

Memoria Proyecto

HeikoShop

Autor: Alberto Cantero Escribano (alberto.cantero@iesjulianmarias.es)

Tutora de proyecto: Marta Alonso Berná

El Proyecto se entregó: el día 24 del mes 5 del año 2023

Índice

Memoria Proyecto HeikoShop	1
1. Presentación	2
1.1. Agradecimientos, dedicatorias	2
1.2. Lista de imágenes/figuras	2
2. Estudio del problema y análisis del sistema	2
2.1. Introducción.	2
2.2. Objetivos (funcionales y técnicos)	4
3. Recursos necesarios para el desarrollo	5
3.1. Recursos hardware	5
3.2. Recursos software	5
4. Desarrollo del proyecto	5
4.1. Capítulos del desarrollo.	5
4.2. Diagrama relacional y diagramas de clases	22
4.3. Interacción con el usuario:	25
4.3.1. Casos de uso.	25
4.3.2. Diseño de la interfaz	33
4.4. Código fuente.	33
5. Fase de pruebas	36
5.1. Pruebas realizadas	36
6. Conclusiones finales	37
7. Documentación del sistema desarrollado	37
7.1. Manual de instalación	37
8. Bibliografía	44
Anexo I: Recursos de la presentación y defensa del proyecto:	47

1. Presentación

Buenos días, mi nombre es Alberto Cantero Escribano y esta es la memoria de mi proyecto. Dicho proyecto ha nacido como la creación de una tienda online para la venta de productos aplicable en un futuro, cuya marca será AlmaHeiko.

1.1. Agradecimientos, dedicatorias

A mi familia y concretamente a mi hermana que fue la principal aportadora de material para el proyecto

1.2. Lista de imágenes/figuras

Estas imágenes serán enviadas con el proyecto general por separado y estarán disponibles en el interior del repositorio Github, agrupadas en sus correspondientes ficheros. Los nombres de las mismas se diferenciarán dependiendo del uso correspondiente que les corresponda.

Las figuras usadas en esta memoria son:

1.

2. Estudio del problema y análisis del sistema

2.1. Introducción.

Este proyecto fue realizado con la idea de otorgar a el cliente (@AlmaHeiko) una tienda de la cual tuviera todo el control posible. Esta necesidad surgió debido a que estuvo vendiendo sus productos durante unos años en ferias del cómic y manga alrededor de España, y muchos compradores solicitaron que se pudieran realizar tales compras desde internet. Por ende, con el objetivo de suplir dicha demanda se creó una tienda en la plataforma Etsy, como se ve en la figura 1. Desgraciadamente, esta plataforma no otorgaba opciones suficientes para todo lo que se buscaba ofrecer en sus productos y muchos otros problemas de la web en cuestión, estos son algunos casos:

- Ésta requiere cierto porcentaje de los ingresos que tu tienda genere, por lo cual los precios de los productos suelen inflarse para suplir este problema.
- Para que tus productos estén bien posicionados en la plataforma es necesario realizar ciertos procesos incoherentes. Como indicar que el envío sea gratuito, haciendo que los precios aumenten aún más para suplir esto.
- Posibilidad de que los ingresos sean bloqueados por razones incoherentes.
- Incapacidad de hacer productos agrupados y ofertas por cantidad de los mismos, esto se expande en el apartado “Desarrollo” el Capítulo.6, Catálogo.
- Métodos de envío obligatorios que requieren una gran inversión por parte del vendedor.

Como se puede ver estos problemas hacen costosa la compra al usuario, aumentando los precios hasta el doble de lo que estos son, y suponen una mayor complejidad a la hora del uso de la web.

Figura 1: Tienda de AlmaHeiko en la plataforma de Etsy.

Debido a estas razones, se decidió aprovechar el proyecto final para investigar y realizar esa tienda con la cual el cliente tendrá todo el control que requiera. Así se emplearán los conocimientos de todas las asignaturas para lograrlo. Siendo estos:

- Despliegue.

En primera instancia se pretendía realizar la aplicación en un servidor de hosting, pues esto otorga facilidades para la realización de algunas funciones como la configuración de un dominio o el servicio ssh. Debido al requerimiento de la

utilización de Docker y el servidor cloud empleados a lo largo del curso, esta idea fue descartada.

- Cliente.

Al ser el proyecto realizado en Wordpress el uso de javascript será muy reducido, pero sí que se le dará uso en esta aplicación para ciertas funciones.

- Servidor.

Similar a la situación anterior el uso de lo aprendido en esta asignatura será escaso pero también se usará en cierta medida.

- Diseño.

Siendo uno de los apartados que más varían de la aplicación web, este será el que más uso tenga, aunque también en una medida inferior en comparación a una aplicación sin el uso de Wordpress.

2.2. Objetivos (funcionales y técnicos)

Los objetivos principales indicados en el anteproyecto fueron ligeramente modificados tras la especificación de docker y clouding. Más concretamente, el objetivo que consta de subir la aplicación web a un servidor hosting el cual también ofrece un servicio de DNS y SSH fue valorado. No obstante, debido a la falta de tiempo, se consideró secundario y no pudo completarse.

Los próximos objetivos también se mencionaron en el anteproyecto, pero estos no se vieron afectados por el cambio ya indicado.

El principal y más general es la creación de una tienda para la venta de productos en su estado más básico, esto se compone de:

- La muestra de los productos en un inventario, así como la posibilidad de visualizarlos individualmente.
- Realizar la inserción de estos en un carro de la compra y el proceso de compra de los seleccionados.
- Una buena presentación visual de la página, en la que el usuario pueda navegar con facilidad y que esta sea una representación de la artista.
- Un buscador de productos tomando en cuenta las etiquetas y categorías de estos mismos, haciéndolo lo más cómodo y eficiente para los usuarios. Las opciones de filtrado estarán presentes en el listado de productos y el buscador se encontrará en todo momento en el menú superior de la página.

También se debería mencionar el objetivo que se nos indicó durante el desarrollo del proyecto, este sería que cualquier persona pudiera instalarse la aplicación en su propio equipo. Debido a ciertas complicaciones, logré realizar dos métodos para lograr este objetivo. Estos serán con el uso de docker Hub y con el uso de un plugin de Wordpress.

3. Recursos necesarios para el desarrollo

3.1. Recursos hardware

- Procesador Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.20 GHz
- RAM instalada 16.0 GB
- Sistema operativo de 64 bits, procesador basado en x64

3.2. Recursos software

Los recursos software empleados para el desarrollo de la aplicación fueron:

- Sistema operativo Window 10
- Servidor local XAMPP, en el cual los datos serían almacenados en una base de datos de la aplicación web mysql.
- Aplicación (CMS) Wordpress para servidor, esta aplicación se configura de manera “automática” en el servidor tras ejecutarla en el mismo e indicar las especificaciones correspondientes, y los plugins que se implementarán en esta.
- Aplicación Gimp, para edición de imágenes
- Aplicación Dia, para la creación del diagrama de clases
- Aplicación (web) Draw.io, para la creación de un diagrama de secuencia y de casos de la página

Los recursos software empleados en el servidor Clouding para el desarrollo de la aplicación fueron:

- Sistema operativo Linux debian 11
- Editor de textos Nano, para la edición de documentos de texto en el servidor.(apt install nano)
- Aplicación Docker, esta aplicación será empleada para la generación de los contenedores correspondientes para la aplicación, Dichos contenedores serán generados serán a partir de las siguientes imágenes.
 - o image: Wordpress
 - o image: mysql:5.7

4. Desarrollo del proyecto

4.1. Capítulos del desarrollo.

Para este apartado se realizará un recorrido por el desarrollo del proyecto, desde el principio de este hasta la finalización del mismo.

Capítulo 1: Servidores Hosting

Al comienzo de la aplicación se planteó la posibilidad de subir la aplicación en cuestión en un servidor hosting, debido a la multitud de webs que ofrecen muchos tipos de ofertas y servicios. A la hora de investigar se recogieron los precios completos ignorando las ofertas

(las cuales llegaban a ser del 40% a 85%), debido a que no se ha podido confirmar que estas se mantengan al renovar el hosting. Por consiguiente los cálculos realizados de cada una de las webs fueron llevados a cabo con los precios completos de los servicios en cuestión. Como resultado las mejores ofertas encontradas fueron las siguientes:

(las cifras están redondeadas en alza, para poder comparar precios con otros servicios y con el objetivo de no gastar por completo el presupuesto o que este no alcance a los servicios seleccionados)

- **IONOS**

Se investigó este servicio debido al gran número de usuarios con el que cuenta en España, y por los buenos precios que oferta. Una de las desventajas de esta plataforma es que solo ofrece dominios ".es" (correspondiente a España) y la intención de la tienda es que tenga un alcance internacional y por ello lo que se busca es que el dominio sea ".com" (correspondiente a comercio).

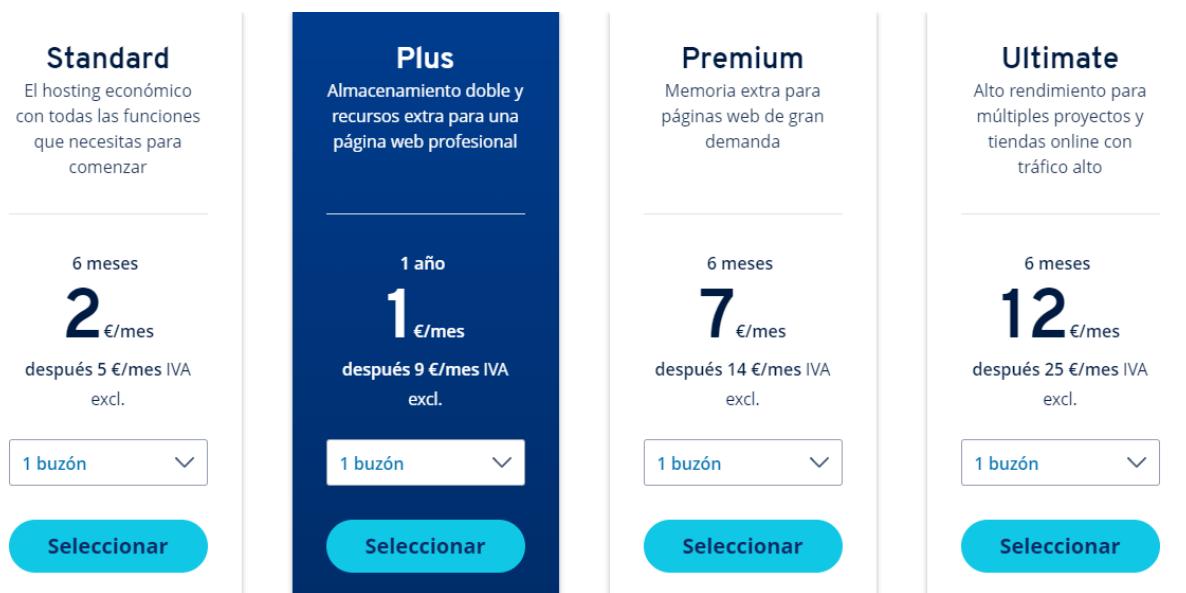


Figura 2: *Planes de servicio IONOS*

¡El dominio que has elegido está disponible!



Figura 3: *Precio dominio IONOS*

- **SITEGROUNDS**

Tras el comienzo del proyecto se investigó este servicio, no obstante, debido a sus altos precios en comparación con los ya comentados se decidió descartar.

A pesar de todo, este servicio es el más recomendado para el comercio, por su velocidad de carga y tiempo de actividad, así como un fuerte enfoque en la seguridad. Por ello tras la entrega de este proyecto, se planteará una comparativa

en detalle con el servicio escogido para tomar una decisión, pues actualmente esta se compondría de una opción más económica o una más eficiente y segura.



Figura 4: *Precio hosting SiteGrounds*

Registro de dominio: **GRATIS durante el 1er año** (después 19,99 €/año IVA no incl.)

Figura 5: *Precio dominio SiteGrounds*

- HOSTGATOR

El servicio de 3 años del Plan Emprendedor que otorga, en el cual eran alrededor de 60€ de hosting y 17€ de dominio al año.

En este plan se ofrece:

- Renovación de dominio y hosting
- SSH gratuito
- Cuentas de correo procesional ilimitadas
- 5 GB de SSD

Seguramente el Plan Personal fuera suficiente el cual es aproximadamente 20€ de hosting al año más barato que el anterior. Se concluyó que valía la pena por los 25 GB de SSD de memoria que ofrecía. Asegurando el poder implementar todos los productos y sus correspondientes imágenes en la mejor calidad posible,(aunque cabe destacar que Wordpress no permite imágenes que tengan mayor tamaño que 2 mg). Seleccionando este servicio como el que se aplicaría en la aplicación final.

Plan Personal

Para quienes están dando sus **primeros pasos**

\$116.20 USD **50% OFF**
\$58.10 USD equivalente a
\$1.63 USD/mes*

Contratar ahora

1 año de **Dominio Gratis** ⓘ
Cuentas de correo profesional **ilimitadas** ⓘ
Aloja un dominio con subdominios ilimitados
5 GB SSD para guardar los archivos de tu sitio
Certificado SSL gratuito

Plan Emprendedor

Para aquellos que **están comenzando**

\$187.01 USD **50% OFF**
\$93.51 USD equivalente a
\$2.60 USD/mes*

Contratar ahora

1 año de **Dominio Gratis** ⓘ
Cuentas de correo profesional **ilimitadas** ⓘ
Aloja un dominio con subdominios ilimitados
25 GB SSD para guardar los archivos de tu sitio
Certificado SSL gratuito

MÁS ECONÓMICO

3 años 50% OFF

Ahorra \$110.44 USD

¡Tu presencia en línea garantizada por los próximos 3 años!

Valor equivalente mensual

\$2.60 USD/mes*

De ~~\$186.99 USD~~
Por **\$93.50 USD** por 3 años

Obtén un año de **Dominio gratis** ⓘ

Figura 6.A: *Planes de servicio Hostinger*

Figura 6.B: *oferta por tres años del Plan emprendedor*

Figura 6: *Precio dominio Hostgator*

✓ Dominio disponible

tiendaproyectodaw05.com

GRATIS

Valor estimado de renovación **\$332.00 MXN**

17,37€

✓ ¡Buena elección! Tendrás un año de **dominio gratis**.
Esta promoción es válida para dominios de ciclo anual.

Figura 7: *Precio dominio Hostgator*

(Toda la investigación anterior, supuso un trabajo de aproximadamente 10 - 15 horas.)

Capítulo 2: Despliegue Contenedor Docker

Tras obtener la información de que el proyecto se desarrolla con docker en el servidor clouding, se investigó qué imagen sería necesaria para el desarrollo de la aplicación web correspondiente. Tras la investigación anterior se encontró en docker hub la imagen oficial de Wordpress (Figura 8), esta otorga la estructura de el docker-compose correspondiente (daw223-05/Instalacion/docker-compose.yml).

En la página de la imagen oficial se indica la posibilidad de instalar Wordpress con plugins y modificaciones directamente, pero teniendo el apartado gráfico de este mismo se decidió no emplear otros métodos con el objetivo de minimizar la aparición de errores.

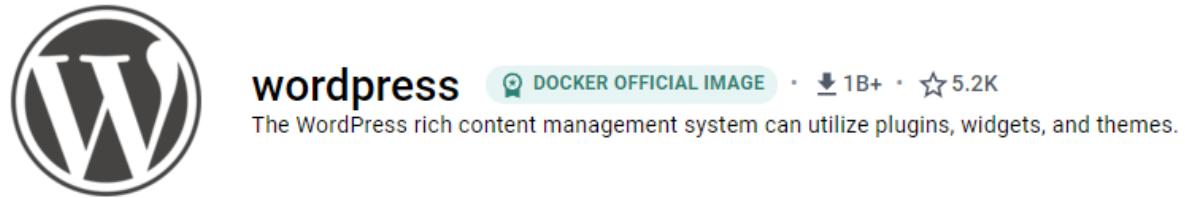


Figura 8: *Imagen oficial de Wordpress*

A continuación se realizaron los procesos correspondientes para la creación de los contenedores de Wordpress en el puerto 8080 y mysql(Figura 9). Una vez instalado se realizarán las configuraciones que exige, implementando el nombre de la página, el nombre de administrador, la contraseña, y el correo relacionado con este.

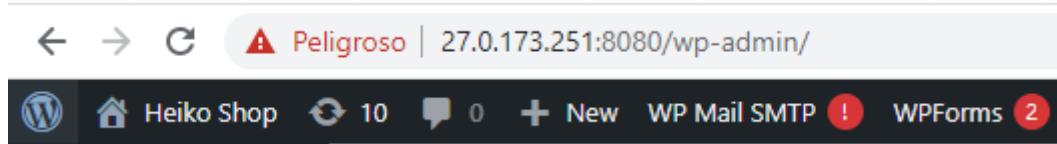


Figura 9: *Resultado de ejecución del contenedor docker Wordpress*

(Toda la investigación y proceso anterior, supuso un trabajo de aproximadamente de 5 - 8 horas.)

Capítulo 3: Autoformación sobre Wordpress y WooCommerce

Tras tener acceso al apartado gráfico de Wordpress se dejó de lado el proyecto por la realización de una investigación y autoformación sobre Wordpress y WooCommerce. La investigación correspondiente se dirigió al aprendizaje de estos campos en cuestión, comprobando que es lo más conveniente para el proyecto y en qué dirección a de evolucionar el mismo. El aprendizaje que se adquiría se ponía en práctica en un servidor local, realizando distintas pruebas e intentos con las herramientas que se descubrían.

(Toda la investigación y proceso anterior, supuso un trabajo de aproximadamente 10 -15 horas.)

Capítulo 4: Theme y Plugins

Antes de continuar creo que es necesario especificar la diferencia entre algunos de los componentes que se usarán en el proyecto, estas se separan en Themes y Plugins:

Un Theme es una plantilla o diseño que se utiliza para dar formato y apariencia a un sitio web. Controla la apariencia visual del sitio web, incluyendo el diseño de la página, el estilo y

la ubicación de los elementos como encabezados, menús, widgets y pies de página. Los documentos ".php" del mismo podrán ser modificados para cambios necesarios, aunque no es recomendable debido a los problemas que esto puede generar o la posible pérdida de estos mismos cambios.

Por otro lado, un plugin es una herramienta que se utiliza para añadir funcionalidades específicas al sitio web. Los plugins se utilizan para extender las capacidades de WordPress, agregando nuevas características o modificando las ya existentes. Los Plugins pueden ser descargados e instalados en el sitio web, y están disponibles para realizar una amplia variedad de tareas. WooCommerce por ejemplo es un plugin de Wordpress que ofrece la posibilidad de crear tu propia tienda con gran cantidad de ayudas y es el cual forma la columna de la aplicación web actual, cabe mencionar que también existen plugins que afecten a otros plugins, por lo cual es recomendable investigar el funcionamiento de estos mismos antes de su instalación.

Tras la explicación anterior se indicará el proceso que se llevó a cabo en el proyecto.

Se realizó una investigación y pruebas de distintos Plugins y Themes de Wordpress, de los cuales algunos estaban creados para compaginar con WooCommerce. Estas pruebas fueron realizadas en el servidor local, buscando lograr diferentes objetivos con estas mismas, estos se desarrollarán en la presentación de cada uno de ellos, y viendo la capacidad que tenían. Pues en ciertas ocasiones estas pruebas terminan concluyendo en que no lograban lo que buscaba concretamente o tenían un muro de pago para lograrlo, este pago era de entre 30 a más de 100 euros anuales. Cuando se aseguraba que el uso del plugin o Theme era necesario en el proyecto final, este se implementaría en el servidor clouding avanzando poco a poco el desarrollo del proyecto. Debido a que otro de los objetivos de estas pruebas fue evitar instalaciones innecesarias en la aplicación para luego eliminarlas, pudiendo estas dejar memoria ocupada o residuos que afecten a otros apartados.

(Toda la investigación y proceso anterior, supuso un trabajo de aproximadamente 20 horas.)

Capítulo 5: Theme y Plugins Utilizados

La investigación anterior dio como resultado un Theme y unos Plugins que merecen la pena ser comentados.

Primeramente, se explicará el Theme empleado para la página. Su nombre es de BlocksyY es suplementado con un segundo theme llamado Blocksy child. Este aparecerá como activo, pero depende de el theme padre Blocksy.

Ofrece una manera intuitiva y rápida para el desarrollo de una web, además de implementar opciones que compagan con el plugin de WooCommerce. A la hora de implementarlo se podrá acceder a un apartado que ofrece la posibilidad de seleccionar alguna de sus páginas de muestra se te da la opción de que la librería deseas utilizar, estas son Elementor y Gutenberg. En primer lugar está Elementor, la cual es una de las librerías más famosas de Wordpress, pues ofrece gran cantidad de características. No obstante, las herramientas que podrían resultar útiles, en este caso las que ofrece para WooCommerce, se encuentran solamente en la versión pro. Por ende se decidió usar la librería de Gutenberg, la cual es

perfecta para lo que se busca en la página, pues no es muy compleja y avanzada como Elementor pero cumple las expectativas necesarias perfectamente. Pues como se expandirá en el apartado “4.2.2. Diseño de la interfaz” este será muy útil en el desarrollo.

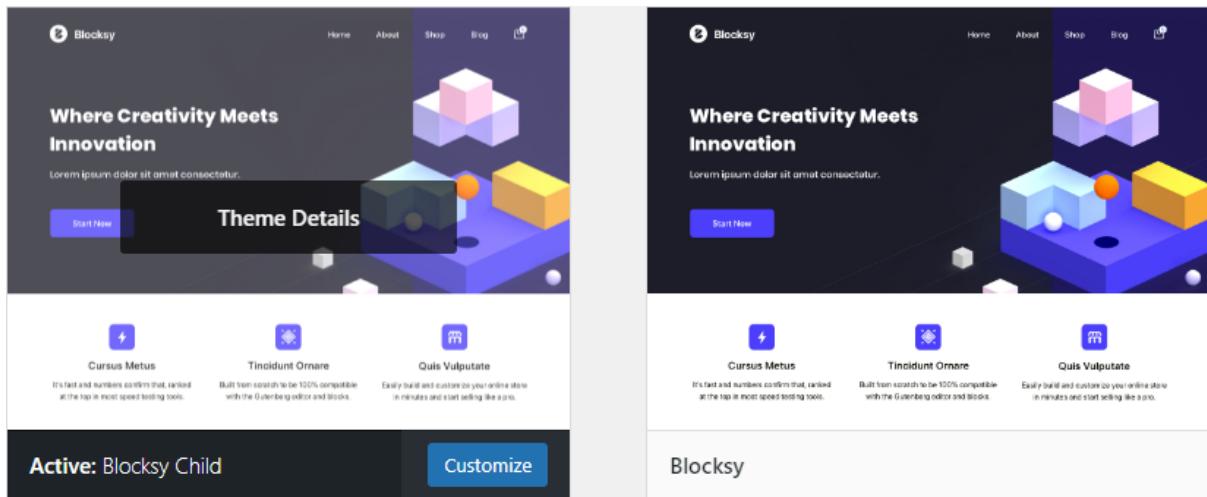


Figura 10: *Theme Blocksy y Blocksy child*

A continuación se indicarán los plugins que se instalaron en la aplicación y algunos convenientes mencionar, junto con sus representaciones en la aplicación wordpress:

- [Blocksy Companion](#):

Este plugin es una extensión que es recomendada instalar con el Theme Blocksy, pues introduce nuevas librerías de estilos y formatos además de integración con WooCommerce para la edición de ciertos apartados de la tienda.



Figura 11: *Plugin Blocksy Companion*

- [Stackable - Gutenberg Blocks](#):

En este caso este plugin se añade al implementar la librería Gutenberg que se mencionó en el theme Blocksy, pues ofrece 30 bloques adicionales de diseño a la biblioteca.



Figura 12: *Plugin Stakable*

- WooCommerce:

Ya se mencionó anteriormente como la columna del proyecto, pues este plugin ofrece todo lo necesario para la creación de una tienda online. De todo lo que ofrece utilice principalmente:

- El sistema de creación de productos, el cual ofrece gran cantidad de edición por producto. Esta se especifica en el apartado “Desarrollo” el Capítulo.6, Catálogo.
- Los métodos en los cuales los productos son mostrados individualmente y listados.
- El proceso de inserción en carro y pago de los productos del mismo.
- La posibilidad de implementar puntuación y comentarios a los productos
- El sistema de búsqueda de productos que ofrece.

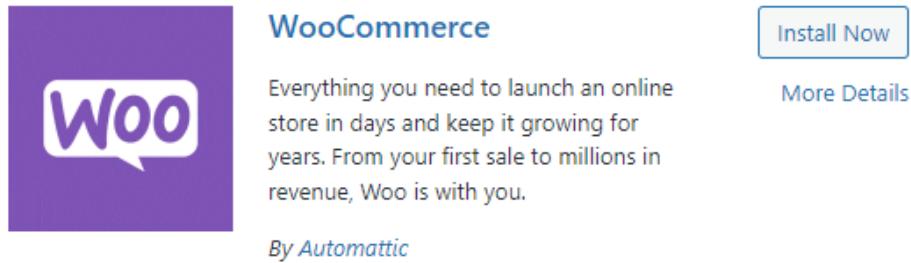


Figura 13: *Plugin WooCommerce*

- WooCommerce PayPal Payments:

Este plugin permite el pago por Paypal en el apartado de compra de WooCommerce, pues por defecto solo está el pago por transacción bancaria, el cual es algo tosco y complejo.

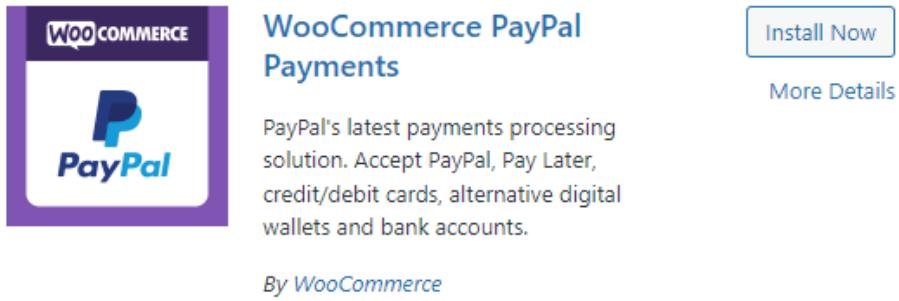


Figura 14: *Plugin WooCommerce Paypal Payments*

- WooCommerce Weight Based Shipping:

Para el control del peso de los productos en el envío es necesario un plugin, este es el que seleccione. Pues WooCommerce ofrece la posibilidad de implementar el peso en las características del producto, pero no ofrece que este se tome en cuenta en el envío. Y con este plugin puedo suplir esa necesidad.



Figura 15: *Plugin WooCommerce Weight Based Shipping*

- **WPC Product Bundles for WooCommerce:**

Debido a ciertas especificaciones que se indicaron para el proceso de selección de ciertos productos. En este caso la posibilidad de que (en los packs de productos) se ofrezcan ciertas ofertas fijas, dependiendo de la cantidad de productos, un ejemplo sería que un pack de 8 pegatinas por separado cada producto cuesta 1,99 pero si se selecciona la oferta de 2 por 2.75 estos son mas barato. Esto fue implementado en la tienda de Etsy de AlmaHeiko de manera muy tosca como se puede ver en la figura 16 y 18, forzando al usuario a indicar manualmente qué productos quiere de la oferta, debido a que no se podía conseguir este procedimiento de otra manera.

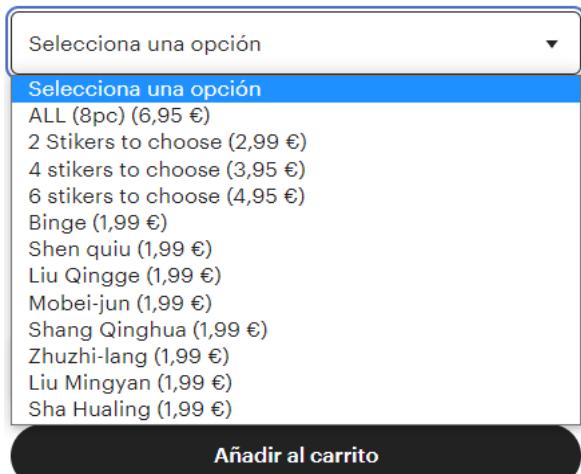


Figura 16.A: *Listado de opciones Etsy*

Figura 16.B: *Personalización pedido Etsy*

Figura 16: realización de *pedido Etsy*

Por ello con el uso de este plugin puede conseguir eso de manera más eficiente. Este genera productos agrupados a los que se podrá acceder y en estos podremos seleccionar los productos correspondientes que queremos añadir.

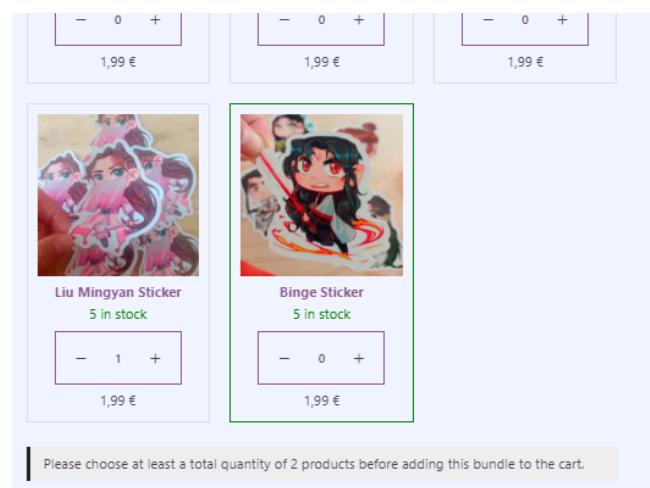


Figura 17: *Bundle de pack de pegatinas oferta 2 por 2.75€*



Figura 18: *Plugin WPC Product Bundles for WooCommerce*

- **Hide Categories and Products for WooCommerce:**

Una especificación que se indicó fue que productos individuales que pertenecen a grupos, solo fueran accesibles desde los packs correspondientes, sin que estos aparecieran en el listado general, esto se puede conseguir fácilmente con el apartado de visibilidad del producto, pero este se vuelve inaccesible desde el grupo correspondiente. Por ello se investigaron maneras distintas de lograrlo hasta dar con este plugin, el cual logra lo necesario utilizando una categoría la cual indicará cuales no han de aparecer y cuales sí, pudiendo acceder a los productos igualmente después desde el grupo.



Figura 19: *Plugin Hide Categories and Products for WooCommerce*

- **WP Code**

Este plugin tiene como objetivo insertar snippets de código en el interior de la estructura de la página, sin la necesidad de modificar el código base del Theme para lograr realizar los cambios que se buscan. Este implementa HTML, CSS, JavaScript y PHP.



Figura 20: *Plugin WPCode*

- **WPForms Lite:**

La función principal de este plugin es la de creación de formularios de contacto desde

una interfaz más fácil e intuitiva.



wpforms

Contact Form by WPForms
– Drag & Drop Form
Builder for WordPress

[Install Now](#)

[More Details](#)

The best WordPress contact form plugin.
Drag & Drop online form builder to
create beautiful contact forms, payment
forms, & other custom forms ...

By WPForms

Figura 21: *Plugin WPForms Lite*

- **WP Mail SMTP:**

El uso de este plugin es el envío de un correo electrónico a través de un formulario generado con el plugin anterior (WPForms Lite). Pero esto genera errores, los cuales parecen ser debido a que es necesario el pago de servicios externos para el correcto funcionamiento de este. Por ello se mantuvieron los plugins para demostrar la intención que se buscaba. Pudiendo realizar el formulario y mandarlo, aunque esta última función no se complete.



**WP Mail SMTP by
WPForms – The Most
Popular SMTP and Email
Log Plugin**

[Install Now](#)

[More Details](#)

Make email delivery easy for WordPress.
Connect with SMTP, Gmail, Outlook,
SendGrid, Mailgun, SES, Zoho, + more.
Rated #1 WordPress SMTP Email plugin.

By WP Mail SMTP

Figura 22: *Plugin WP Mail SMTP*

- **All-in-One WP Migration**

El plugin en cuestión fue instalado para lograr el apartado “7.1 Manual de instalación”, debido a que en un comienzo se pensó que este sería imposible debido a cierto detalle . La funcionalidad de este es realizar una exportación de la web en un formato concreto (.wpress), y tras esto en una web de Wordpress diferente en la cual esté instalado este mismo plugin se podrá importar la página web con el archivo correspondiente. Esto sobreescibirá la página creando una copia exacta de tu página migrada.



All-in-One WP Migration

[Install Now](#)

[More Details](#)

Move, transfer, copy, migrate, and
backup a site with 1-click. Quick, easy,
and reliable.

By ServMask

Figura 23: *Plugin All in One WP Migration*

- [WooCommerce Payments](#) (No instalado/Ni en uso):

Este plugin ofrece más métodos de pago a la hora de realizar la compra de los productos, como pagos con tarjeta de diferentes bancos. Pero a la hora de realizar las configuraciones de este plugin exige confirmaciones legales que me preocuparon, pues hay algunas que no soy responsable de contestar (pues es información que el cliente ha de indicar), este imformacion luego podría chocar si se vuelve a implementar en una nueva tienda (en este caso la que realizare después de la práctica) y por petición de el cliente me solicitó esperar a la web comercial para este apartado.

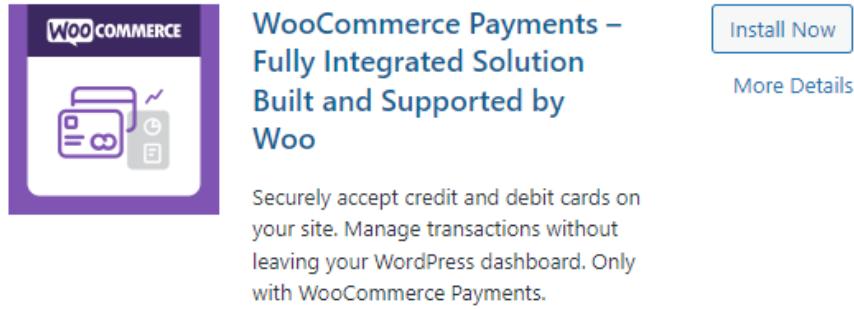


Figura 24: *Plugin WooCommerce Payments*

Los próximos plugins mencionados no estarán instalados en la aplicación, pero sí que es conveniente mencionarlos, para poner de ejemplo algunos casos en los cuales no se implementaron muchos de los plugins que se llegaron a investigar.

- [Block Slider](#) (No instalado/Ni en uso):

Este plugin acompaña a el theme the Blocksy con Gutenberg. Añade la posibilidad de implementar sliders en la web creada de manera más simple y personalizada. Pues estos podrán ser auditados externamente con detalle para implementarlos fácilmente en la web. Tiene mucho potencial pero por tiempo se decidió desechar para centrarse en otras funciones.



Figura 25: *Plugin Block Slider*

- [Product Price by Quantity for WooCommerce](#) (No instalado/Ni en uso):

Este plugin fue instalado para lograr conseguir lo que se comentó en el plugin (WPC Product Bundles for WooCommerce), pero este plugin no lograba lo que se necesitaba

específicamente y por ello no se implementó.



Figura 26: *Plugin Product Price by Quantity*

- [WPC Grouped Product for WooCommerce](#) (No instalado/Ni en uso): Debido a un problema que el plugin (WPC Product Bundles for WooCommerce) generaba a la hora de implementar bundles a grupos, se realizaron pruebas con este plugin buscando solucionarlo, pero tenía un muro de pago para la cantidad de productos que se podían implementar, y por eso fue descartado.



Figura 27: *Plugin WPC Grouped Product*

Capítulo 6: Apartados del proyecto

Tras conocer lo que se implementó en el proyecto se puede explicar mejor el desarrollo del mismo, pues solo se mencionaron las herramientas. El desarrollo se separara en cuatro apartados: Diseño, Configuración, Contenidos y Modificación.

(En los próximos apartados son el desarrollo que se llevó a cabo en el servidor local y el servidor clouding en conjunto para el control del tiempo, pues los procesos son muy similares al final)

Diseño

En este apartado se utilizó lo aprendido en la investigación sobre Wordpress y WooCommerce que se mencionó anteriormente. Pero el desarrollo del mismo se explicará en el apartado “4.2.2. Diseño de la interfaz”

(Toda la investigación y proceso anterior, supuso un trabajo de aproximadamente 20 - 30 horas.)

Configuración

Muchos de los plugins implementados tienen una configuración interna para especificar el objetivo de estos mismos, como el propio plugin de WooCommerce, el cual pide detalles de la “empresa”, o el plugin de Weight Based Shipping que es necesario indicar los detalles de

los precios por pesos correspondientes. También añadir la configuración a herramientas externas como a la cuenta de Paypal Developer, la cual ofrece una cuenta de prueba para realizar pagos y otra para recibirlos de esta.

(Toda la investigación y proceso anterior, supuso un trabajo de aproximadamente 10 horas.)

Catálogo

Tras tener las configuraciones correspondientes de la tienda se comenzaron a introducir los productos que se venderán en esta. Existe la posibilidad de introducir estos mismos desde un archivo Excel, pero como se implementaron plugins que afectan a algunos productos, para evitar fallos estos fueron introducidos manualmente, con los formularios que WooCommerce ofrece en Wordpress. Estos productos tienen una gran cantidad de edición, de la cual me gustaría detallar unos apartados, estos se pueden ver señalados en la figura 31.

- Product name: Como este indica es el título del producto, el cual también estará presente en el link de este, además de ser el dato a tomar en cuenta en el sistema de búsqueda.
- Product Description: en el cual podemos poner la descripción del producto
- Product Data: Es el apartado más importante, pues en este se indicarán muchos detalles, estos detalles pueden variar dependiendo de la selección de product data que indiquemos.
 - Simple, el cual es un producto básico sin ningún tipo de condición
 - Variable, este tiene una condición que ofrece distintas versiones del mismo producto
 - Smart bundle, pertenece al plugin WPC Product Bundles.
 - Agrupado, en este se implementaran los productos que se mencionaron anteriormente para generar grupos y sea más fácil acceder a estos mismos.

Aunque generalmente en este apartado se indica la cantidad de inventario y cómo manejarlo, el peso y precio, etc.

- Product Imagen y Product Gallery: en las cuales se mostrarán las imágenes relacionadas con el producto.
- Product Categories: Con los productos que se proporcionaron se decidió que se implementarían 6 categorías en las que se organizaron los productos.
- Product Tags: Con los productos que se me proporcionaron se decidió implementar Tags dependiendo de detalles generales sobre los producto, como que es o a qué serie pertenece.

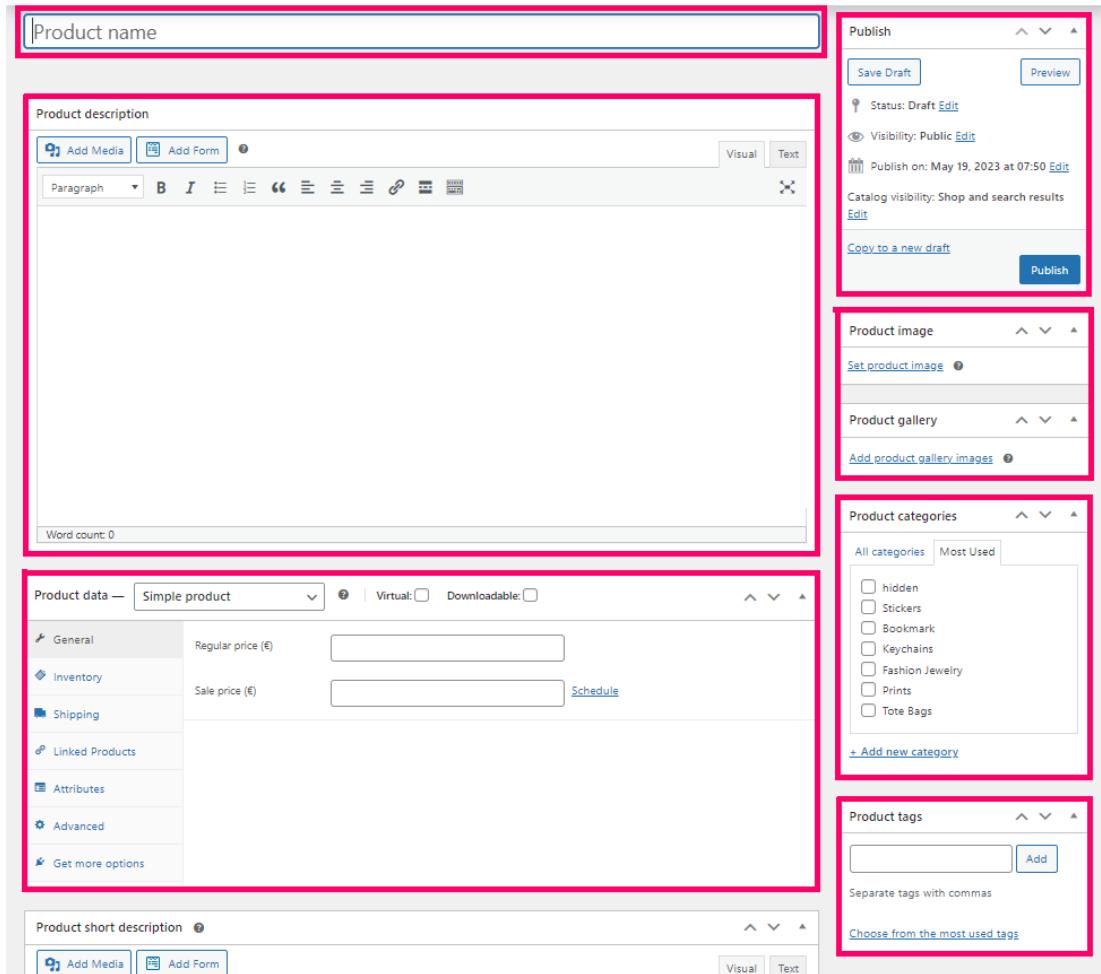


Figura 28: Pestaña Add New WooCommerce

(Toda la investigación y proceso anterior, supuso un trabajo de aproximadamente 10 -15 horas.)

Modificación

En este apartado se comentará el uso de código que implemente en el proyecto. Pues se modificaron ciertos aspectos con funciones javascript en el proyecto, que se ejecutaban tras que el apartado php de la pagina fuera generado. Al comienzo de este desarrollo se realizaron en el código base del Theme, pero luego se descubrió el plugin WP Code, el cual realizaba estos mismos procesos pero evitando que las funciones se pierdan a la hora de actualizar el theme, y quede más limpio. La explicación de las funciones en el apartado “4.3 Código Fuente” de manera más detenida.

(Toda la investigación y proceso anterior, supuso un trabajo de aproximadamente 15 horas.)

Capítulo 7: Información legal

También se realizó una pequeña investigación sobre cómo se deberían completar los apartados de Términos y condiciones, además de las políticas de privacidad. Para las políticas de privacidad se utilizó la página por defecto que proporciona el Theme de Blocksy editando para

cuadrar con la web. Para los términos y condiciones se busco que era necesario indicar, entonces se siguió el cuestionario que proporciona la web termify.io, este solo sirvió de guía pues al completarlo es necesario pagar por conseguir generar uno con los datos implementados.

No se entró en mucho detalle durante el desarrollo de este apartado debido a que en los apartados legales serán investigados más en profundidad en el futuro.

The screenshot shows a website header with a logo 'Alma Heiko' featuring a red stylized flower or heart. The menu includes links for HOME, PRODUCTS, THE ARTIST, and CONTACT, along with a search icon and a shopping cart icon showing 1 item. The main content area has a light blue background and displays the following sections:

- Privacy Policy**
- Who we are**

Our website address is: <http://27.0.173.251:8080/>
- Comments**

When visitors leave comments on the site we collect the data shown in the comments form, and also the visitor's IP address and browser user agent string to help spam detection.

An anonymized string created from your email address (also called a hash) may be provided to the Gravatar service to see if you are using it. The Gravatar service privacy policy is available here: <https://automattic.com/privacy/>. After approval of your comment, your profile picture is visible to the public in the context of your comment.
- Cookies**

If you leave a comment on our site you may opt-in to saving your name, email address and website in cookies. These are for your convenience so that you do not have to fill in your details again when you leave another comment. These cookies will last for one year.

Figura 29: *Página Política de privacidad*

The screenshot shows a website header with a logo 'Alma Heiko' featuring a red stylized flower or heart. The menu includes links for HOME, PRODUCTS, THE ARTIST, and CONTACT, along with a search icon and a shopping cart icon showing 1 item. The main content area has a light blue background and displays the following sections:

- Terms and Conditions**
- Effective Date: 24/06/2023**
- This document sets forth the terms and conditions for the use of the HeikoShop website (the "Website") located at <http://27.0.173.251:8080>. Please read these terms and conditions carefully before using the Website.
- By accessing or using the Website, you agree to be bound by these terms and conditions. If you do not agree with any part of these terms and conditions, you must not use the Website.
- 1. Personal Information Collection:**

We collect personal information from users, including name/username, phone number, email addresses, and passwords, in order to provide our services. By using the Website, you consent to the collection and use of your personal information as described.
- 2. Email Communications:**

We may send emails or newsletters to users who opt-in to receive them. By providing your email address, you agree to receive such communications from us. You can unsubscribe or opt-out at any time.
- 3. Digital Analytics:**

We do not use any digital analytics software for tracking purposes.
- 4. Advertising:**

Our Website does not display ads.
- 5. Facebook Pixel:**

We do not have the Facebook pixel installed on our Website.
- 6. Retargeting:**

We do not use retargeting techniques for advertising.

Figura 30: *Página Términos y condiciones*

(Toda la investigación y proceso anterior, supuso un trabajo de aproximadamente 3 horas.)

Capítulo 8: Backups

Como se mencionó anteriormente, el proyecto fue desarrollándose en el servidor clouding directamente. Y para la seguridad del mismo se realizaron copias de los directorios de los contenedores docker correspondientes, por si en caso de que surgiera algún error poder subir estos mismos. Se realizaron distintas pruebas con estos para que estos se guardarán de manera segura en el interior del repositorio GitHub que nos indicaron utilizar, pero estos directorios eran un tamaño muy grande, inclusive si estos están comprimidos y Github no permite esas capacidades a no ser de que sea instalada una extensión para ello (Git Large File Storage). Entonces se descartó esa posibilidad, dejando solo backups en el interior del servidor.

Pero tras la investigación del capítulo siguiente, se decidió utilizar el plugin All-in-One WP Migration para el guardado de backups. Ya que este tiene un apartado para estos mismos.

(Toda la investigación y proceso anterior, supuso un trabajo de aproximadamente 5 horas.)

Capítulo 9: Instalación

Comentar también el desarrollo del apartado “7.1 Manual de instalación” la cual es un manual de instalación para que cualquier persona sea capaz de copiar el proyecto en otro dispositivo. Al comienzo se intentó el uso de un docker-compose que utilizará imágenes generadas de los propios contenedores. Pero debido a un detalle que no se tomó en cuenta al principio del proyecto, se supuso imposible este proceso.

Entonces se realizó una investigación para poder arreglar este problema, en esta se encontró el plugin All-in-One WP Migration, el cual lograba realizar una copia directa del proyecto, esta es posible importarla en una nueva aplicación Wordpress, la explicación de este proceso se podrá ver en el método 2 del apartado “7.1 Manual de instalación”.

Pero tras decidir que este sería el método a utilizar para la instalación, se intentaron más pruebas para lograr el objetivo del apartado con los procesos intencionados, este se consiguió. Finalizando con dos métodos para instalar el proyecto, por si fuera necesario.

(Toda la investigación y proceso anterior, supuso un trabajo de aproximadamente 10 - 15 horas.)

4.2. Diagrama relacional y diagramas de clases

Este apartado es algo complejo de realizar, puesto que la base de datos de Wordpress está hecha para el propio funcionamiento de este mismo. Y el plugin empleado a lo largo del proyecto, en este caso WooCommerce, complica mucho más esta base de datos.

La organización de Wordpress es muy cerrada, pues está estructurada de forma que sólo es comprendida por este mismo o los plugins que se instalen. Por ello realizar modificaciones directamente en la base de datos es muy complejo y peligroso.

En las siguientes figuras muestran la estructura de Wordpress y WooCommerce de manera superficial

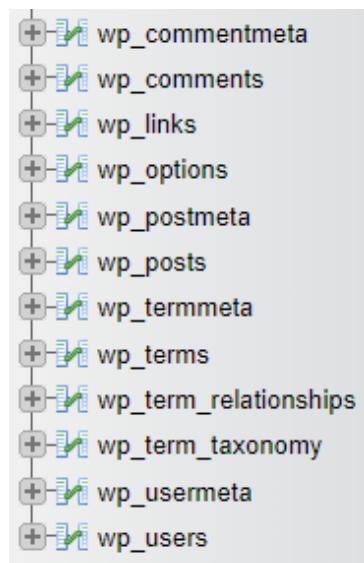
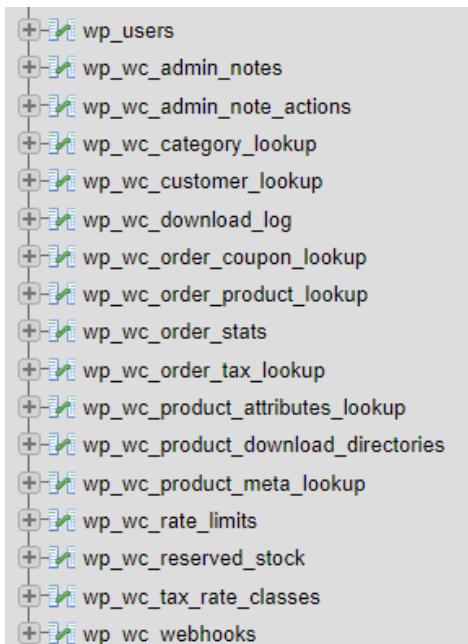


Figura 31: Base de datos de Wordpress sin ningún plugin implementado.

Las tablas individuales mostradas en la figura 31, son la organización básica de la aplicación Wordpress. A continuación, se detalla brevemente la información básica de algunos de los elementos que componen dicha organización.

- wp_users: guarda la información de los usuarios de Wordpress.
- wp_posts: son algo similar a objetos con ciertos de datos, esta tabla solo muestra todos ellos pero no sus datos.
- wp_postmeta: son todas las relaciones entre los datos de los post con los post en cuestión



Tras la instalación de WooCommerce, se generarán nuevas tablas. En este caso no se consideró necesaria la investigación del funcionamiento interno de cada uno de los componentes debido a la densidad de información de todos ellos. Esto se demuestra en la figura 32, que nos muestra una porción de todas las tablas que WooCommerce genera al ser instalado. No obstante, se podría decir que la manera de organizar los productos es más fácil de entender en el apartado visual de WooCommerce.

Figura 32: *Base de datos de Wordpress tras la instalación de WooCommerce .*

Diagrama relacional

En un principio, un “boceto” de la base de datos que se hubiese estructurado sería el siguiente, se tendrá acceso a el archivo de las imágenes en el contenedor GitHub (daw223-05/Tablas_Esquemas/BocetoEsquema.dia):

La base de datos almacenará información sobre productos, incluyendo su nombre, peso, precio, valoración general y descripción. Los productos estarán clasificados en categorías y cada uno pertenece a una sola categoría. Además, los productos estarán etiquetados con tags, que son palabras clave que permiten una fácil búsqueda y que pueden ser compartidos entre varios productos.

Los usuarios tendrán una cuenta con su correo electrónico y nombre. Los usuarios podrán generar comentarios sobre los productos, que incluirán una valoración numérica y un texto descriptivo. Cada comentario estará asociado a un producto específico y al usuario que lo haya generado. Cada usuario solo podrá generar un comentario por producto.

Los usuarios también podrán realizar compras, seleccionando uno o más productos. La información sobre las compras se almacenará en la base de datos y estará relacionada con el usuario correspondiente

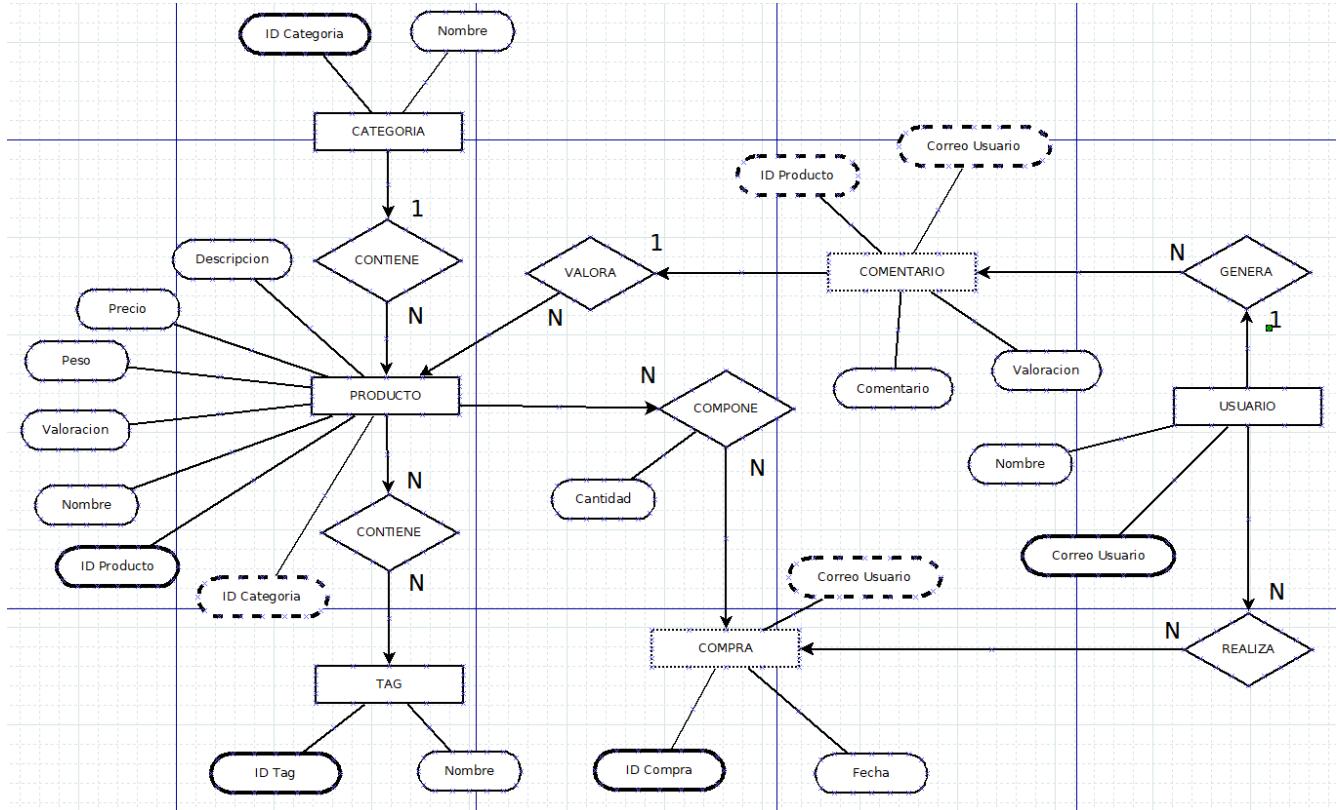


Figura 33: Boceto diagrama relacional base de datos

- La tabla Producto está relacionada con la tabla Categoria a través de una clave foránea (ID Categoria).
- La tabla Producto está relacionada con la tabla Tag a través de una tabla intermedia Producto-Tag. Relación CONTIENE.
- La tabla Comentario está relacionada con las tablas Usuario y Producto a través de claves foráneas (Correo Usuario e ID Producto).
- La tabla Compra está relacionada con la tabla Usuario a través de una clave foránea (Correo Usuario).
- La tabla DetalleCompra está relacionada con las tablas Compra y Producto a través de una tabla intermedia Producto-Compra (ID Compra e ID Producto). Relación COMPONE.

Aunque la estructura de base de datos Wordpress y WooCommerce sea cerrado y complejo a la hora de su comprensión. Las funcionalidades que se buscan con el boceto mostrado logran el mismo resultado que la base de datos en cuestión aunque no se explique con detalle su proceso.

Pues como se mencionó anteriormente, WooCommerce ofrece la posibilidad de insertar productos con características similares a las indicadas en el diagrama, además de la posibilidad de realizar compras. Pero no se implementa directamente la posibilidad de generar un usuario para esto mismo, aunque el usuario sí que realice las funcionalidades no puede crear cuenta.

4.3. Interacción con el usuario:

4.3.1. Casos de uso.

Para comenzar se explicaran las páginas principales por las cuales el usuario se desplaza.

Home:

Página principal, la cual nos mostrará de manera visual las categorías de los productos que se ofrecen, además de mensajes dirigidos al usuario, y finaliza con los productos más populares de la tienda junto con un acceso a la página “The Artist”.

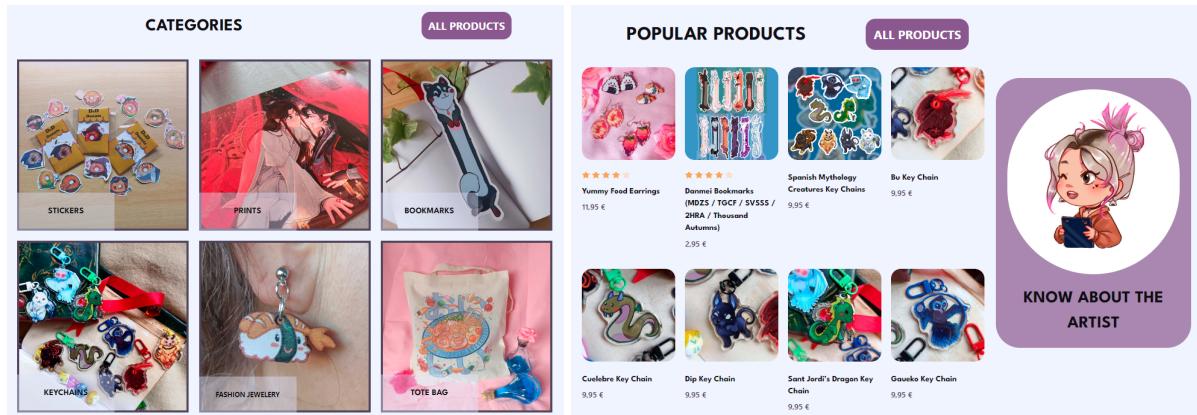


Figura 34.A: Categorías Home

Figura 34.A: Productos Populares Home

Figura 34: Página Home

The artist:

Página donde se presentará información de la artista (@AlmaHeiko). Con una descripción de esta, además de enlaces donde poder conocer más información, terminando con los estudios donde trabajó.

Alma Cantero Escribano

Hello, I'm Alma and I have just finished my latest internships. I graduated in 2D Animation from U-fad and completed the Concept Art Master at Taller. Over the past two years, I have also worked as a Concept Artist at animation companies such as Anima, Kinection, Pentakill Studios, and Smile and Learn.

I especially love Concept Art in all its forms and the ability to create anything with the right skills. As a consumer of series and video games, I highly value the pre-production and concept art phase, considering it to be the most crucial for achieving a successful outcome as it lays the foundation of a project.

I am constantly seeking to develop my skills and expand my knowledge within the animation and video game industry; I aim to grow both personally and professionally in order to achieve the best possible results.

If you want to know more, please don't hesitate to contact me. I'll be delighted to provide further information.

I stream on Twitch
Here I usually stream when I draw or I play videogames with friends, feel free to join for a chat.

I'm active in social media
I usually upload content of my work, projects and life events I attend.

My artstation
Here you can find my work and projects.

My linkedin
Here you can find more information about me.

I've Worked In

PENTAKILL STUDIOS, ANIMA, SMILE AND LEARN

Figura 35.A: The artist parte superior

Figura 35.A: The artist parte inferior

Figura 35: Página The Artist

Contact:

En esta página se ofrece información de contacto para los clientes, desde contacto directo con la artista a contacto de soporte por teléfono y correo electrónico. También está implementado un formulario que otorga la posibilidad de contactar con servicio técnico sin necesidad de usar la aplicación de correo electrónico directamente.

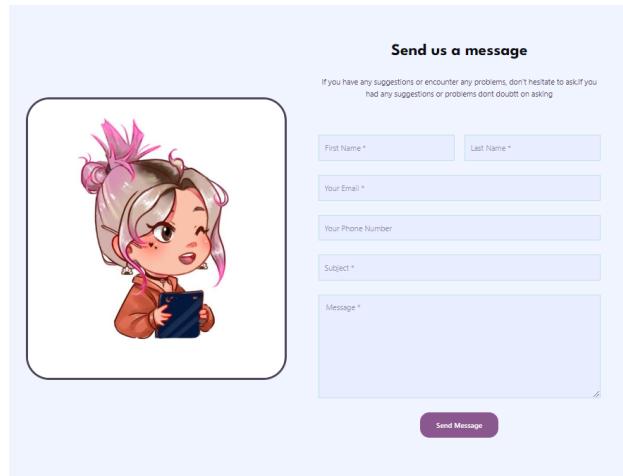


Figura 36: *Formulario Contact Us*

Products:

Esta es la página desde la cual los productos serán listados y se podrán realizar ciertos filtros para que la búsqueda sea lo más precisa posible. Las opciones de estos filtros están en el sidebar al lado del listado junto con un acceso directo a la página “The Artist”.

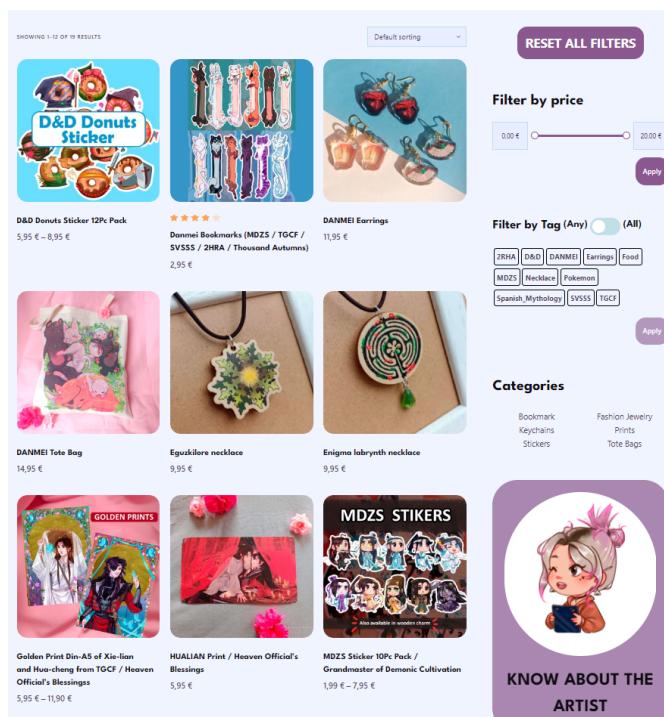


Figura 37: *Listado de productos y sidebar*

Single Product:

Desde esta página se mostrarán los detalles del producto en cuestión. Desde las imágenes que lo representan, su descripción, categoría y tags. El usuario (si el producto en cuestión lo permite o no, dependiendo si es un producto perteneciente a un grupo o no) podrá poner comentarios/reviews con puntuación que afectarán al producto valorado.

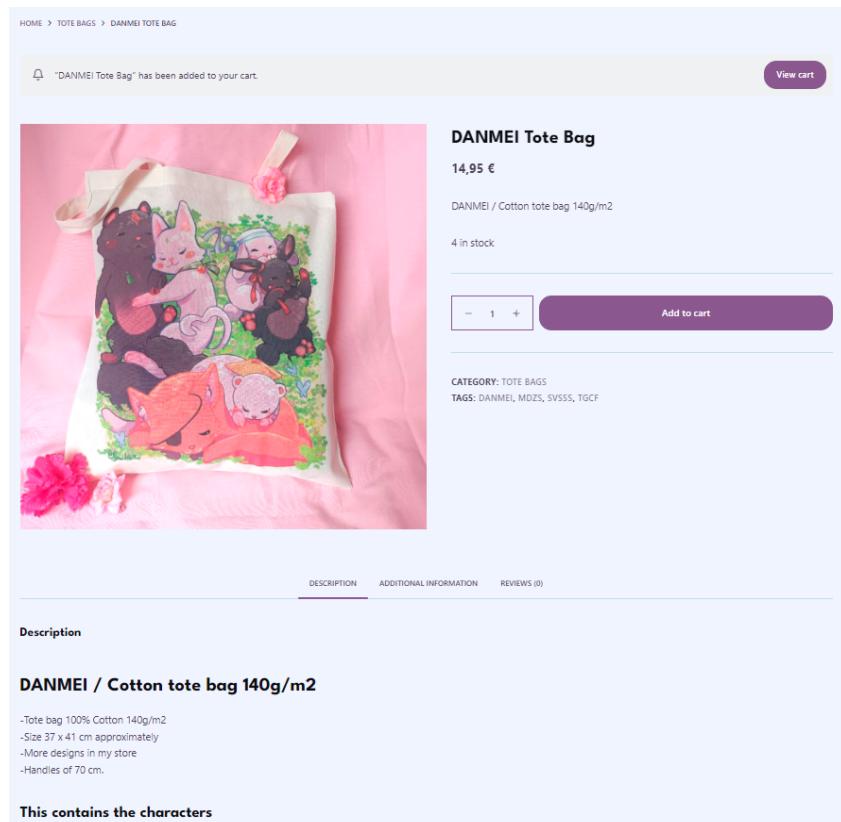


Figura 38: Página Single product de Tote Bag

Search Product:

Esta página servirá para listar los productos filtrados por nombre con los datos que ofreció el usuario.

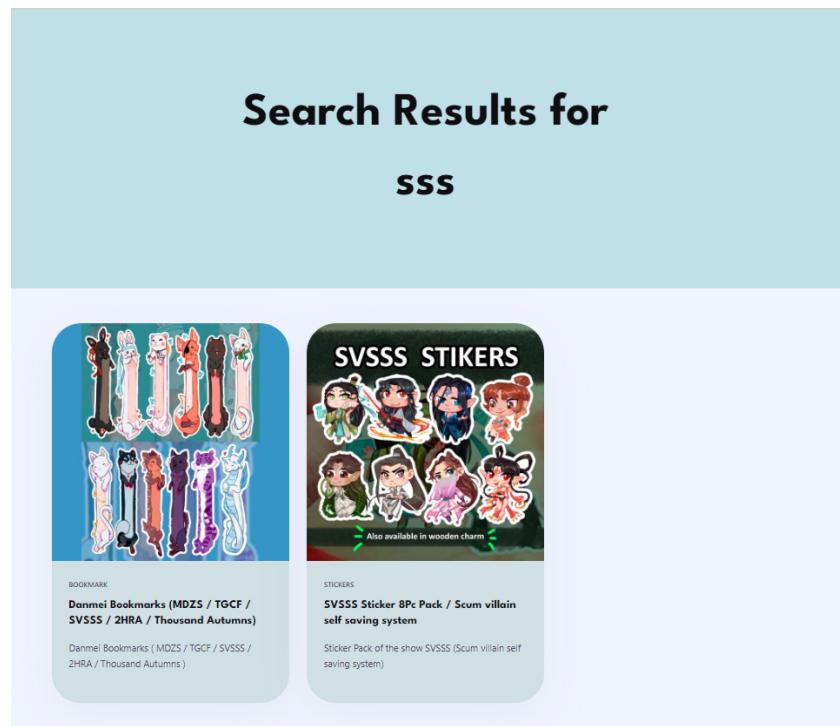


Figura 39: Página Search en funcionamiento

Cart:

La finalidad de esta página será mostrar los productos que se están dispuestos a comprar, y la posible edición de estos mismos, aumentando su cantidad o removiendo los de el carro.

Product	Quantity	Subtotal
DANMEI Tote Bag 14.95 €	- 1 +	14.95 €

[Update cart](#)

Cart totals

Subtotal	14,95 €
Shipping	Calculate shipping
Total	14,95 €

[Proceed to checkout](#)

Figura 40: Carrito de la compra

Checkout:

La función de esta pagina sera la finalización de la compra, en la cual se especificará la dirección de envío y se ofrecerán los métodos de pago correspondientes

Checkout

Billing details

First name *	Last name *
Alberto	Cantero

Company name (optional)

Country / Region *

Taiwan

Street address *

Calle falsa

Apartamento falso

Town / City *

Ciudad Falsa

State / County *

Provincia Falsa

Postcode / ZIP *

123456

Phone *

555000555

Email address *

emailfalso@gmail.com

[Ship to a different address?](#)

Order notes (optional)

Notes about your order, e.g. special notes for delivery.

Your order

Product	Subtotal
DANMEI Tote Bag x 1	14.95 €
Subtotal	14.95 €
Shipping	Weight Based Shipping: 10.55 €
Total	25,50 €

Direct bank transfer

PayPal

Pay via PayPal.

PayPal

SOFORT

Your personal data will be used to process your order, improve your experience on this website, manage access to your account, and other purposes described in our privacy policy.

Figura 41: Página Checkout para finalizar pago.

Tras ello el desplazamiento por la página se compondría de lo siguiente.

La página tendrá en todo momento un menú superior y el footer de la página permite el acceso en todo momento a las páginas “Home”, “Products”, “The Artist”, “Contact”, “Search Product” y “Cart”.



Figura 42: Página Checkout para finalizar pago.



Figura 43: Página Checkout para finalizar pago.

Diagrama de casos del usuario

En el diagrama de casos de el usuario, nos mostrará las funcionalidades que el usuario podrá realizar en el interior de la página.

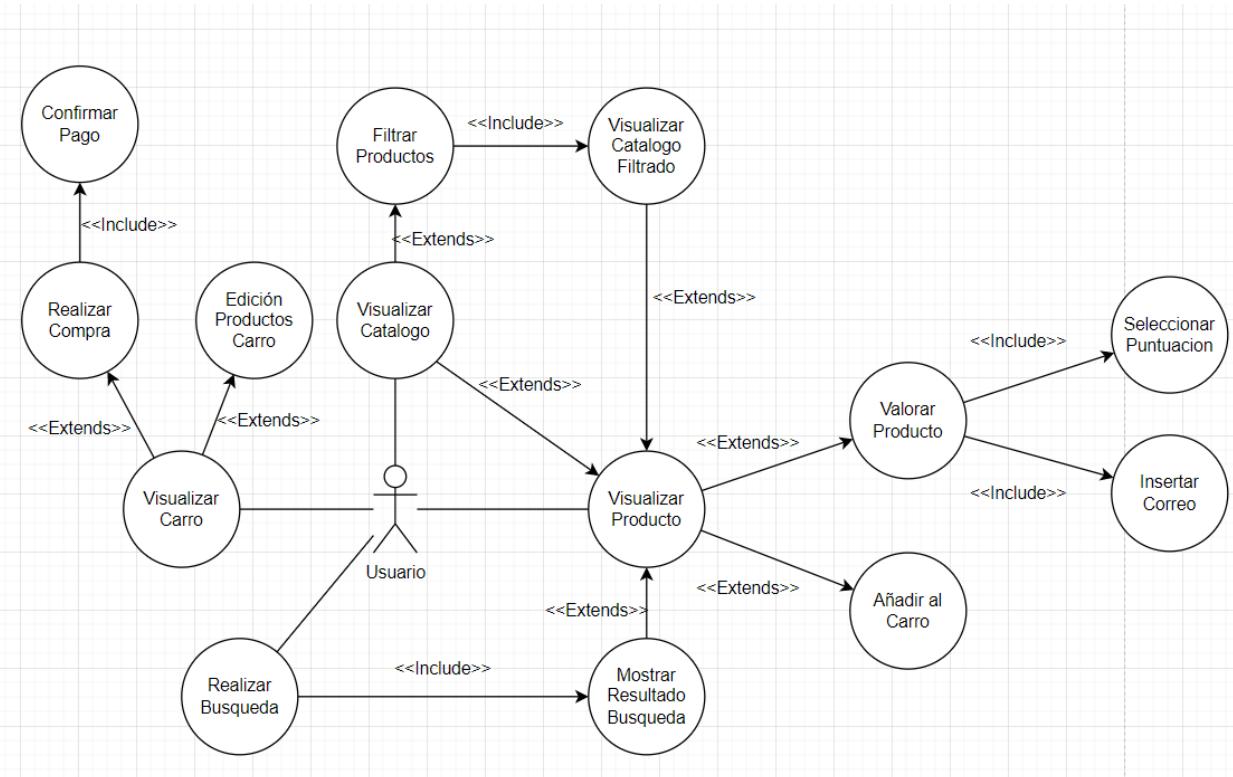


Figura 44: Diagrama de casos del usuario.

Diagramas de secuencia de las operaciones del usuario

Siguiendo el diagrama de casos anterior, se mostrarán los diagramas de secuencia de las operaciones que el usuario podrá realizar.

Realizar Búsqueda

Este diagrama muestra el comportamiento de la operación de búsqueda que realiza el usuario. Esta operación podrá ser ejecutada en cualquier momento durante la estancia en la página. Se compone de el usuario realizando la búsqueda, está realiza la búsqueda correspondiente en la base de datos. Devolviendo la página "Search Product" con un listado de los productos encontrados, que se podrán seleccionar y acceder a estos.

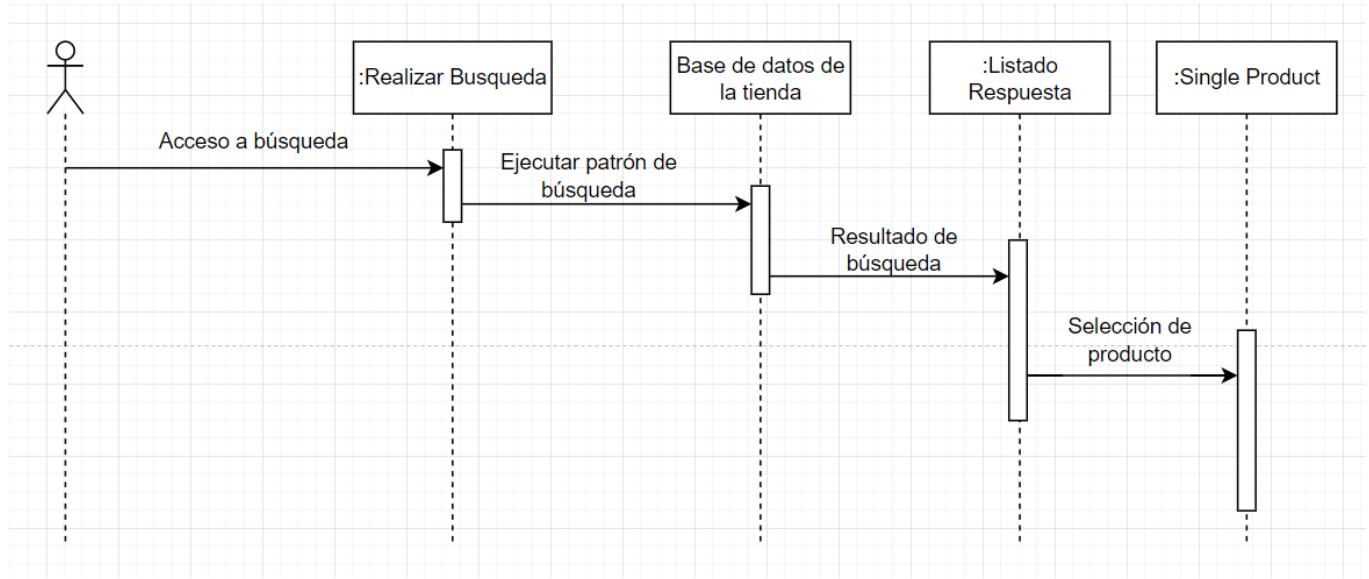


Figura 45: Diagrama de secuencia de búsqueda.

Catálogo de productos

Este diagrama muestra la interacción que un usuario puede realizar sobre el listado de productos de la página “Products”. La interacción principal es la aplicación de filtros, los cuales actualizan el listado dependiendo de estos.

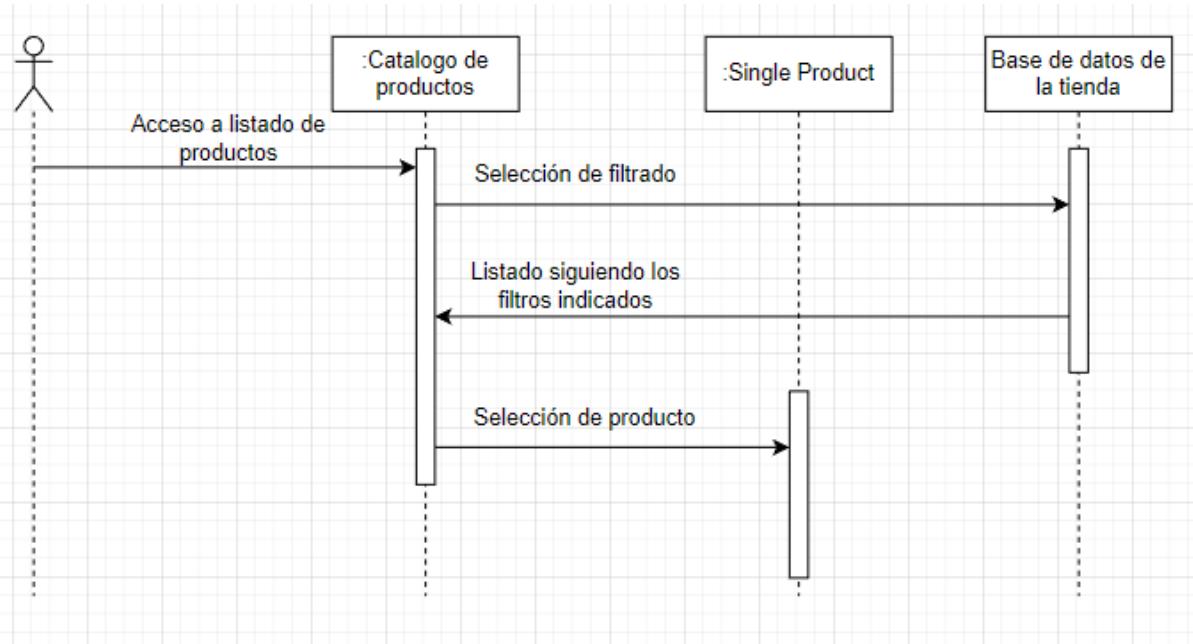


Figura 46: Diagrama de secuencia del catálogo.

Visualizar Producto

Este diagrama indica las interacciones que realiza un usuario en el interior de la página “Single Product”. Estas son:

- La inserción del producto en el carrito de la compra, el cual actualiza la página para que los cambios se muestren.
- La valoración del producto, en la cual este se generará en la base de datos y un administrador se hará cargo de confirmar la publicación de este en el producto.

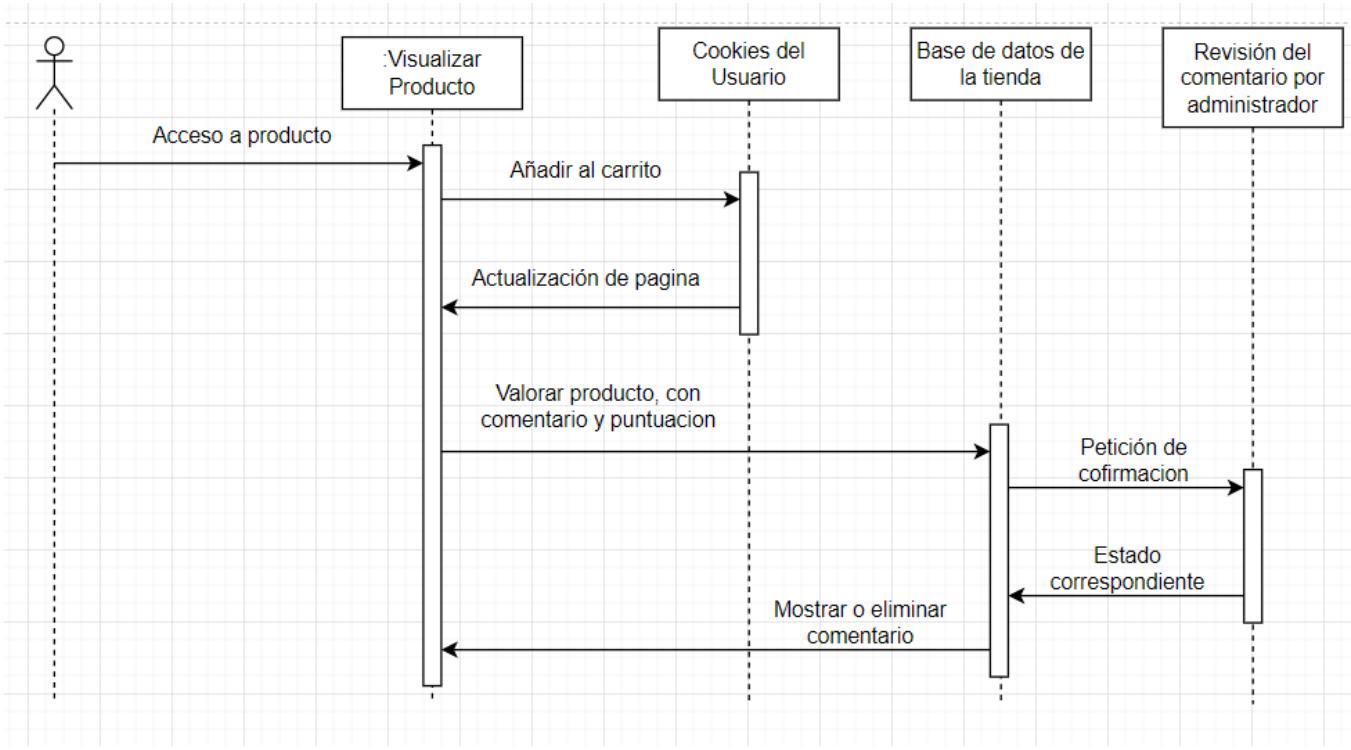


Figura 47: Diagrama de secuencia del producto.

Carro

Este diagrama muestra el comportamiento de las interacciones que realiza el usuario en la página “Cart”. Estos son:

- La edición de los productos del mismo, que actualiza la página para mostrar los resultados.
- La realización del pedido, el cual nos ofrece un formulario que tras completarlo confirmando el pago, la cantidad de los productos correspondientes se actualiza y se genera un pedido el cual tiene estados relacionados con el.

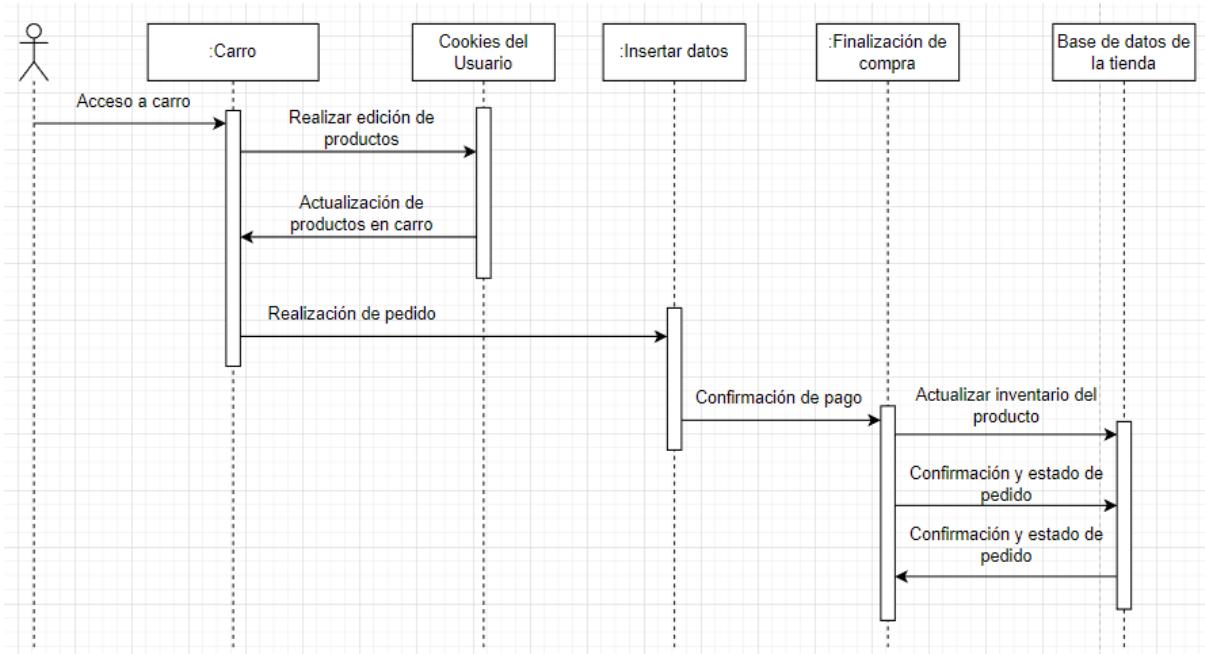


Figura 48: Diagrama de secuencia del carro.

4.3.2. Diseño de la interfaz

El proceso de diseño comenzó con la creación de múltiples ideas que poder llegar a implementar en la estructura y diseño de la página, como animaciones, colores y otros, pero tras realizar las investigaciones sobre Wordpress y WooCommerce se descubrieron ciertas herramientas para realizar este apartado de manera mucho más sencilla y óptima, manteniendo la posibilidad de implementar las ideas desarrolladas tras el uso de estas.

Estas fueron el theme Blocksy , el plugin Stackable , la librería de Gutenberg y el plugin WP Code (desarrollado en el apartado 4.3. Código fuente). Gracias a Blocksy se pueden generar webs iniciales que sirvan de base para lo que quieras conseguir, desde blogs, servicios, hasta tiendas, que es lo que nos ocupa ahora. Concretamente fue seleccionada “Home Decor” (<https://demo.creativethemes.com/blocksy/home-decor/>) la cual era perfecta para usar de base para mi proyecto, pues ofrecía un listado de productos, header, footer y sidebar por mencionar los que más me beneficiaron.

El footer y sidebar, entre otros se pudieron editar por separado desde el apartado de widgets de Wordpress, se mencionó anteriormente pero no se describió en detalle, un widget es un componente visual compuesto por bloques que puede ser agregado, eliminado y editado en diferentes áreas de la interfaz de la web. Esta edición será acompañada gracias a los bloques que ofrece Stackable pudiendo modificarlos hasta lograr el objetivo correspondiente de cada uno y el plugin WP Code el cual permitió una edición y creación más personalizada a partir de código.

Después están las ediciones más generales que Blocksy nos deja editar, como la paleta principal de la web, el header de esta o cómo está compuesto, la organización y detalles del listado de productos o la página de compra de los mismos.

Por último las páginas como “Home” o “The artist”, que fueron desarrolladas manualmente con el editor de páginas de Wordpress, las cuales se apoyan de los bloques de Stackable para los apartados de estas mismas.

Se buscó que el diseño fuera lo más agradable y fácil para el usuario posible, evitando poner animaciones o detalles innecesarios. Antes de comenzar el proyecto el plantamiento de este era el opuesto, pero tras plantear las ideas visuales que se pretendían implementar a la página a bastantes fuentes (amigos y familiares). Estas fuentes aconsejaron que evitará tales distracciones visuales y simplificará en favor del usuario.

También se buscó mostrar unos colores y estilo que muestre una representación del artista detrás de los productos, usando colores rosados y morados, junto con azules claros para acompañar.

4.4. Código fuente.

Al utilizar Wordpress el código fuente que se desarrolló directamente no es mucho, pues lo que compone todo Wordpress y WooCommerce es, aparte de complejo, muy cerrado y modificar este directamente es muy peligroso además de que este se puede ver sobreescrito por una actualización. Pero sí que es importante mencionar los cambios que fueron implementados para lograr los objetivos correspondientes.

Las modificaciones fueron implementadas al código con el plugin WP Code. Estas funciones son ejecutadas tras que el php de la página sea cargado, pudiendo hacer así las operaciones necesarias.

Estas funciones podrán ser revisadas (por no ocupar todo el espacio de la memoria) en GitHub (daw223-05/Modificaciones/{nombre_de_la_funcion})

Estos son los siguiente, con su versión correcta y su versión sin modificar visualmente:

TextoSwitch

Esta función implementa un HTML el cual muestra un título para la barra de progreso de los filtros por Tag, y un switch el cual indicará la opción en cuestión a tomar en cuenta a la hora de filtrar estos mismos entre Any (que filtra los productos por los que tengan al menos un Tag de los indicados) y All (que filtra los productos por los que tengan los Tag indicados)

Filter by Tag (Any)

Figura 49: apartado TextoSwitch.

BarraTags

Esta función implementa un HTML el cual muestra una “barra de progreso”, pues como tal simula una, para los filtros por Tag.

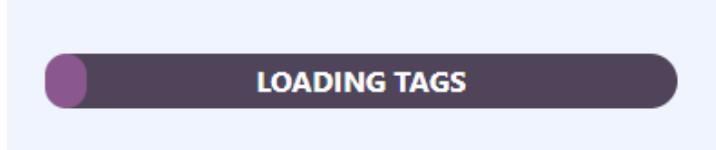
LOADING TAGS

Figura 50: apartado BarrasTags.

checkedActivoinactivoTag

Con las implementaciones anteriores está la función recoge estos bloques y otros más para realizar las siguientes funcionalidades dependiendo de ciertos estados. Los estados en cuestión son si el filtro de tags es cambiado, si se encuentra en una página de Tag o de productos sin filtro, el tipo de filtro activo, entre otros.

Estos factores afectan a la visualización y funcionamiento de la barra de progreso y los filtros And y All por tag.

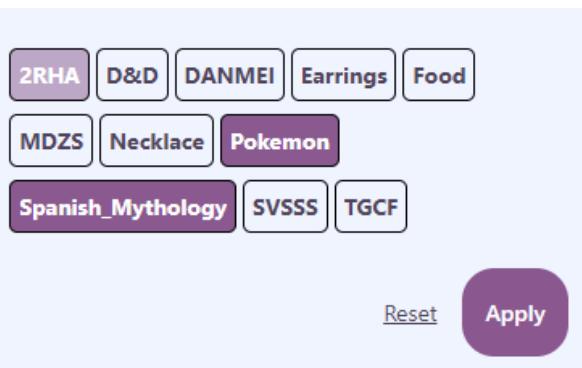


Figura 51: apartado checkedActivoinactivoTag.

checkedActivоНactivoCategori

Su finalidad es detectar si se está en una página de categoría y en los listados de estas se marcará la correspondiente.

Categories

Bookmark	Fashion Jewelry
Keychains	Prints
Stickers	Tote Bags

Figura 52: apartado *checkedActivоНactivoCategori*.

productSalesChange

Esta función tiene como objetivo mostrar de manera correcta los bundles de oferta por cantidad, de los productos agrupados que los poseen, por ejemplo el pack de 8 pegatinas, pues al implementar los bundles en el grupo estos implementa un contador para realizar su compra, pero esta es imposible pues es necesario acceder a este para poder indicar cuales del grupo se quieren elegir, pues si “suma” la cantidad en el grupo e intenta realizar la compra esta dará un error. Entonces para evitar este problema se seleccionan las celdas correspondientes a los bundles, los cuales se detectan gracias a las clases con los que se generan, y se cambian por un texto diferente.

SALE	SVSSS Sticker 8Pc Pack (Completo Pack of 8, or 8 of your selection)	FOR ONLY 6,95 €
SALE	SVSSS Sticker 8Pc Pack (6 stickers to choose)	FOR ONLY 4,95 €
SALE	SVSSS Sticker 8Pc Pack (4 stickers to choose)	FOR ONLY 3,95 €
SALE	SVSSS Sticker 8Pc Pack (2 stickers to choose)	FOR ONLY 2,99 €
	<input type="button" value="-"/> <input type="text" value="0"/> <input type="button" value="+"/> Zhuzhi-lang Sticker	1,99 € 5 in stock
	<input type="button" value="-"/> <input type="text" value="0"/> <input type="button" value="+"/> Shen Quiu Sticker	1,99 € 5 in stock

Figura 53: apartado *productSalesChange*.

CssTienda

Gran parte de lo mencionado se vio complementado por esta implementación. La cual es un css con las clases correspondientes que algunas funciones utilizan o ediciones específicas externas a las funciones para lograr ciertos objetivos.

5. Fase de pruebas

5.1. Pruebas realizadas

En este apartado , se evaluará el funcionamiento y la usabilidad de la aplicación. Siguiendo una serie de pautas que se deben cumplir para que el nivel de usabilidad de la aplicación sea lo más alto posible. Estas son:

- Ponerse en el lugar del usuario y pensar como nos gustaría encontrar la información.
- Desarrollar un correcto árbol de navegación sin excesivos enlaces.
- Un buen diseño visual que transmita lo propuesto sin que produzca confusión.

Con estas pautas se realizaron las pruebas correspondientes desde el principio del desarrollo, realizando las pruebas para asegurar la ejecución y uso de los apartados de la web.

Estas pruebas se realizaron en cada implementación que se presentaba e incorporaba a la aplicación, siguiendo el proceso que esta llevaba hasta su finalización. Comprobando que todo se realiza como se espera logrando los objetivos pertinentes.

Las pruebas a destacar son:

La navegación por la página:

Esta prueba fue una que se realizó de manera constante, pues cada cambio que se realizaba en la interfaz de las páginas tenía que ser comprobado para asegurarse de que este se imprimía en la página correctamente y realiza la función que se propone, además de asegurar que al deformarse la página este sea responsive.

La realización de búsqueda:

Este apartado tuvo gran cantidad de pruebas realizadas para lograr que la búsqueda fuera por descripción, pero esto se acabó descartando debido a la complicación para el proyecto que esto supuso.

La inserción de valoraciones:

Al tener la capacidad de comentar productos en la página oficial de Etsy, se realizaron las pruebas correspondientes para comprobar que la funcionalidad era similar a la que se buscaba.

La realización de compra y edición de carro:

El proceso de compra ha de ser cómodo y simple para el usuario, por ello se buscó que este fuera lo más intuitivo a partir de la realización de pruebas de estas operaciones.

6. Conclusiones finales

Ha de comentarse que esta práctica fue un proceso de aprendizaje para un proyecto personal que se realizará en un futuro, con los mismos objetivos cumplidos en esta página, pero con muchas funcionalidades y aspectos que se quieren implementar y mejorar.

Estos son los siguientes:

El uso de un servidor hosting con servicio dns y ssh.

Implementar todo el catálogo de productos que el cliente ofrece, pues en esta tienda implemente una cuarta parte por tiempo, cubriendo los mínimos y necesarios para presentar el funcionamiento del proyecto.

Un buscador por descripción de producto funcional, pues en este proyecto intenté implementarlo en este, pero fracasó, entonces decidí posponerlo para el proyecto final.

El uso de Block Slider para la página principal de la tienda, haciendo esta más llamativa.

Mejorar el funcionamiento de WPForms y WP Mail SMTP para que realice las funcionalidades correctamente o buscar otras herramientas para lograr esto mismo.

Con la función anterior, poder implementar un apartado de encargos y diseños personalizados. En el cual estos se podrán realizar y se podrán mostrar en esa misma página los resultados de estos y muchos otros que se hayan realizado.

Implementación de más métodos de pago a la hora de realizar la compra de productos.

Un sistema de notificaciones a usuarios interesados, que tras introducir su correo electrónico reciban avisos sobre la tienda, desde notificaciones de productos nuevos, rebajas, etc.

Para lograr esto es necesario el uso de unos servicios externos los cuales pueden llegar a ser gratuitos o de pago, pero en la mayor cantidad de los casos es necesario un dominio. Además de las implementaciones de MailPoet para lograr más posibilidades a los correos electrónicos mandados y las razones de los mismos.

Estos son los que más entusiasmo y tiempo quiero dedicar tras completar el curso.

7. Documentación del sistema desarrollado

7.1. Manual de instalación

En este apartado veremos cómo poder instalar la página web en cuestión en el interior de un servidor o dispositivo debian. Durante el proyecto se prepararon dos métodos distintos para realizar este apartado

Cuando estos se completen logrando así una copia exacta de la web en cuestión. Si se quiere acceder al wp-admin de esta página será necesario el usuario y contraseña de la página original (usuario:allberto,contraseña:ProyectoAlbertoDaw05).

Método.1. Uso de imágenes personalizadas

1. Actualizamos el servidor

Desde el servidor ejecutamos `apt update`, este actualizará el servidor. También realizamos la instalación de nano con el comando `apt install nano`, esta aplicación nos permitirá editar ficheros de texto con facilidad.

```
root@srv1:~# apt update
```

Figura 54: comando de actualización en servidor.

```
root@srv1:~/proyecto# apt install nano
```

Figura 55: comando de instalación de aplicación nano.

2. Directorio para el proyecto

Generamos una carpeta en el interior de este equipo con el comando `mkdir proyecto`, donde guardaremos los archivos y ficheros correspondientes además de realizar las correspondientes operaciones. Accederemos a este mismo con el comando `cd proyecto`.

```
root@srv1:~# mkdir proyecto  
root@srv1:~# cd proyecto/  
root@srv1:~/proyecto#
```

Figura 56: creación de carpeta proyecto.

3. Instalación de docker

Entonces continuamos con el comando de instalación de docker, `apt install docker docker-compose docker.io`

```
root@srv1:~/proyecto# apt install docker docker-compose docker.io
```

Figura 57: comando de instalación de aplicación docker

4. Ejecutamos contenedor Portainer

Por si fuera necesario, instalaremos el contenedor Portainer. Este nos ofrece una manera visual para visualizar los contenedores, imágenes y más detalles de la ejecución de docker. Para ello implementaremos el siguiente comando, con el cual logramos la ejecución de Portainer en el puerto 9000.

```
docker run -d -p 8100:9000 --name=portainer --restart=always -v  
/var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest  
root@srv1:~/proyecto# docker run -d -p 8100:9000 --name=portainer --restart=always -v /var/run  
/docker.sock:/var/run/docker.sock -v portainer data:/data portainer/portainer-ce:latest
```

Figura 58: comando de creación y ejecución container Portainer

5. Ejecución de docker-compose.yml

En el interior de Github se encuentra un archivo docker-compose.yml, este será necesario para la ejecución de los contenedores del proyecto. Será necesario realizar una pequeña modificación a este mismo, con el comando `nano docker-compose.yml` podremos editarlo y

en el interior de este los apartados que pone “{TU IP}”, hay que sustituirlos por la ip del equipo o servidor en el que se están realizando las operaciones, tras realizar la modificación pertinente lo guardamos pulsando Ctrl + O.

```
version: '3.1'

services:

  wordpress:
    image: alberto0505/wordpress-tienda:1.0
    restart: always
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
      WORDPRESS_CONFIG_EXTRA: |
        define( 'WP_HOME', 'http://{TU IP}:8080' );
        define( 'WP_SITEURL', 'http://{TU IP}:8080' );

    volumes:
      - wordpress:/var/www/html

  db:
    image: alberto0505/db-tienda:1.0
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql

volumes:
  wordpress:
  db:
```

Figura 59: *documento docker-compose.yml* método 1

Este documento nos importará las imágenes del proyecto subidas a docker Hub. Con el uso del comando `docker-compose -f {nombre del fichero} up -d` dentro del repositorio en el que se encuentre el documento.

```
root@srv1:~/proyecto# docker-compose -f docker-compose-propio.yml up -d
Creating network "proyecto_default" with the default driver
Creating volume "proyecto_wordpress" with default driver
Creating volume "proyecto_db" with default driver
Creating proyecto_wordpress_1 ... done
Creating proyecto_db_1 ... done
```

Figura 60: *ejecución del docker-compose.yml*

Este nos generará el proyecto al que se podrá acceder con el enlace `{servidor o equipo local}:8080/` dando por finalizado este procedimiento.

Método.2. En cualquier aplicación wordpress

En este método comenzaremos, desde cualquier aplicación wordpress instalada, en este caso será una aplicación wordpress generada con la imagen oficial de wordpress con el docker-compose que nos proporciona la documentación de esta misma.

Siguiendo los 4 primeros pasos del método anterior, realizando operaciones distintas desde el apartado 5.

5. Ejecución de docker-compose.yml

Con la documentación que nos ofrece la imagen oficial de wordpress generamos el docker-compose.yml, este será necesario para generar los contenedores.

```
version: '3.1'

services:

  wordpress:
    image: wordpress
    restart: always
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exempledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_DATABASE: exempledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql

volumes:
  wordpress:
  db:
```

Figura 61: *documento docker-compose.yml método 2*

Para generar este a mano con el comando `nano docker-compose.yml` directamente, este nos pondrá a editar lo. Introducimos la información correspondiente (cuálquiera de los dos es válido, y lo guardamos pulsando `Ctrl + O`.

Tras ello ejecutaremos el documento docker-compose que generará un contenedor Wordpress en el puerto 8080 y un contenedor mysql. El comando correspondiente que ha de ser usado es `docker-compose -f {nombre del fichero} up -d` en el directorio donde se encuentre, como se muestra en el método anterior en la figura 60.

6. Inicio de la aplicación Wordpress

Con lo anterior completo, podremos acceder al contenedor Wordpress con el enlace `{servidor o equipo local}:8080/` nada más acceder a este link nos enviará a la configuración

correspondiente de Wordpress. Estas configuraciones no tendrán mucha importancia, pues se sobreescibirá más adelante.

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password Very weak Hide

Important: You will need this password to log in. Please store it in a secure location.

Confirm Password Confirm use of weak password

Your Email

Double-check your email address before continuing.

Search engine visibility Discourage search engines from indexing this site

It is up to search engines to honor this request.

Figura 61: formulario configuración Wordpress

7. Instalación del Plugin All-in-One WP Migration

Tras completar la configuración correspondiente e ingresar el usuario y contraseña que acabamos de indicar estaremos en la interfaz de Wordpress, en el menú de la izquierda se encontrará una opción llamada Plugins, en esta seleccionamos Add New / Añadir Nuevo.

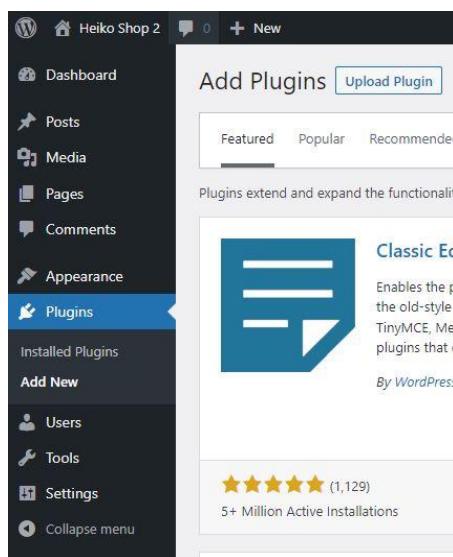


Figura 62: barra lateral izquierda Wordpress

En el buscador de la página en la que nos encontramos escribimos "All-in-One WP Migration" y seleccionamos el correspondiente cuyo autor es "ServMask", como se ve en la figura 23 de la página X.

Después de instalarlo y activarlo recargamos la página, y en el menú de la izquierda se habrá añadido una nueva opción para operar con el plugin, la cual tendrá una opción de importar.

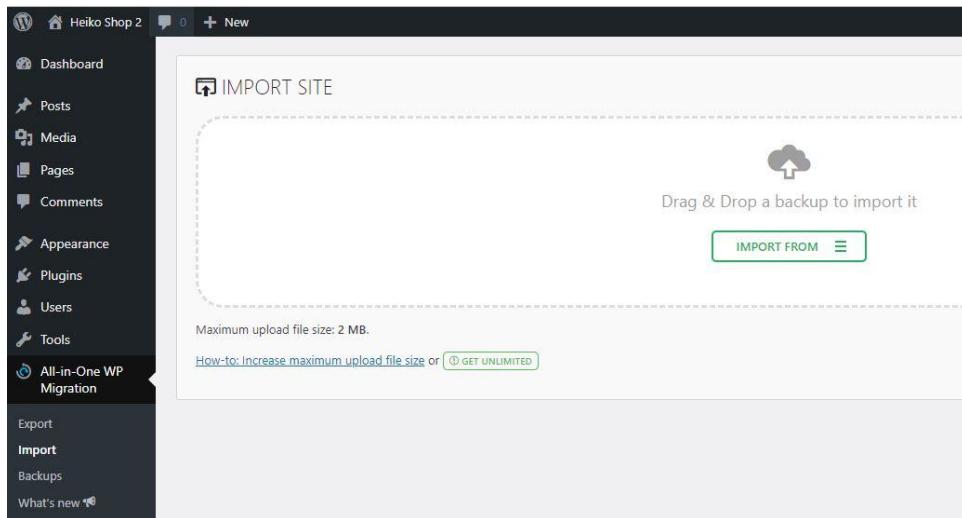


Figura 63: *interfaz plugin All-In-One*

Antes de continuar con la importación, hay un paso más que se debe llevar a cabo, pues el archivo a importar es de un tamaño mayor a el permitido por Wordpress, como se puede ver el máximo de capacidad que permite es de 2 MB.

3. Configuracion capacidad maxima de Wordpress

Para lograr que Wordpress nos permita importarlo tendremos que realizar un cambio en un documento que se encuentra en el contenedor de este mismo, el cambio permitirá aumentar el máximo a 500MB.

Para lograr esto primero accederemos a el contenedor, necesitaremos el id que se generó para el contenedor, para lograr saber cual es se ejecutará `docker ps` este comando nos mostrará los contenedores en ejecución y los ids de los mismos. Entonces implementamos el id o nombre del contenedor Wordpress en el comando siguiente `docker exec -it {nombre o id del contenedor} /bin/bash`. Tras la ejecución de este estaremos en el interior del contenedor docker de Wordpress.

```
root@srv1:~/proyecto# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
7580d9fee805        mysql:5.7         "docker-entrypoint.s..."   7 minutes ago     Up 7 minutes      3306/tcp, 33060/tcp   proyecto_db_1
52a9dd475a0b        wordpress          "docker-entrypoint.s..."   7 minutes ago     Up 7 minutes      0.0.0.0:8080->80/tcp   proyecto_wordpress_1
root@srv1:~/proyecto# docker exec -it proyecto_wordpress_1 /bin/bash
root@52a9dd475a0b:/var/www/html#
```

Figura 64: *listado de contenedores y acceso exec*

Desde aquí realizaremos nuevamente el comando `apt update`, para actualizar el contenedor, tras ello realizaremos `apt install nano`, esto nos permitirá poder editar el interior del fichero, este documento se encuentra en la carpeta `var/www/html/` que posiblemente será en la que estemos nada más acceder.

```
root@52a9dd475a0b:/var/www/html# apt update
```

Figura 65: *comando de actualización en contenedor.*

Entonces Realizamos las modificaciones correspondientes con el comando `nano` y el documento `.htaccess` (el punto es necesario). En el interior de este insertamos las

siguientes líneas de código, las cuales nos aumentaran la capacidad de los archivos que Wordpress permite importar, y guardamos con Ctrl + O .

```
GNU nano 5.4 .htaccess
# BEGIN WordPress
# The directives (lines) between "BEGIN WordPress" and
# dynamically generated, and should only be modified via .htaccess
# Any changes to the directives between these markers will be overwritten.
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteRule .* - [E=HTTP_AUTHORIZATION:{HTTP:Authorization}]
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>

php_value upload_max_filesize 500M
php_value post_max_size 500M
php_value memory_limit 500M
php_value max_execution_time 0
php_value max_input_time 0

# END WordPress
```

Figura 66: documento .htaccess modificado.

El resultado es un aumento del máximo permitido a 500 MB

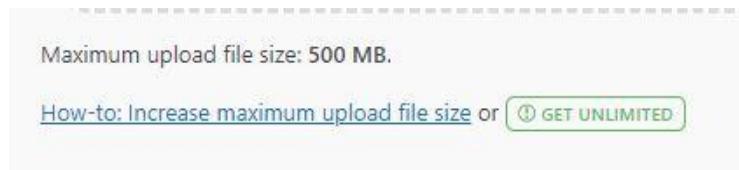


Figura 67: resultado de la modificación realizada.

8. Importacion final de la pagina

Tras esto desde la página que vimos anteriormente de el plugin All-in-One WP Migration realizaremos la importación con el archivo que se encuentra en (daw223-05/Instalacion/CopiaTienda.wpress)

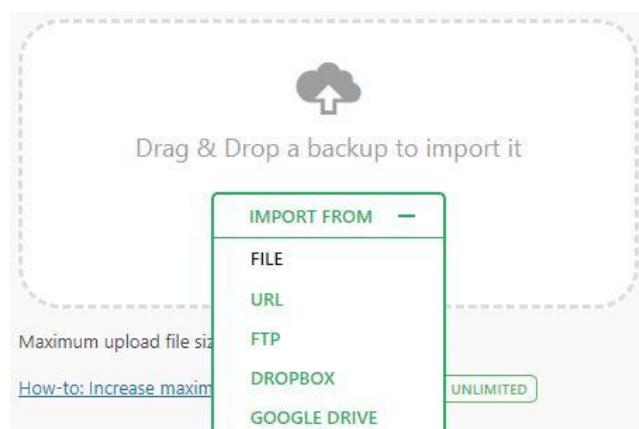


Figura 68: interfaz de importación de All-In-One

✓ hoy (1)			
 CopiaTienda.wpress	18/05/2023 19:38	Archivo WPRESS	338,272 KB
> al principio de esta semana (1)			

Figura 69: *Fichero copia de proyecto*

Tras esto se comenzará a importar y saldrá un aviso que nos advertirá que la página actual se sobreescibirá con la que estamos importando, seleccionamos proceder.

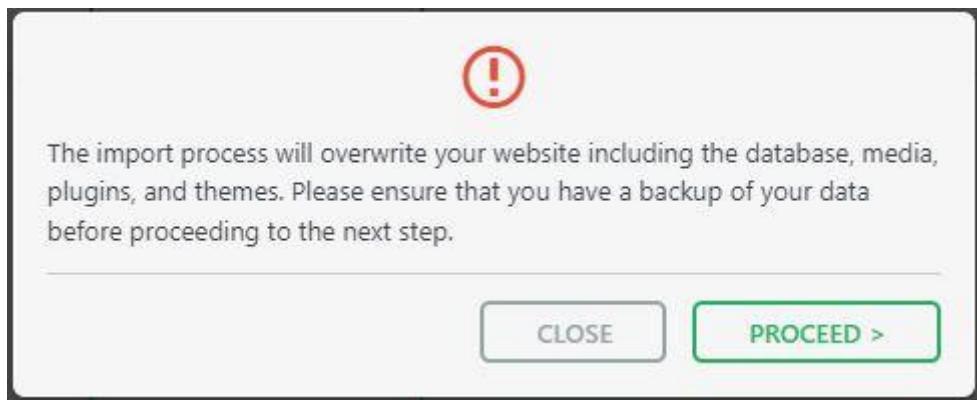


Figura 70: *advertencia plugin All-In-One*

Finalizando el proceso, logrando así una copia exacta de la web en cuestión. Si se quiere acceder al wp-admin de esta página será necesario el usuario y contraseña de la página original (usuario:allberto,contraseña:ProyectoAlbertoDaw05).

8. Bibliografía

Incluirá toda la documentación consultada: libros, apuntes, páginas web, foros...

- Aplicaciones usadas
 - Servidor XAMPP (<https://www.apachefriends.org/es/index.html>)
 - Wordpress (<https://es.Wordpress.org/download/>)
 - Gimp (<https://www.gimp.org/downloads/>)
 - Día (<http://dia-installer.de/download/index.html.en>)
 - Draw.io (<https://app.diagrams.net/>)
- Información general de los conceptos a tratar
 - Web principal de Wordpress (<https://Wordpress.com/es/>)
 - Repositorio de imágenes de docker (<https://hub.docker.com>)
 - Imagen oficial de Wordpress con indicaciones de instalación y uso de la misma (https://hub.docker.com/_/Wordpress)
- Servicios hosting investigados.
 - <https://www.hostinger.es/>
 - <https://www.ionos.es/>
 - <https://www.hostgator.com/>
 - <https://www.godaddy.com/es-es>
 - <https://www.getambassador.com/>

- <https://www.siteground.es/>
- Webs que me otorgaron documentación para ciertos apartados
 - (Despliegue)
 - <https://docs.github.com/es>
 - <https://docs.docker.com/>
 - <https://www.debian.org/doc/>
 - https://hub.docker.com/_/Wordpress
 -
 - (Código css , js y php)
 - <https://developer.mozilla.org/es/>
 - <https://www.w3schools.com/>
 - <https://www.php.net/>
 -
 - (Interfaz y desarrollo)
 - <https://creativedothemes.com/blocksy/docs/>
 - <https://www.hostinger.es/tutoriales/que-es-widget-Wordpress>
 - <https://Wordpress.com/es/support/widgets/>
 - <https://WooCommerce.com/documentation/WooCommerce/>
 -
 - (Paypal)
 - <https://developer.paypal.com/>
 - <https://WooCommerce.com/document/paypal-checkout-espanol/>
 - <https://www.hostinger.es/tutoriales/configurar-paypal-en-WooCommerce>
 -
 - (Ciertas aplicaciones para casos específicos)
 - <https://gitmind.com/es/>
 - <http://www.gimp.org.es/>
 - <http://dia-installer.de/download/index.html.en>
- Webs que me otorgaron información relevante para el proyecto
 - <https://translatepress.com/best-WooCommerce-translation-plugin/>
 - <https://ignaciosantiago.com/mejor-hosting-WooCommerce/>
- Los Tutoriales de YT que más ayuda me ofrecieron para el uso de las herramientas e investigación, en mayor o menor medida.
 - Tutorial WORDPRESS 2022 desde CERO (Parte 01)
(<https://www.youtube.com/watch?v=le8G-10DCtg>)
 - Tutorial WORDPRESS 2022 desde CERO (Parte 02)
(<https://www.youtube.com/watch?v=VJHNkpjWTVw>)
 - Tutorial WooCommerce 2021 COMPLETO - Para Principiantes
(https://www.youtube.com/watch?v=1uf_S8427us)
 - Tutorial de WOOCOMMERCE (2022) en Español [Parte 02]
(<https://www.youtube.com/watch?v=6e0-M7qLD7M>)
 - Tutorial de WOOCOMMERCE (2022) en Español [Parte 03]
(<https://www.youtube.com/watch?v=xdVdXmoy8qE>)

- WooCommerce Shipping | Weight Based Shipping | FREE Plugin (<https://www.youtube.com/watch?v=ja2TuC1dCoc>)
 - Cómo COPIAR una PÁGINA WEB en WORDPRESS (<https://www.youtube.com/watch?v=93L701ggjq0>)

- Páginas oficiales de los plugins utilizados
 - Blocksy Companion (<https://Wordpress.org/plugins/blocksy-companion/>)
 - Stackable (<https://Wordpress.org/plugins/stackable-ultimate-gutenberg-blocks/>)
 - WooCommerce (<https://Wordpress.org/plugins/WooCommerce/>)
 - WooCommerce Weight Based Shipping (<https://Wordpress.org/plugins/weight-based-shipping-for-WooCommerce/>)
 - Hide Categories and Products for WooCommerce (<https://Wordpress.org/plugins/hide-categories-products-WooCommerce/>)
 - WooCommerce PayPal Payments (<https://Wordpress.org/plugins/WooCommerce-paypal-payments/>)
 - WPC Product Bundles for WooCommerce (<https://Wordpress.org/plugins/woo-product-bundle/>)
 - WP Code (<https://Wordpress.org/plugins/insert-headers-and-footers/>)
 - All-in-One WP Migration (<https://Wordpress.org/plugins/all-in-one-wp-migration/>)

Los siguientes plugins están instalados pero su uso es reducido o inactivo.

- WooCommerce Payments (<https://Wordpress.org/plugins/WooCommerce-payments/>)
- Contact Form by WPForms (<https://Wordpress.org/plugins/wpforms-lite/>)
- WP Mail SMTP by WPForm (<https://Wordpress.org/plugins/wp-mail-smtp/>)

Los siguientes plugins no llegaron a la versión final pero sí se realizó un estudio y pruebas con estos mismos

- WordPress Slider Plugin (<https://Wordpress.org/plugins/block-slider/>)
- Product Price by Quantity for WooCommerce (<https://Wordpress.org/plugins/wholesale-pricing-WooCommerce/>)
- WPC Grouped Product for WooCommerce (<https://Wordpress.org/plugins/wpc-grouped-product/>)
- Elementor Website Builder (<https://Wordpress.org/plugins/elementor/>)
- HT Slider For Elementor (<https://Wordpress.org/plugins/ht-slider-for-elementor/>)

- Páginas usadas para la resolución de dudas durante el proyecto, la mayoría de dudas se pudieron resolver simplemente investigando en la documentación correspondiente vistas anteriormente.
 - <https://stackoverflow.com/> (Las dudas que se me resolvieron en esta web fueron pocas, por la razón que mencionaré más adelante)
 - <https://chat.openai.com/> (Me gustaría mencionar que hasta el principio del proyecto no pensé que llegaría a darle uso a ChatGPT, pero durante el

comienzo del proyecto decidí preguntar a esta ia sobre alguno de los errores que me llegaban a ocurrir, y aunque no me diera una solución exacta de cómo resolver este mismo en algunas ocasiones, conseguía indicarme las posibles causas de este, lo que me era suficiente para lograr solucionarlos.)

- <https://termify.io/generate-terms-and-conditions> (esta pagina sirvio de guia para la realización de los términos y condiciones de la web)
- La información de los productos y detalles de estos mismos, ademas de informacion de como estos quieren ser organizados fue obtenida de la dueña de los mismos (AlmaHeiko) y su tienda de Etsy
(https://www.etsy.com/uk/shop/AlmaHeiko?ref=related&listing_id=1329721350)

9. Anexos

Anexo I: Recursos de la presentación y defensa del proyecto:

- Como base de la defensa, se dará uso a una presentación PowerPoint, la cual no guiará por los puntos correspondientes a mostrar.
- Servidor clouding donde se aloja la aplicación, en la cual se realizan las operaciones correspondientes. Este será presentado desde un navegador web, para mostrar de manera más interactiva lo que se presenta en el PowerPoint anteriormente mencionado.