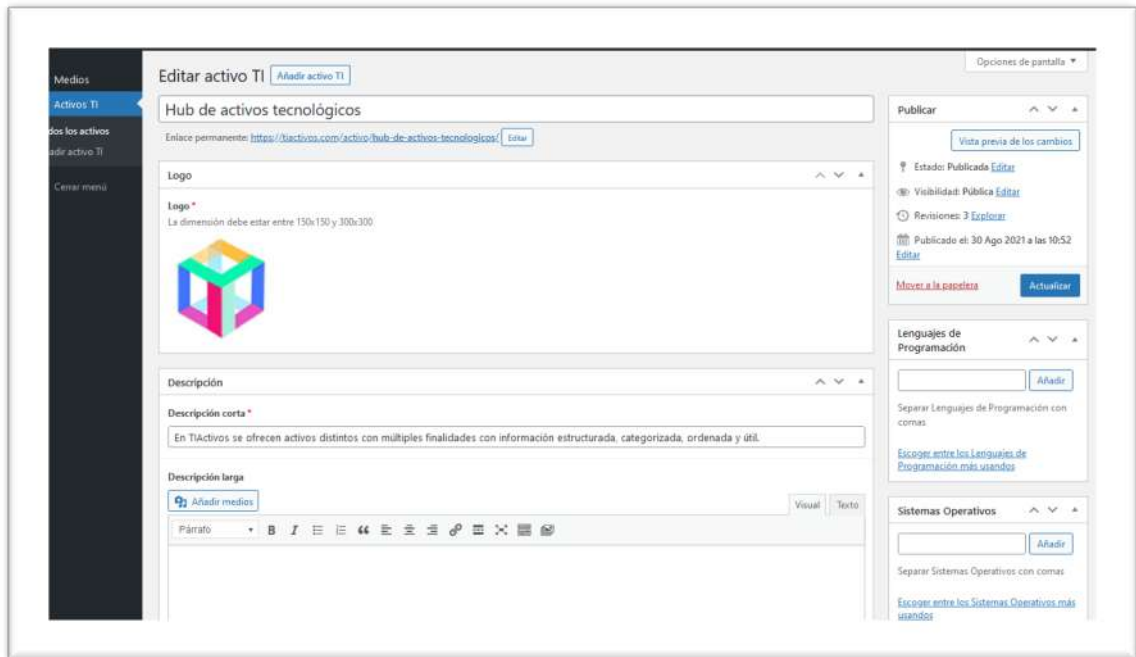


Figura 7.6 Pantalla edición de un activo

7.5 Borrar un activo

Objetivo:	Borrar un activo desde el panel del administración
Precondiciones	El usuario está registrado con perfil de administrador o EDITORACTIVOS
Postcondiciones	Si el actor principal tiene perfil EDITORACTIVOS sólo puede borrar los activos que ha creado.
Actor principal	Usuario registrado con permisos

Escenario de éxito principal:

1. El usuario hace clic en todos los activos.
2. El usuario hace clic en "**Enviar a la papelera**" en la fila de un activo.
3. El activo de borra y se envía a la papelera.

Escenario de fracaso:

1. El usuario se sitúa sobre un activo propiedad de otro usuario y tiene perfil EDITORACTIVOS.
2. No aparece la opción "**Enviar a la papelera**", sólo aparece la opción ver.

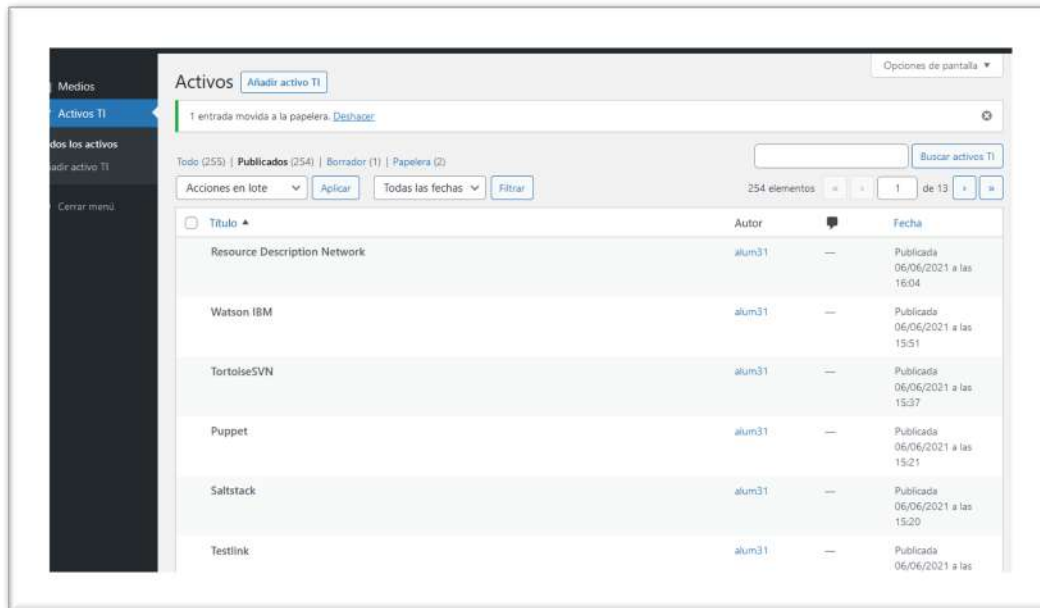


Figura 7.7 Pantalla confirmación borrado de un activo

7.6 Lenguajes programación

7.6.1 Alta de un lenguaje de programación

Objetivo:	Crear un nuevo lenguaje de programación.
Precondiciones	El usuario está registrado con perfil de administrador o EDITORACTIVOS
Postcondiciones	El nuevo lenguaje creado se añade al sistema
Actor principal	Usuario registrado con permiso administrador

Escenario de éxito principal:

1. El usuario introduce un lenguaje de programación que no existe aún.
2. El usuario hace clic en **"Añadir nuevo lenguaje de programación"**.
3. El lenguaje se añade al listado.

Escenario de fracaso:

1. El lenguaje de programación ya existe.

2. Mensaje de advertencia "Ya existe en esta taxonomía un término con el nombre y el slug facilitados."

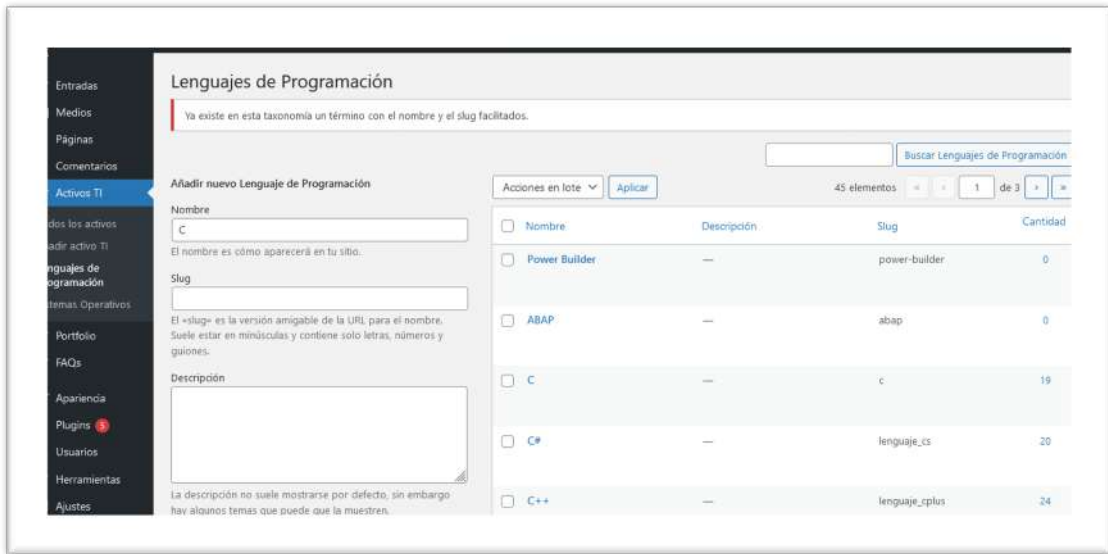


Figura 7.8 Pantalla advertencia lenguaje de programación duplicado

7.6.2 Edición de un lenguaje de programación

Objetivo:	Editar un lenguaje de programación.
Precondiciones	El usuario está registrado con perfil de administrador
Postcondiciones	Se actualizan los cambios sobre el lenguaje
Actor principal	Usuario registrado con permisos administrador

Escenario de éxito principal:

1. El usuario hace clic en "**Editar**" al situarse sobre la fila del lenguaje de programación.
2. El usuario edita los campos del lenguaje de programación.
3. El usuario hace clic en el botón "**Actualizar**".
4. Los cambios se reflejan en el listado de lenguajes de programación.

Escenario de fracaso:

1. El campo "slug" coincide con el de otro lenguaje de programación ya existe.
2. Mensaje de advertencia "**Elemento no actualizado.**".

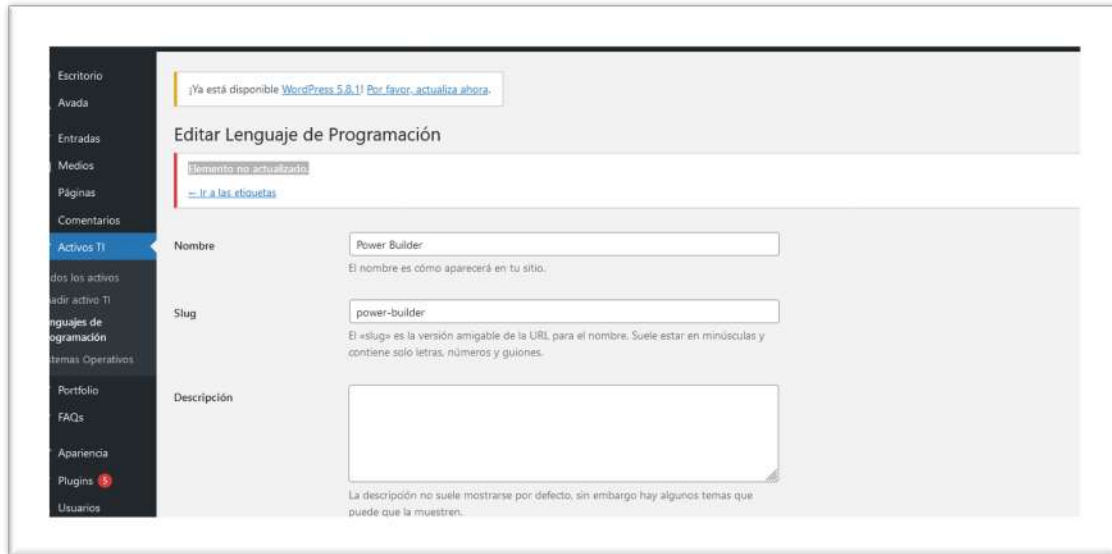


Figura 7.9 Pantalla editar un lenguaje de programación

7.6.3 Borrar de un lenguaje de programación

Objetivo:	Borrar un lenguaje de programación.
Precondiciones	El usuario está registrado con perfil de administrador
Postcondiciones	Se elimina el lenguaje del sistema, que ya no aparecerá en ningún activo
Actor principal	Usuario registrado con permisos administrador

Escenario de éxito principal:

1. El usuario hace clic en **"Borrar"** al situarse sobre la fila del lenguaje de programación.
2. El usuario confirma el borrado haciendo clic en el botón **"Aceptar "**.
3. El lenguaje de programación se elimina del listado.

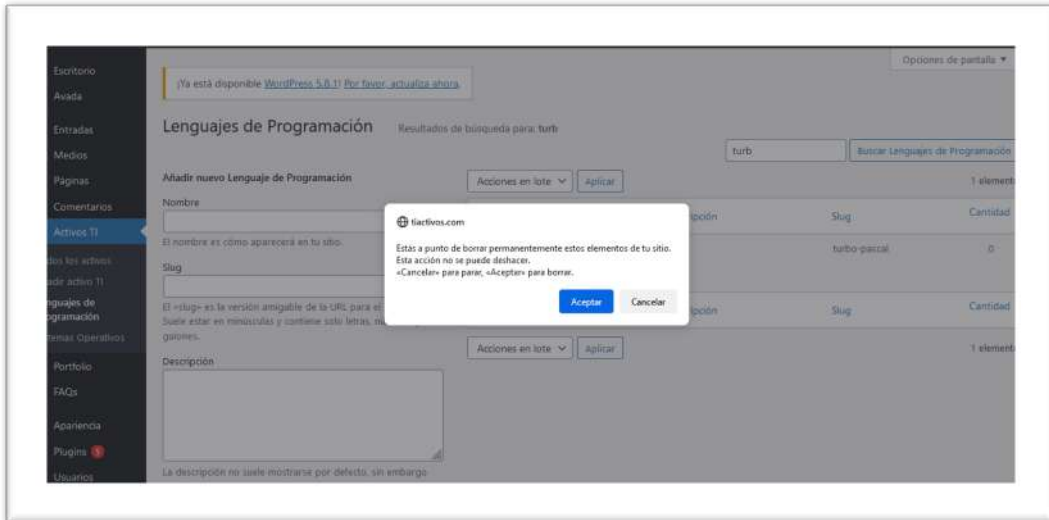


Figura 7.10 Pantalla confirmación de borrado de un lenguaje de programación.

7.7 Sistemas Operativos

7.7.1 Alta de un sistema operativo

Objetivo:	Crear un nuevo sistema operativo.
Precondiciones	El usuario está registrado con perfil de administrador
Postcondiciones	Se añade el nuevo sistema operativo
Actor principal	Usuario registrado con permisos administrador

Escenario de éxito principal:

1. El usuario introduce un sistema operativo que no existe aún.
2. El usuario hace clic en **"Añadir nuevo sistema operativo"**.
3. El sistema operativo se añade al listado.

Escenario de fracaso:

1. El sistema operativo ya existe.
2. Mensaje de advertencia "Ya existe en esta taxonomía un término con el nombre y el slug facilitados."

7.7.2 Edición de un sistema operativo

Objetivo:	Editar un sistema operativo.
Precondiciones	El usuario está registrado con perfil de administrador
Postcondiciones	Se actualizan los cambios al sistema operativo
Actor principal	Usuario registrado con permisos administrador

Escenario de éxito principal:

1. El usuario hace clic en "**Editar**" al situarse sobre la fila del sistema operativo.
2. El usuario edita los campos del sistema operativo.
3. El usuario hace clic en el botón "**Actualizar**".
4. Los cambios se reflejan en el listado de sistemas operativos.

Escenario de fracaso:

1. El campo "slug" coincide con el de otro sistema operativo ya existe.
2. Mensaje de advertencia "**Elemento no actualizado.**".

7.7.3 Borrar de un sistema operativo

Objetivo:	Borrar un sistema operativo.
Precondiciones	El usuario está registrado con perfil de administrador
Postcondiciones	
Actor principal	Usuario registrado con permisos administrador

Escenario de éxito principal:

1. El usuario hace clic en "**Borrar**" al situarse sobre la fila del sistema operativo.
2. El usuario confirma el borrado haciendo clic en el botón "**Aceptar**".
3. El sistema operativo se elimina del listado.

7.8 Visualizar páginas del sitio Web

7.8.1 Visualizar la página principal

Objetivo:	Visualizar una página principal del sitio Web
Precondiciones	Se solicita al navegador la URL del sitio Web <code>www.tiactivos.com</code>
Postcondiciones	Se muestra la página principal.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario puede visualizar la página principal ajustada a la resolución de pantalla del dispositivo utilizado.



Figura 7.11 Pantalla inicio vista en un equipo de sobremesa

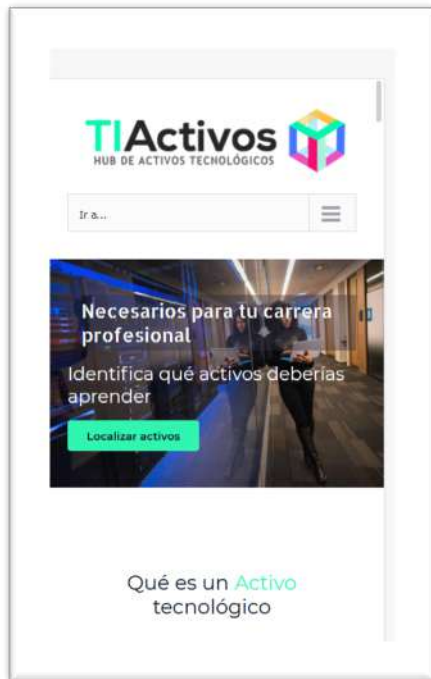


Figura 7.12 Pantalla inicio vista en un teléfono móvil 375x667px

7.8.2 Visualizar entrada del blog

Objetivo:	Visualizar una entrada del blog
Precondiciones	Se solicita al navegador la URL de una entrada del blog.
Postcondiciones	Se muestra la entrada en pantalla.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario puede visualizar una entrada del blog del sitio Web ajustada a la resolución de pantalla del dispositivo utilizado.

La estructura de información de un Activo

Inicio / Tipos de Proyectos / La estructura de información de un Activo



La estructura de información de un Activo

La estructura de información de un Activo en **TIActivos** está compuesta de 6 bloques.

- **Cabecera**, en la que se incluye una descripción breve y si existen versiones de software libre y/o de pago.
- **Descripción**, se muestra información general del activo, las funcionalidades que ofrece, etc.
- **Caracterización**. Uno de los objetivos de **TIActivos** es ofrecer una clasificación de los diferentes activos. Aquí se muestra la categorización de un activo atendiendo a diferentes parámetros: Qué es, su tipo, en qué tipo de proyectos principalmente se podría usar, para qué actividades, cuáles serían los perfiles profesionales que lo utilizan, etc. Finalmente, y si tiene sentido, para qué lenguajes de programación y/o sistemas operativos está preparado el activo.
- **Comparable con**. Es una información que se considera muy valiosa, y se incluye en varios activos. Con el gran número de soluciones y herramientas disponibles actualmente, es normal que existan diferentes alternativas que puedan usarse de forma equivalente con la misma finalidad. En este apartado se muestra una comparación de los elementos más relevantes dependiendo del tipo de activo, con otros activos equivalentes.
- **Estándares**. La finalidad de este apartado responde a otro de los objetivos importantes de **TIActivos**, que consiste en mostrar especialmente para profesionales y organizaciones TI, en qué proceso de un estándar tiene más sentido usar el activo correspondiente. Es decir, para las organizaciones y departamentos TI hoy en día es necesario seguir estándares dentro de los servicios TI que están bajo su responsabilidad (desarrollo de software, gestión de servicios, seguridad, etc.). Para tener optimizados y bien desplegados sus procesos, es imprescindible utilizar soluciones y herramientas que ayuden a trabajar con control, efectividad y calidad.
- **Aprendizaje**. En esta sección se mostrarán referencias en Internet que ayuden a conocer de manera más amplia sobre el activo en cuestión (referencias de descripción), o por otra parte, a formarse y aprender sobre el mismo (referencias de formación).

Compartir



Q Buscar...

Popular

Reciente



Ejemplo de descripción y caracterización: XAMPP



La estructura de información de un Activo



Los activos y los estándares de proceso

Entradas recientes

> Activos para perfiles profesionales

> La estructura de información de un Activo

> Los activos y los estándares de proceso

> Sin herramientas no podemos hacer DevOps

> CMMI efectivo mediante activos

Categorías

> Tipos de Proyectos

Figura 7.13 Pantalla visualización de una entrada del blog

7.8.3 Enlaces banda lateral derecha del blog

Objetivo:	Probar el funcionamiento de los enlaces situados en la banda derecha del blog del sitio Web.
Precondiciones	Se ha cargado en el navegador la página principal del blog o una entrada del mismo.
Postcondiciones	Se muestra la entrada o página enlazada.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario hace clic en el enlace y se muestra la entrada o página Web.



Figura 7.14 Pantalla principal del blog

7.9 Realizar búsquedas de activos

7.9.1 Buscar todos los activos

Objetivo:	Buscar todos los activos publicados en el sitio Web.
Precondiciones	No se ha establecido ninguna condición de filtro para los activos.
Postcondiciones	Se muestra el total de activos paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario hace clic en el enlace "**BUSCAR ACTIVOS**" y se muestra el listado con todos los activos.

7.9.2 Buscar activos filtrados por el campo qué es

Objetivo:	Buscar todos los activos publicados que en el campo "Qué es" tienen el valor lenguaje de programación o framework desarrollo.
Precondiciones	En el formulario de búsqueda de han chequeado esas condiciones
Postcondiciones	Se mantienen las condiciones chequeadas en el formulario de búsqueda y se muestran los activos que las cumplen paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario chequea los valores en el campo del formulario de condiciones de búsqueda.
2. El usuario hace clic en el enlace "**BUSCAR**"
3. Se muestra el listado de activos que cumplen al menos una de las condiciones.

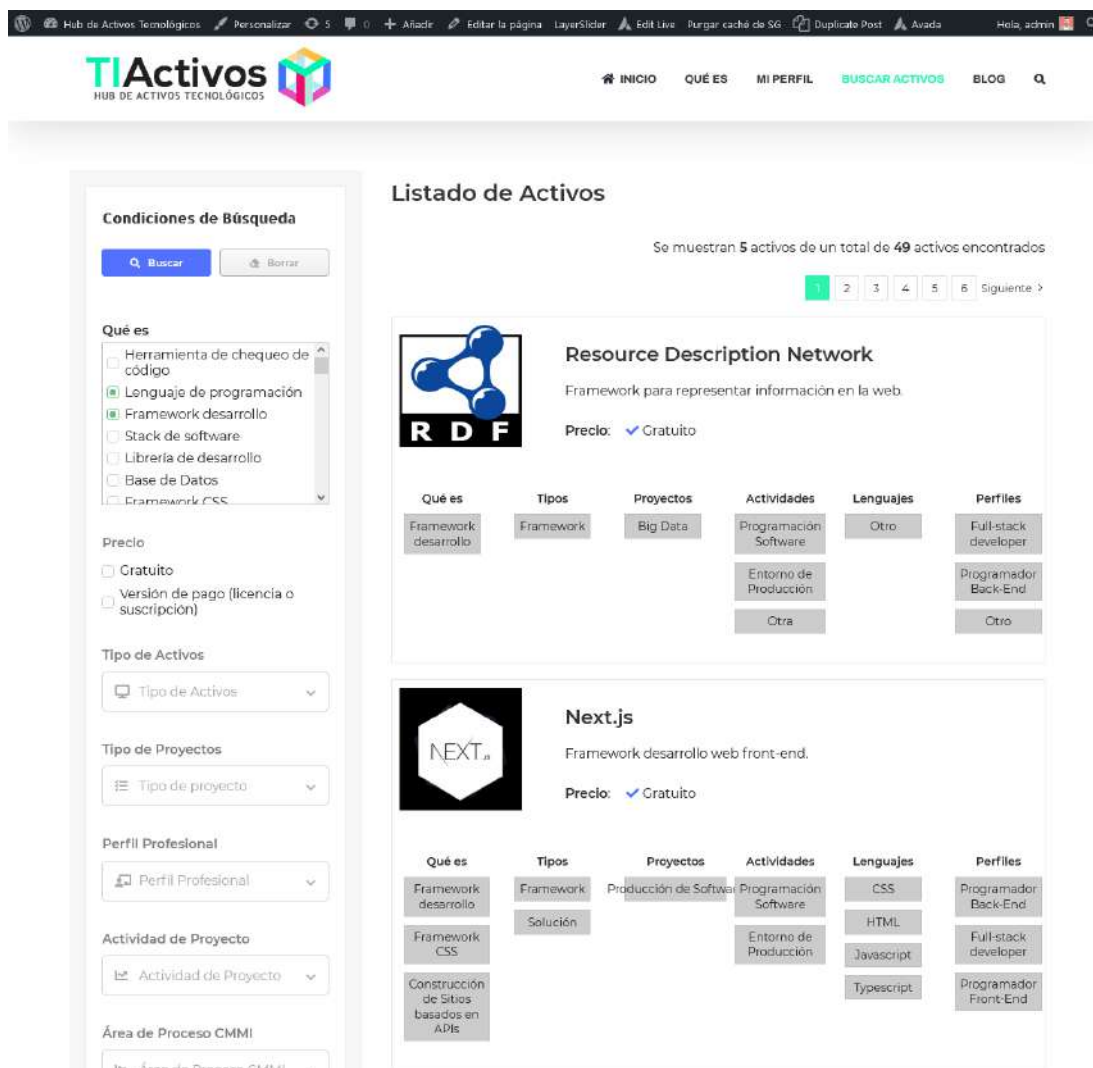


Figura 7.15 Pantalla de resultado de búsqueda con condiciones

7.9.3 Buscar activos filtrados por los campos qué es y precio

Objetivo:	Buscar todos los activos publicados que en el campo Qué es tienen el valor lenguaje de programación o framework desarrollo y que el precio es versión de pago.
Precondiciones	En el formulario de búsqueda de han chequeado esas condiciones.
Postcondiciones	Se mantienen las condiciones chequeadas en el formulario de búsqueda y se muestran los activos

	que las cumplen paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario checea los valores en el campo del formulario de condiciones de búsqueda.
2. El usuario hace clic en el enlace **"BUSCAR"**
3. Se muestra el listado de activos que cumplen al menos una de las condiciones para el campo qué es y además el precio incluye el valor versión de pago.

7.9.4 Buscar activos filtrados por el campo tipo de activo

Objetivo:	Buscar todos los activos publicados que tengan como tipo de activo el valor librería.
Precondiciones	En el formulario de búsqueda de ha seleccionado ese valor para el campo.
Postcondiciones	Se mantienen las condiciones seleccionadas en el formulario de búsqueda y se muestran los activos que las cumplen paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario selecciona el valor en el campo "Tipo de Activos" del formulario de condiciones de búsqueda.
2. El usuario hace clic en el enlace **"BUSCAR"**
3. Se muestra el listado de activos que cumplen esa condición.

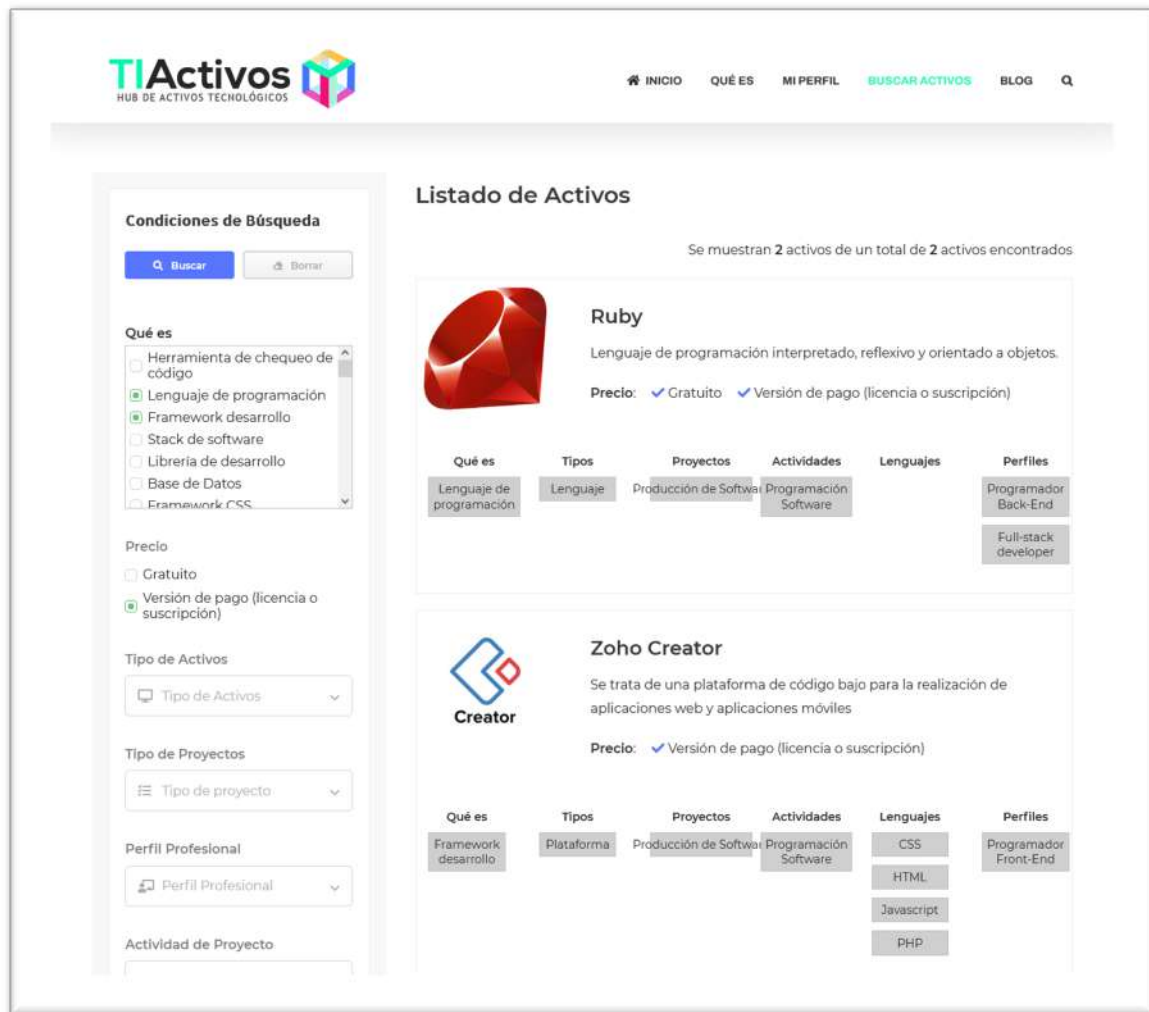


Figura 7.16 Pantalla de resultado de búsqueda con valor que es igual a lenguaje de programación o framework y precio versión de pago.

7.9.5 Buscar activos filtrados por el campo tipo de proyecto

Objetivo:	Buscar todos los activos publicados que tengan como tipo de proyecto el valor "Gestión del servicio TI".
Precondiciones	En el formulario de búsqueda de ha seleccionado ese valor para el campo.
Postcondiciones	Se mantienen las condiciones seleccionadas en el formulario de búsqueda y se muestran los activos que las cumplen paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario selecciona el valor en el campo "Tipo de Proyectos" del formulario de condiciones de búsqueda.
2. El usuario hace clic en el enlace "**BUSCAR**"
3. Se muestra el listado de activos que cumplen esa condición.

7.9.6 Buscar activos filtrados por el campo perfil profesional

Objetivo:	Buscar todos los activos publicados que tengan como perfil profesional el valor "Arquitecto".
Precondiciones	En el formulario de búsqueda de ha seleccionado ese valor para el campo.
Postcondiciones	Se mantienen las condiciones seleccionadas en el formulario de búsqueda y se muestran los activos que las cumplen paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario selecciona el valor en el campo "Perfil Profesional" del formulario de condiciones de búsqueda.
2. El usuario hace clic en el enlace "**BUSCAR**"
3. Se muestra el listado de activos que cumplen esa condición.

7.9.7 Buscar activos filtrados por el campo actividad de proyecto

Objetivo:	Buscar todos los activos publicados que tengan como actividad de proyecto el valor "DevOps".
Precondiciones	En el formulario de búsqueda de ha seleccionado ese valor para el campo.
Postcondiciones	Se mantienen las condiciones seleccionadas en el formulario de

	búsqueda y se muestran los activos que las cumplen paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario selecciona el valor en el campo "Actividad de Proyecto" del formulario de condiciones de búsqueda.
2. El usuario hace clic en el enlace "**BUSCAR**"
3. Se muestra el listado de activos que cumplen esa condición.

7.9.8 Buscar activos filtrados por el campo área de proceso CMMI

Objetivo:	Buscar todos los activos publicados que tengan como área de proceso CMMI el valor "3-VAL-Validación".
Precondiciones	En el formulario de búsqueda de ha seleccionado ese valor para el campo.
Postcondiciones	Se mantienen las condiciones seleccionadas en el formulario de búsqueda y se muestran los activos que las cumplen paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario selecciona el valor en el campo "Área de proceso CMMI" del formulario de condiciones de búsqueda.
2. El usuario hace clic en el enlace "**BUSCAR**"
3. Se muestra el listado de activos que cumplen esa condición.

7.9.9 Buscar activos filtrados por el campo objetivos específicos CMMI

Objetivo:	Buscar todos los activos publicados que tengan como objetivos específicos
-----------	---

	CMMI el valor "3-TS: Describir y concretar el Diseño".
Precondiciones	En el formulario de búsqueda de ha seleccionado ese valor para el campo.
Postcondiciones	Se mantienen las condiciones seleccionadas en el formulario de búsqueda y se muestran los activos que las cumplen paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario selecciona el valor en el campo "Objetivos Específicos CMMI" del formulario de condiciones de búsqueda.
2. El usuario hace clic en el enlace **"BUSCAR"**
3. Se muestra el listado de activos que cumplen esa condición.

7.9.10 Buscar activos filtrados por el campo proceso PMBOK

Objetivo:	Buscar todos los activos publicados que tengan como proceso PMBOK el valor "Crear y manejar Planificaciones".
Precondiciones	En el formulario de búsqueda de ha seleccionado ese valor para el campo.
Postcondiciones	Se mantienen las condiciones seleccionadas en el formulario de búsqueda y se muestran los activos que las cumplen paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario selecciona el valor en el campo "Proceso PMBOK" del formulario de condiciones de búsqueda.
2. El usuario hace clic en el enlace **"BUSCAR"**

3. Se muestra el listado de activos que cumplen esa condición.

7.9.11 Buscar activos filtrados por el campo gestión del servicio TI

Objetivo:	Buscar todos los activos publicados que tengan como gestión del servicio TI el valor "Operación del Servicio. Gestión de Problemas".
Precondiciones	En el formulario de búsqueda de ha seleccionado ese valor para el campo.
Postcondiciones	Se mantienen las condiciones seleccionadas en el formulario de búsqueda y se muestran los activos que las cumplen paginados de 5 en 5 resultados.
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario selecciona el valor en el campo "Gestión del Servicio TI" del formulario de condiciones de búsqueda.
2. El usuario hace clic en el enlace "**BUSCAR**".
3. Se muestra el listado de activos que cumplen esa condición.

7.10 Ver el contenido de un activo

7.10.1 Ver la ventana principal del activo

Objetivo:	Ver la ficha de un activo determinado mostrando la pestaña principal con la descripción del activo.
Precondiciones	Se ha pedido al servidor la URL de un activo.
Postcondiciones	Se muestran enlaces al resto de pestañas que agrupan la información del activo

Actor principal	Cualquier usuario
-----------------	-------------------

Escenario de éxito principal:

1. El usuario hace clic en el enlace al activo
2. Se muestra la ficha del activo con la descripción.

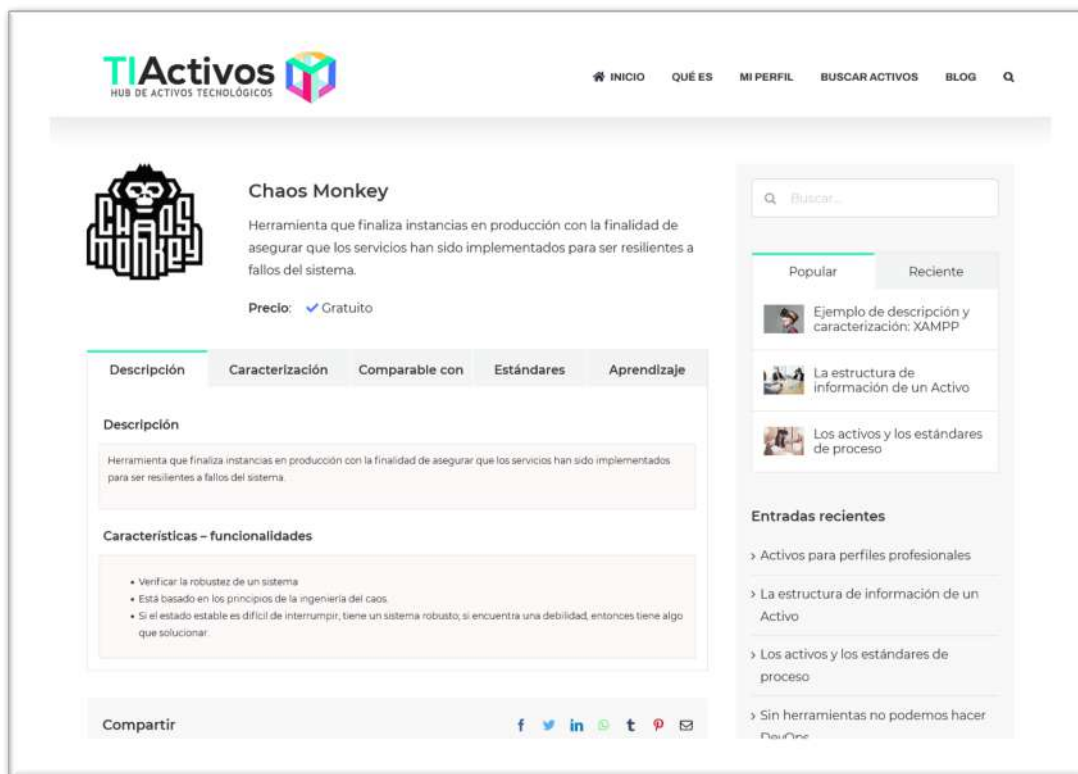


Figura 7.17 Pantalla ficha de un activo pestaña descripción

7.10.2 Ver la pestaña caracterización de un activo

Objetivo:	Ver la pestaña caracterización de la ficha de un activo.
Precondiciones	Se muestra la ficha del activo.
Postcondiciones	Se muestran enlaces al resto de pestañas que agrupan la información del activo
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario hace clic en la pestaña "**Caracterización**" del activo
2. Se muestra información de caracterización del activo.

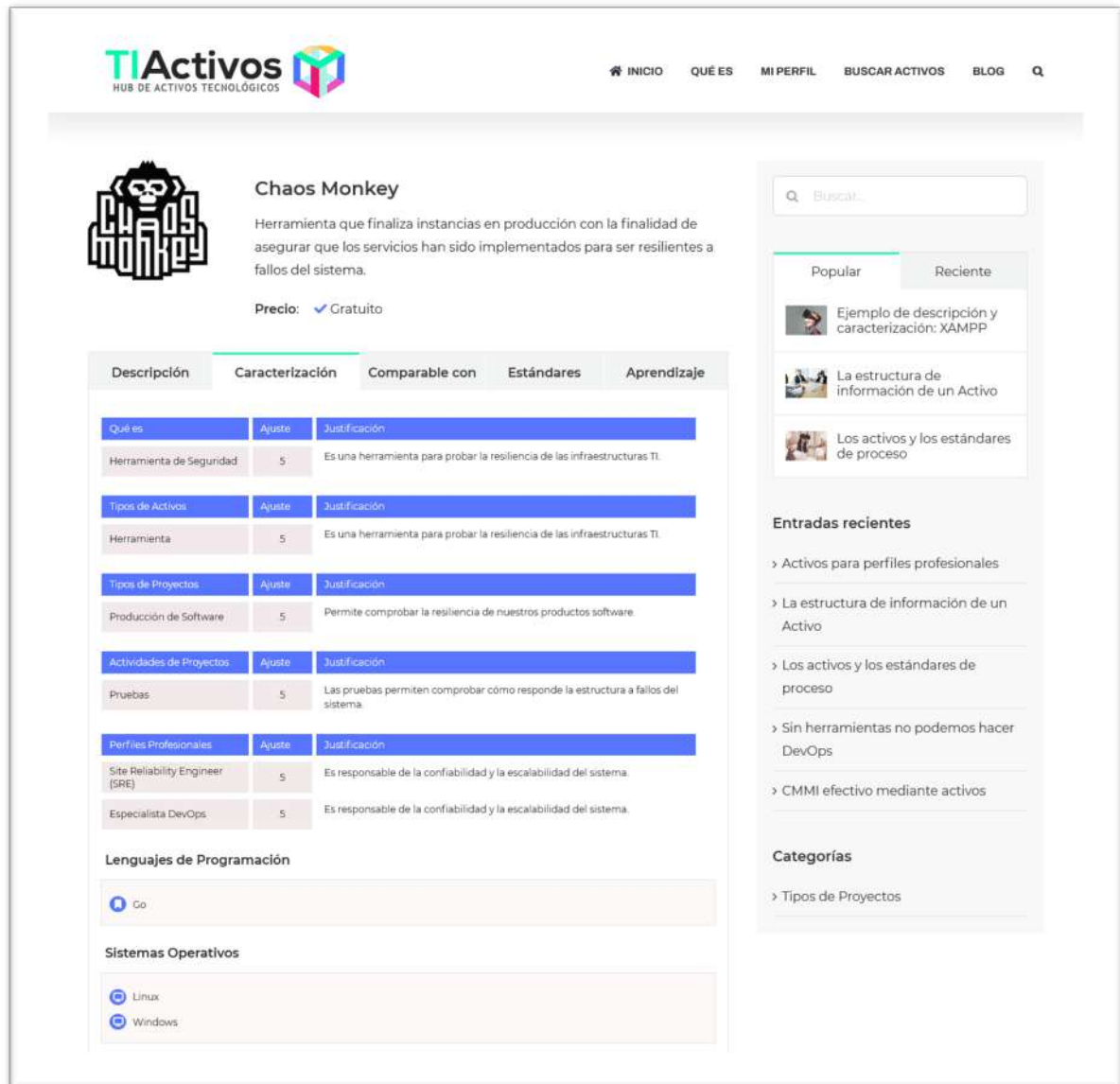


Figura 7.17 Pantalla ficha de un activo pestaña caracterización

7.10.3 Ver la pestaña comparable con de un activo

Objetivo:	Ver la pestaña "Comparable con" de la ficha de un activo.
Precondiciones	Se muestra la ficha del activo.
Postcondiciones	Se muestran enlaces al resto de pestañas que agrupan la información del activo
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario hace clic en la pestaña **"Comparable con"** del activo
2. Se muestra información de caracterización del activo.

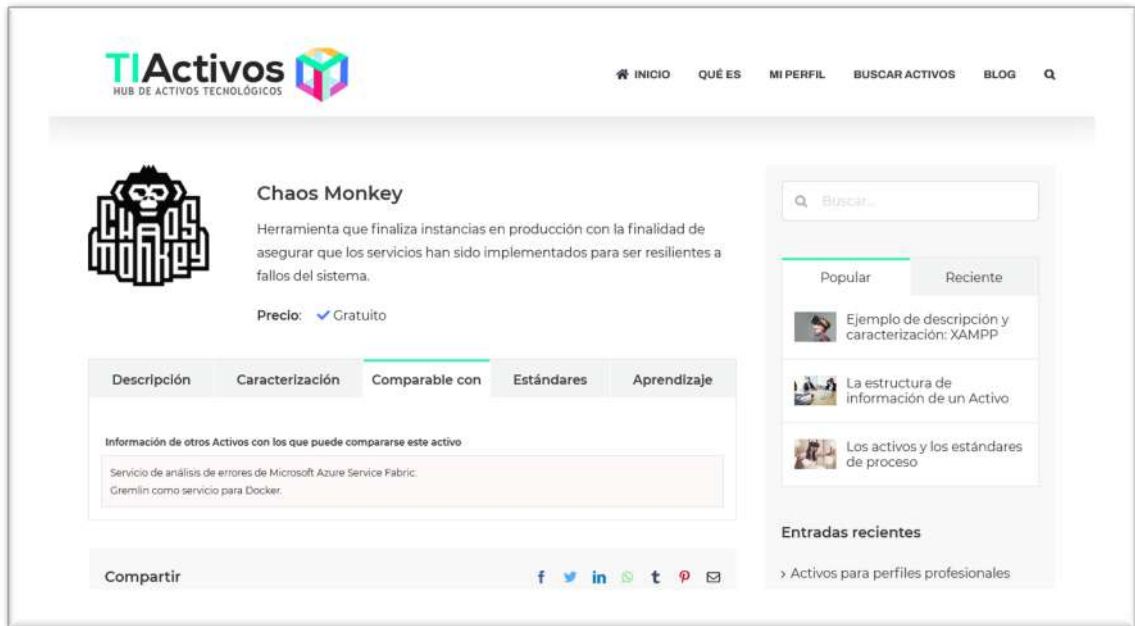


Figura 7.18 Pantalla ficha de un activo pestaña comparable con

7.10.4 Ver la pestaña estándares de un activo

Objetivo:	Ver la pestaña "Estándares" de la ficha de un activo.
Precondiciones	Se muestra la ficha del activo.
Postcondiciones	Se muestran enlaces al resto de pestañas que agrupan la información del activo
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario hace clic en la pestaña **"Estándares"** del activo
2. Se muestra información de caracterización del activo.

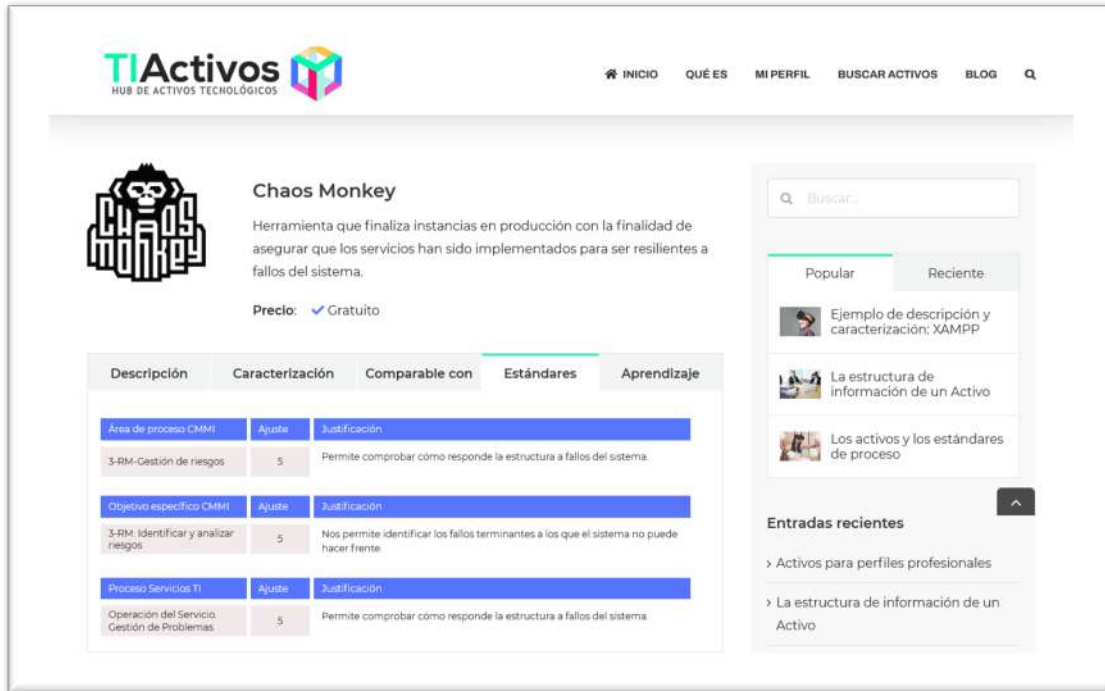


Figura 7.19 Pantalla ficha de un activo pestaña estándares

7.10.5 Ver la pestaña aprendizaje de un activo

Objetivo:	Ver la pestaña "Aprendizaje" de la ficha de un activo.
Precondiciones	Se muestra la ficha del activo.
Postcondiciones	Se muestran enlaces al resto de pestañas que agrupan la información del activo
Actor principal	Cualquier usuario

Escenario de éxito principal:

1. El usuario hace clic en la pestaña "**Aprendizaje**" del activo
2. Se muestra información de caracterización del activo.



Figura 7.19 Pantalla ficha de un activo pestaña aprendizaje

7.11 Pruebas realizadas

A continuación se recogen una tabla con todas las pruebas realizadas que forman parte del plan de pruebas.

Caso de uso	Fecha	Respon- -sable	Entorno	Resultados obtenidos	Pasa
Crear un nuevo activo	08/02/21	JCRD ²³	PC	Se crea el activo XAMPP	Sí
Modificar un activo creado por el usuario	08/02/21	JCRD	PC	Se editan los campos del activo	Sí
Crear un nuevo activo	08/02/21	JCRD	PC	Se crea el activo PRUEBA3	Sí
Borrar un activo creado por el usuario	08/02/21	JCRD	PC	El activo PRUEBA3 queda eliminado	Sí
Crear un lenguaje de programación	12/02/21	JCRD	PC	Se crea el lenguaje PHP	Sí
Modificar un lenguaje de programación	12/02/21	JCRD	PC	Se edita el lenguaje PHP	Sí

²³ José Carlos Ruiz Díaz

Asignar el nuevo lenguaje al activo XAMPP	12/02/21	JCRD	PC	Se asigna el lenguaje PHP	Sí
Borrar el lenguaje de programación PHP	12/02/21	JCRD	PC	Se borra el lenguaje PHP	Sí
Crear un nuevo sistema operativo	12/02/21	JCRD	PC	Se crea el sistema operativo Android	Sí
Editar el sistema operativo Android	12/02/21	JCRD	PC	Se modifica el sistema operativo Android	Sí
Crear un nuevo sistema operativo	12/02/21	JCRD	PC	Se crea el sistema operativo MS-DOS	Sí
Borrar el sistema operativo MS-DOS	12/02/21	JCRD	PC	Se elimina el sistema operativo MS-DOS	Sí
Se asigna el sistema operativo Windows al activo XAMPP	12/02/21	JCRD	PC	Se asigna al activo XAMPP el sistema operativo Windows	Sí

Crear un usuario EDITORACTIVOS	08/02/21	JCRD	PC	Crear el usuario "Editor de Activos"	Sí
Ver un activo	23-04-21	JCRD	PC	No se muestra la valoración del tipo de activo	No
	06-06-21				Sí
Ver un activo	23-04-21	JCRD	PC	No se muestra la justificación de la valoración del tipo de activo	No
	06-06-21				Sí
Buscar activos	23-04-21	JCRD	PC	Muestra todos los activos, no filtra los resultados por ninguna condición	No
	24-06-21			Filtra por algunos campos	Sí
Ver un activo	06-06-21	JCRD	PC	Muestra todos los campos.	Sí
Ver un activo	06-06-21	JCRD	Móvil	Muestra todos los campos, pero se solapa la visualización	No
	24-06-21			El activo se ve correctamente	Sí

Buscar activos	06-06-21	JCRD	PC	Se visualizan los resultados de búsqueda con formato	Sí
Buscar activos	06-06-21	JCRD	Móvil	Se visualizan los resultados de la búsqueda con formato	Sí
Filtro por campo "Qué es"	06-06-21	JCRD	PC	No muestra ningún activo en los resultados	No
	24-06-21			Se muestran los resultados	Sí
Buscar con condiciones de búsqueda	06-06-21	JCRD	PC	Se quedan marcadas la condiciones en el formulario de búsqueda.	Sí
Acceder a activo desde búsqueda	06-06-21	JCRD	PC	Solo funciona si el título del activo y el slug coinciden	No
	06-06-21			Funciona para todos los activos	Sí

Buscar por campo precio	06-06-21	JCRD	PC	Muestra los resultados que cumplen la condición	Sí
Ver resumen de activo	06-06-21	JCRD	PC	Los resultados de la búsqueda muestran una ficha resumen del activo	Sí
Mostrar número de activos en la búsqueda	10-06-21	JCRD	PC	Se incluye información del número de resultado encontrados.	Sí
Filtro por campo "Qué es"	24-06-21	JCRD	PC	Funciona la búsqueda por este campo	Sí
Buscar con condiciones de búsqueda	24-06-21	JCRD	PC	Muestra los activos que cumplen las condiciones.	Sí
Navegar por resultados de la búsqueda	24-06-21	JCRD	PC	No funciona la paginación.	No
	01-07-21				Sí
Ver un activo	24-06-21	JCRD	Móvil	Muestra todos los campos de forma correcta	Sí

Buscar con condiciones de búsqueda	29-06-21	JCRD	PC	Los resultados se muestran paginados de 5 en 5.	Sí
Buscar con condiciones de búsqueda	01-07-21	JCRD	PC	Las condiciones se mantienen seleccionadas al navegar por la paginación.	Sí
Búsqueda con condiciones en varios campos	08-07-21	JCRD	PC	Las búsquedas muestran correctamente los resultados, paginan y se puede acceder a la ficha de los activos encontrados.	Sí
Búsqueda con condiciones en varios campos	08-07-21	JCRD	Móvil	Las búsquedas muestran correctamente los resultados, paginan y se puede acceder a la ficha de los activos encontrados.	Sí
Resetear las condiciones de búsqueda	08-07-21	JCRD	PC	Las condiciones se borran y los resultados de la búsqueda muestra todos los activos registrados	Sí

Conclusiones

Finalmente, como líneas de trabajo futuro se presentan la ampliación de funcionalidades del sitio Web, para permitir la propuesta por parte de los usuarios de dar de alta nuevos activos en el "hub". La posibilidad de sugerir actualizaciones o modificación de la información existente sobre un activo.

La implementación de alguna funcionalidad que permita "comparar" activos que estén igualmente catalogados, si bien este objetivo puede resultar difícil al tener los activos diferentes criterios en función a sus características específicas. El planteamiento será en cualquier caso tener formatos y vistas preparadas, que luego se puedan usar específicamente para algunas comparaciones, aunque se construyan a medida y no de forma automática.

Entre las líneas posibles de trabajo para ampliar las funcionalidades ofrecidas por el sitio Web, se podrá incorporar la posibilidad de realizar revisiones, comentarios y opiniones sobre los activos. En esta misma línea de interacción con el usuario, también podría resultar interesante la incorporación a la ficha de activos de un sistema de valoración o puntuación por parte de los usuarios visitantes del sitio, permitiendo incorporar a las comparativas entre activos este nuevo campo.

Igualmente se desea proporcionar un formulario que permita a los usuarios participar en la mejora de los contenidos:

- Permitted que sugieran cambios en los activos actuales, en cualquiera de sus campos, por ejemplo proporcionando nuevas referencias descriptivas y de aprendizaje.
- Permittedles sugerir la incorporación de nuevos activos a la Web.

En el plano personal este proyecto me ha permitido adquirir conocimientos acerca de WordPress, uno de los sistemas de gestión de contenidos más extendidos en la actualidad, aportándome una visión práctica de gran valor y que considero de gran utilidad a nivel profesional.

Para la implementación de este sitio Web se han usado tecnologías como PHP y JavaScript, muy presentes en la actualidad y que a nivel profesional son muy demandadas, además de herramientas de gran utilidad como Visual Studio Code y los repositorios Git, que unidos a los complementos para WordPress nos han permitido realizar un sitio muy elegante y práctico en el que poder recoger, almacenar y consultar activos tecnológicos.

Sin duda, el sitio Web cumple con las especificaciones funcionales originales y ha sido utilizado para la carga de activos tecnológicos por parte de los alumnos de la asignatura calidad del software. Esto ha sido posible gracias a que hemos utilizado DevOps, adaptando el desarrollo del proyecto a las necesidades de los usuarios durante todo el proceso de desarrollo.

Por último, quiero agradecer y resaltar la labor realizada por mi tutor, que siempre me ha alentado a superar las dificultades que he encontrado durante la realización de este trabajo fin de grado, encaminando de la mejor manera posible mis investigaciones y esfuerzos para encontrar la solución más adecuada y compartiendo conmigo su experiencia y conocimientos.

Bibliografía

- [1] «w3techs,» 11 07 2021. [En línea]. Available: https://w3techs.com/technologies/overview/content_management.

- [2] F. G. d. Zúñiga, «www.arsys.es,» 07 julio 2019. [En línea]. Available: <https://www.arsys.es/blog/programacion/introduccion-a-laragon/>. [Último acceso: julio 2021].

- [3] jmaleman, «Configurar Visual Studio Code para el desarrollo con PHP,» 19 02 2019. [En línea]. Available: <https://www.jmaleman.dev/blog/2019/02/19/configurar-visual-studio-code-para-el-desarrollo-de-php/>.

- [4] I. Ruiz, «Introducción a Git y GitHub,» 26 03 2021. [En línea]. Available: <https://carontestudio.com/blog/introduccion-a-git-y-github/>.

- [5] «Introducción GitLab,» 12 10 2017. [En línea]. Available: <https://desarrolloweb.com/articulos/introduccion-gitlab.html>.

- [6] J. R. Cristina, «Que es DevOps y sobre todo que no es DevOs,» 26 11 2015. [En línea]. Available: <https://www.paradigmadigital.com/techbiz/que-es-devops-y-sobre-todo-que-no-es-devops/>.

- [7] S. Chacon, «Ramificaciones en Git - ¿Qué es una rama?,» 10 03 2015. [En línea]. Available: <https://git-scm.com/book/es/v2/Ramificaciones-en-Git-%C2%BFQu%C3%A9-es-una-rama%3F>.

- [8] Microsoft, «Administradores de Microsoft Teams para el ámbito educativo,» 2021. [En línea]. Available: <https://docs.microsoft.com/es-es/microsoftteams/teams-quick-start-edu>.
- [9] E. Marcia, «DEV Community,» 10 Julio 2020. [En línea]. Available: <https://dev.to/erickmarcia/php-xdebug-laragon-visual-studio-code-4ocd>.
- [10] L. Arnedo, J. Gama y G. Prat Masramon, «ProyectosAgiles.org,» 2015 Octubre 25. [En línea]. Available: <https://proyectosagiles.org/que-es-scrum/>. [Último acceso: 31 07 2021].
- [11] K. Sutherland y J. Schwaber, La Guía Scrum, 2020.
- [12] D. Martínez, «DevOps una metodología que trabaja en el desarrollo de software ágiles,» 9 Mayo 2019. [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/devops-metodologia-desarrollo-software.html>. [Último acceso: 31 Julio 2021].
- [13] J. Lirón, «Avada Theme: El tema WordPress más vendido del mundo,» 21 Febrero 2015. [En línea]. Available: <https://www.creartiadvirtual.com.es/avada-theme/>. [Último acceso: 18 08 2021].
- [14] V. Garagulia, «User Role Editor,» 2014 Junio 30. [En línea]. Available: <https://www.role-editor.com/>. [Último acceso: 2021 Febrero 26].
- [15] M. Beckwith, J. Hawkins y B. Williams, «Custom Post Type UI,» 14 Noviembre 2015. [En línea]. Available: <https://es.wordpress.org/plugins/custom-post-type-ui/>. [Último acceso: 12 febrero 2021].
- [16] B. Messenlehner y J. Coleman, Building Web Apps with WordPress, Sebastopol: O'Reilly Media, 2014, pp. 12-14.

- [17] WordPress, «Plugin HandBook,» 30 Octubre 2014. [En línea]. Available: <https://developer.wordpress.org/plugins/plugin-basics/best-practices/>. [Último acceso: 2021 Febrero 26].
- [18] I. Dunn, «Implementing the MVC Pattern in WordPress Plugins,» 21 Enero 2013. [En línea]. Available: <https://iandunn.name/content/presentations/wp-oop-mvc/mvc.php#/>. [Último acceso: 2021 Febrero 26].
- [19] Varios, «Entendiendo el Modelo - Vista - Controlador,» 2020 Noviembre 30. [En línea]. Available: <https://book.cakephp.org/2/es/cakephp-overview/understanding-model-view-controller.html>. [Último acceso: 2021 Febrero 26].
- [20] D. Vinson, «WordPress Plugin Boilerplate,» 19 Febrero 2019. [En línea]. Available: <https://github.com/DevinVinson/WordPress-Plugin-Boilerplate>. [Último acceso: 26 Febrero 2021].
- [21] WordPress, «Theme Handbook,» 2015 Junio 5. [En línea]. Available: <https://developer.wordpress.org/themes/basics/theme-functions/>. [Último acceso: 2021 Febrero 26].
- [22] WordPress, «Theme Handbook,» 2015, 14 Enero 2015. [En línea]. Available: <https://developer.wordpress.org/themes/basics/template-files/>. [Último acceso: 2021 Febrero 26].
- [23] J. Casares, «Make WordPress Hosting,» 24 Junio 2021. [En línea]. Available: <https://make.wordpress.org/hosting/handbook/server-environment/>. [Último acceso: 14 Agosto 2021].
- [24] C. Lamle y B. Riley, «Duplicator - WordPress Migration Plugin,» Snap Creek, 7 Febrero 2021. [En línea]. Available: <https://es.wordpress.org/plugins/duplicator/>. [Último acceso: 12 Febrero 2021].

Apéndice A

Manual de Instalación

Requerimientos:

Para la correcta instalación del software será necesario disponer de un servidor Web, preferiblemente Apache o nginx, compatible con PHP versión 7.4 o superior, un sistema de base de dato MySQL 5.6 o superior o MariaDB versión 10.1 o superior. Compatible con HTTPS. [23]

Para poder facilitar el despliegue de la instalación se ha utilizado el complemento para WordPress "**Duplicator PRO**" que facilita las migraciones y las copias de seguridad de sitios Web. Por lo tanto permite clonar correctamente un sitio de una ubicación a otra. [24]

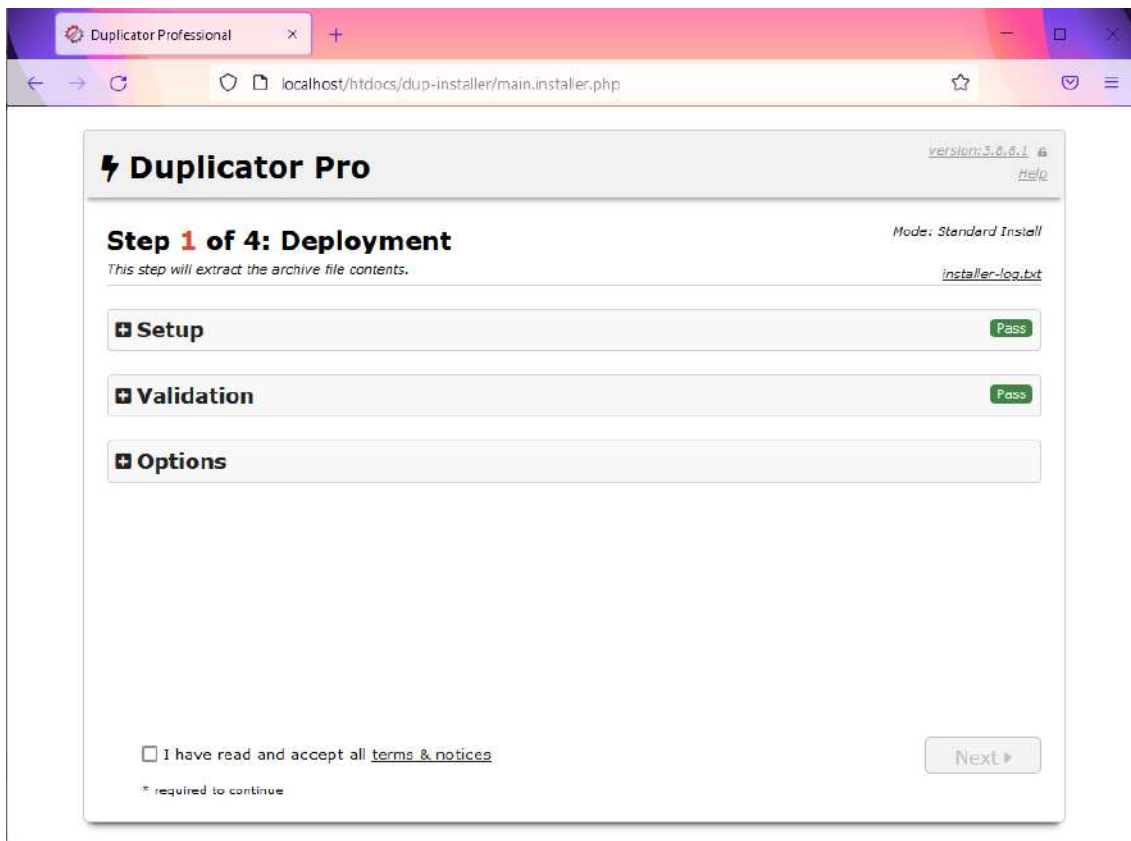
Por medio de este complemento se ha generado un paquete de instalación que contiene dos ficheros y que automatiza y facilita el proceso.

Instalación en servidor Web

Una vez descargados los ficheros de instalación "installer.php" y "tiactivos_6739d0a29e8c0e6d6816_20210913223224_archive.daf", es necesario subirlos por FTP al directorio raíz de publicación en el servidor.

Paso 1: Despliegue

Después debemos abrir un navegador y solicitar la dirección del fichero de Instalación. En nuestro ejemplo es: <http://localhost/htdocs/installer.php>



Tras leer y aceptar los términos y condiciones hacemos clic en el botón "Next" y comienza la descompresión del fichero que contiene la información para la instalación.



Paso 2: Instalar la base de datos

Debemos indicar el nombre del huésped (host) donde se encuentra el servidor de bases de datos, el nombre de la base de datos donde se va a almacenar la información, debemos indicar el usuario de base de datos y su contraseña de acceso.

En la parte inferior tenemos una botón para comprobar la conexión con la base de datos.

Duplicator Pro version:3.6.6.1 [Help](#)

Step 2 of 4: Install Database Mode: Standard Install

This step will install the database from the archive. [installer-log.txt](#)

Basic cPanel

Setup

Action: Create New Database

Host: localhost

Database: DBnueva

User: root

Password: valid database user password

Validation

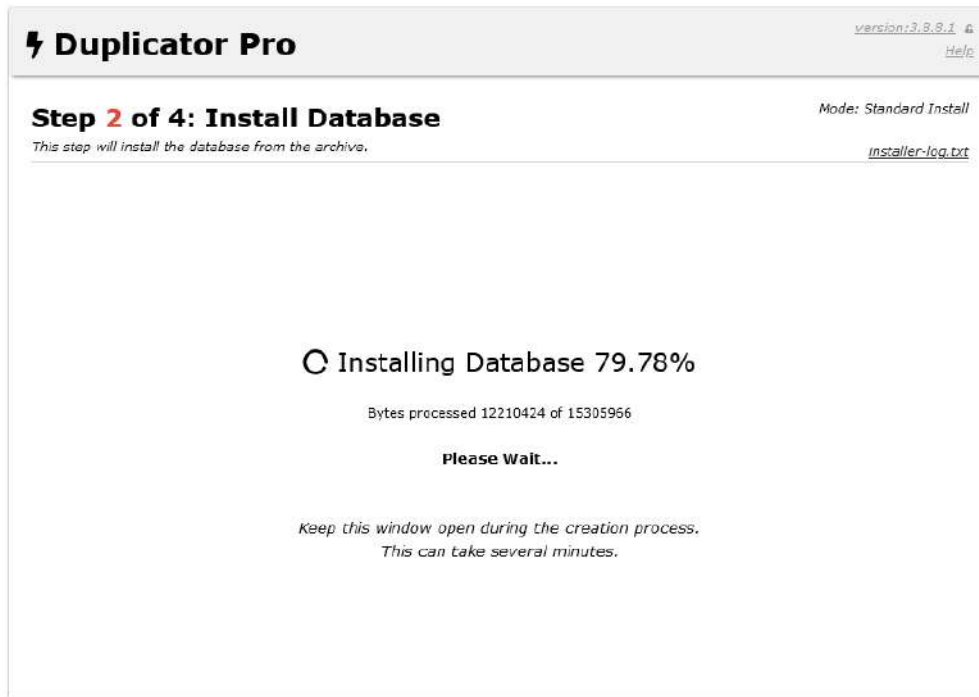
[Retry Test](#)

To continue click the 'Test Database' button to retest the database setup.

Options

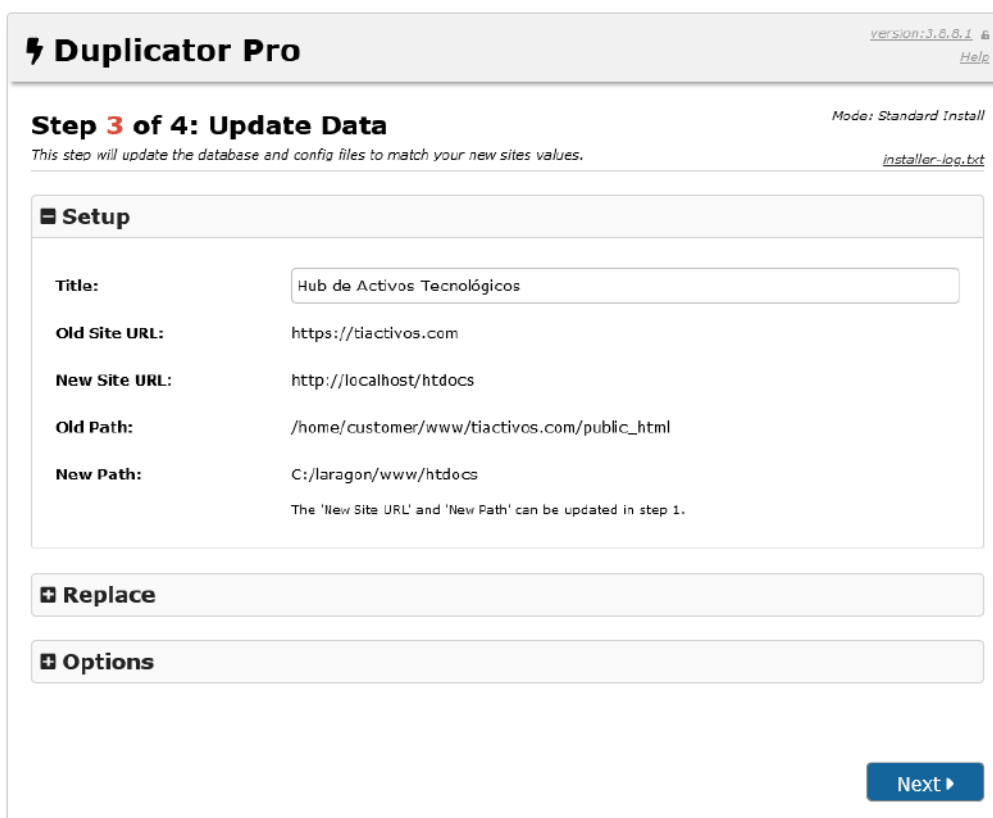
Test Database Next

Tras realizar satisfactoriamente una prueba de acceso a la base de datos podemos continuar con nuestra instalación, creándose las tablas necesarias en la base de datos.



Paso 3: Actulizar los datos

Actualizamos la base de datos y los valores de configuración para que encajen con los nuevos valores del sitio Web.

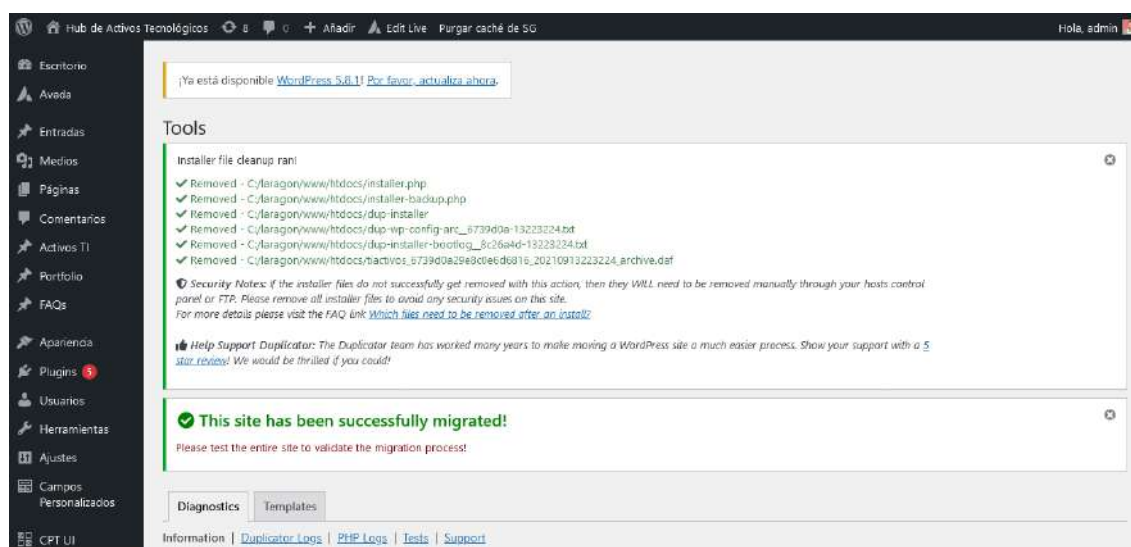


Es importante mantener la ventana abierta mientras se ejecuta el proceso de instalación.

Paso 4: Acceder al panel de administrador

Una vez finalizada la instalación, nos pedirá que nos identifiquemos con nuestro usuario y contraseña para poder acceder al panel de administración del sitio Web.

El usuario **"administrador"** tendrá como contraseña **"tiactivos"**



Una vez finalizada la instalación se produce el borrado de todos los ficheros utilizados durante la misma, quedando el espacio Web preparado para poder gestionar activos de las tecnologías de la información.



UNIVERSIDAD
DE MÁLAGA

| uma.es

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga