Escuela Politécnica Superior de Jaén

Grado en Ingeniería Informática

UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

DESARROLLO DE UNA TIENDA ONLINE DE ARTÍCULOS DE CERÁMICA MEDIANTE EL USO DE UN CMS

Alumno: Raúl Fernández Tirado

Tutor: Prof. D. José Ignacio Gómez Espínola
Dpto: Departamento de Informática

Septiembre, 2022

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don José Ignacio Gómez Espínola, tutor del Trabajo Fin de Grado titulado: Desarrollo de una tienda online de artículos de cerámica mediante el uso de un CMS, que presenta Raúl Fernández Tirado, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2022

El alumno:

FERNANDEZ TIRADO RAUL - 26259804Z
Firmado digitalmente por FERNANDEZ TIRADO RAUL - 26259804Z
Fecha: 2022.09.09 19:25:13 +02'00'

Raúl Fernández Tirado

El tutor:

GOMEZ ESPINOLA JOSE IGNACIO - 24277832K
Firmado digitalmente por GOMEZ ESPINOLA JOSE IGNACIO - 24277832K
Fecha: 2022.09.09 19:31:46 +02'00'

José Ignacio Gómez Espínola

Dedicatoria

Me gustaría comenzar dedicando este proyecto a las personas que han hecho posible que yo hoy este aquí escribiendo esto.

En principio, llegar hasta aquí no hubiera sido posible sin mis padres, que me han dado una educación, una motivación y un apoyo moral y económico para seguir día a día y no rendirme.

Continuar dedicando este proyecto a familiares como mi tío que sin tener yo unas nociones avanzadas de programación me ayudo a formarme en lo que fue un gran paso para mí en el desarrollo web, enseñándome PHP desde cero, lo que supuso luego poder encontrar trabajo de forma más sencilla y poder costearme una gran parte de mis estudios.

Por último, dedicarle este trabajo a la persona que me aguanta, me motiva, me ayuda en cada momento para que estos últimos y duros años no me viniera abajo ni me rindiese, a mi pareja Raquel.

Índice

CAPITULO 1. INTRODUCCIÓN 9

1.1 Motivación 9

1.2 Objetivos 9

CAPITULO 2. ESTADO DEL ARTE 11

2.1 Introducción a los CMS 11

2.2 Definición 12

2.3 Características de un CMS 12

2.4 Ventajas de un CMS 13

2.5 Inconvenientes de un CMS 13

2.6 CMS más populares 14

2.7 Elección y justificación 17

2.8 Análisis requerimientos 18

2.9 Análisis de hosting 22

CAPITULO 3. INGENIERÍA DEL SOFTWARE 27

3.1 Introducción 27

3.2 Metodología de la ingeniería del software 27

3.2.1 Definición 27
3.2.2 Metodologías 27
3.2.3 Elección de la metodología 29

3.3 Análisis y diseño 30

3.3.1 Definición de requisitos 30
3.3.2 Requisitos funcionales 30
3.3.3 Requisitos no funcionales 33
3.3.4 Planificación 34
3.3.5 Diagramas de clases 38
3.3.6 Casos de uso 39
3.3.7 Diagramas de secuencias 40

3.4 Instalación 62

3.5 Implementación y desarrollo 75

3.6 Pruebas 83

CAPITULO 4. CONCLUSIÓN 92

4.1 Conclusiones 92

4.2 Posibles desarrollos futuros 93

ANEXOS 94

Información de acceso y pruebas 94

Manual de usuario 95

Software utilizado 103

BIBLIOGRAFÍA 106

Índice de Figuras

Figura 1. Logo "The nu concept" 9
Figura 2. Historia del CMS 11
Figura 3: Imágen CMS 12
Figura 4. Top eCommerce technologies based on market share in 2022 14
Figura 5. Tabla con los CMS eCommerce Top en 2022 14
Figura 6. Magento Open Source eCommerce 18
Figura 7. Datos certificado SSL 21
Figura 8. Características certificado SSL elegido 21
Figura 9. Creación registro CNAME para validar el certificado 22
Figura 10. PiensaSolutions logo 26
Figura 11. Vista de configuración del servidor por PiensaSolutions 26
Figura 12. Esquema metodología ágil 28
Figura 13. Esquema metodología iterativa 28
Figura 14. Esquema metodología en cascada 29
Figura 15. Diagrama de Gantt 37
Figura 16. Casos de uso Usuario 39
Figura 17. Casos de uso Usuario administrador 39
Figura 18. Diagrama de secuencia, navegar por la tienda 40
Figura 19. Diagrama de secuencias, consultar categorías 41
Figura 20. Diagrama de secuencias, consultar detalles producto 42
Figura 21. Diagrama de secuencia, búsqueda de productos 43
Figura 22. Diagrama de secuencia, envío de review 44
Figura 23. Diagrama de secuencia, cambiar idioma 45
Figura 24. Diagrama de secuencia, añadir productos al carro 46
Figura 25. Diagrama de secuencia, registro de usuario 47
Figura 26. Diagrama de secuencia, login 48
Figura 27. Diagrama secuencia, modificar información usuario 49
Figura 28. Diagrama secuencia, cerrar sesión 50
Figura 29. Diagrama secuencia, acceder historial pedidos 51
Figura 30. Diagrama secuencia, añadir producto lista deseos 52
Figura 31. Diagrama secuencia, acceder lista deseos 53
Figura 32. Diagrama de secuencia, inicio de sesión administrador 54
Figura 33. Diagrama de secuencia, cerrar sesión administrador 55
Figura 34. Diagrama de secuencia, administrar categorías 56
Figura 35. Datos de acceso al servidor 62
Figura 36. Comando de acceso al servidor 62
Figura 37. Comando actualización del servidor 62
Figura 38. Comando instalación Apache2 62
Figura 39. Comando instalación MySQL Server 62
Figura 40. Comprobación estado Apache2 63
Figura 41. Comprobación del estado de MySQL 63
Figura 42. Comando instalación paquete para gestionar repositorios 63
Figura 43. Comando para añadir repositorio personal (PPA) 63
Figura 44. Comando para instalar PHP 7.4 y extensiones 63
Figura 45. Comprobación de la versión de PHP 64
Figura 46. Configuración registro A del dominio 64
Figura 47. Ficheros del certificado en el servidor 65
Figura 48. Comando activación módulo SSL 65

Figura 49. Comando recargar el servidor Apache2 65
Figura 50. Fichero de configuración del certificado en el servidor 66
Figura 51. Comando activación host virtual SSL 66
Figura 52. Comando reinicio servidor Apache2 67
Figura 53. Comprobación de instalación de certificado 67
Figura 54. Evaluación del test SSL de SSL Labs..... 67
Figura 55. Detalles del certificado instalado, SSL Labs 68
Figura 56. Comando para obtener el paquete de Elasticsearch 68
Figura 57. Comando instalación Elasticsearch 68
Figura 58. Comando inicio y mostrar estado del servicio de Elasticsearch 68
Figura 59. Comando acceso a la consola de la base de datos..... 69
Figura 60. Órdenes para iniciar la configuración de la base de datos 69
Figura 61. Comando creación de directorios base..... 70
Figura 62. Comando para clonar el repositorio de Magento Open Source eCommerce 2.4.0 70
Figura 63. Comando para añadir un usuario..... 70
Figura 64. Comando para añadir el usuario al grupo de Apache..... 70
Figura 65. Comando para dar permisos a las carpetas de static, media y etc . 70
Figura 66. Comando para dar permisos a las carpetas de static, media y etc . 70
Figura 67. Comando para cambiar el propietario del directorio 70
Figura 68. Comando para dar permisos de ejecución a bin/magento..... 70
Figura 69. Comando instalación de Composer 70
Figura 70. Comando instalación Magento 2.4.0..... 71
Figura 71. Inicio de proceso de instalación de Magento 2.4.0 71
Figura 72. Fin instalación Magento 2.4.0 71
Figura 73. Comando para asignar propietario www-data al directorio base 72
Figura 74. Página por defecto de Magento 72
Figura 75. Página por defecto del panel de administración 1 73
Figura 76. Página por defecto del panel de administración 2 73
Figura 77. Instalación paquete español Magento 2 75
Figura 78. Cambio de idioma de la tienda online 75
Figura 79. Cambio de idioma del panel de administración 76
Figura 80. Ajuste inicial de Magento 1 76
Figura 81. Ajuste inicial de Magento 2 76
Figura 82. Ajuste inicial de Magento 3 77
Figura 83. Pre-instalación tema Breeze Evolution 77
Figura 84. Instalación tema Breeze Evolution..... 77
Figura 85. Aplicación del tema 77
Figura 86. Estilo CSS de la imagen de fondo en la página de inicio..... 78
Figura 87. Estilo CSS bug ancho de la imagen de fondo..... 78
Figura 88. Modificación de estilo CSS en la imagen de fondo..... 79
Figura 89. Página inicio con imagen arreglada 79
Figura 90. Inserción código footer..... 81
Figura 91. Footer personalizado 81
Figura 92. Diseño página de error 404..... 82
Figura 93. Prueba, añadir producto al carrito..... 83
Figura 94. Prueba, tramitar pedido desde el carrito 83
Figura 95. Prueba, formulario datos de dirección de envío..... 84
Figura 96. Prueba, resumen del pedido 85
Figura 97. Prueba, pago PayPal 1 85

Figura 98. Prueba, pago PayPal 2 86
Figura 99. Prueba, comentario confirmación compra 86
Figura 100. Prueba, impreso compra 87
Figura 101. Prueba, comprobación pedida. 87
Figura 102. Prueba, dimensiones PC 88
Figura 103. Prueba, dimensión smartphone sin menú 89
Figura 104. Prueba, dimensión smartphone con menu desplegado 89
Figura 105. Prueba, página con idioma español 91
Figura 106. Prueba, página con idioma inglés 91
Figura 107. Manual, página principal 95
Figura 108. Manual, cabecera y categorías 95
Figura 109. Manual, cabecera cuenta 95
Figura 110. Manual, pantalla login 96
Figura 111. Manual, pantalla de registro 97
Figura 112. Manual, información de contacto 98
Figura 113. Manual, formulario de contacto 98
Figura 114. Manual, búsqueda avanzada 99
Figura 115. Manual, formulario de pedidos y devoluciones 100
Figura 116. Manual, pantalla "sobre mi" 101
Figura 117. Manual, ilustraciones del lugar de trabajo 101
Figura 118. Manual, vista productos y categorías 102
Figura 119. Manual, producto al detalle 102
Figura 120. Manual, pantalla de "Mi cuenta" 103
Figura 121. Terminal MacBook conectado al servidor 103
Figura 122. FileZilla software 104
Figura 123. Visual Studio Code 104

Índice de Tablas

Tabla 1. Análisis de precios para hostings VPS.....23
Tabla 2. Análisis de recursos para hostings VPS 24
Tabla 3. Planificación, duración del proyecto.....36
Tabla 4. Planificación, presupuesto del proyecto.....37

CAPITULO 1. INTRODUCCIÓN

1.1 Motivación

Empecé en el mundo de la informática gracias al desarrollo web mediante CMS, es curioso que termine haciendo este trabajo de fin de grado sobre la misma temática que me hizo ser curioso en este tema. Empecé como afición por crear páginas webs en servidores gratuitos y con dominios gratuitos y al no saber de programación, comencé a cambiar código para hacer las páginas más personalizables en CMS como Drupal, Joomla o PHPNuke, esto me hizo pensar en que yo quería saber hacer esas páginas desde cero, entender todo el código y poder crear algo más que una simple página web, esta fue una de las razones que me hizo elegir este grado y este trabajo profesional para mi futuro.

Llevo 2 años trabajando como programador PHP desarrollando aplicaciones web con frameworks, pero creo que actualmente lo más rápido y eficiente para un negocio online es realizar una aplicación web a través de un CMS, ya que te viene preparado para lo que necesitas y te permite personalizarlo dentro de unos límites.

1.2 Objetivos

El objetivo principal de este trabajo es realizar una tienda online mediante un CMS, para ello hemos contactado con una cliente llamada Cristina Núñez que tiene una marca llamada “The nu concept” para vender sus productos de cerámica. La página en cuestión será https://www.thenuconcept.com

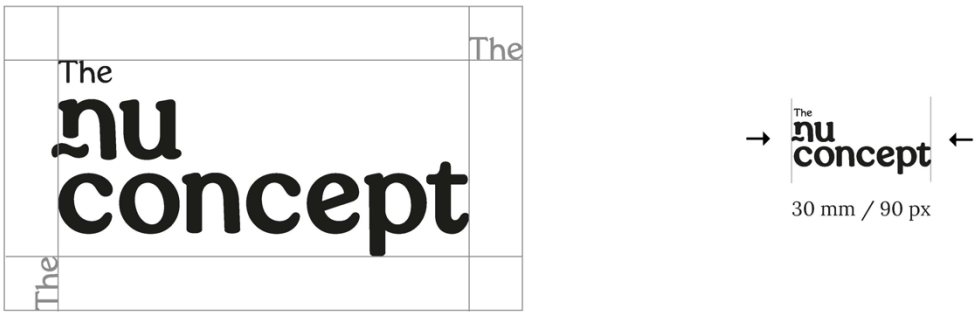


Figura 1. Logo "The nu concept"

Hasta ahora ha utilizado la red social Instagram para comerciar sus productos y publicitarse, por eso es ideal que nuestra cliente tenga un sitio web donde pueda vender sus productos y su marca de una manera más eficiente y llevando el control de todo su negocio.

Para conseguir nuestro objetivo principal deberemos fijar unos objetivos a tener en cuenta en el desarrollo del proyecto:

- Estudiaremos las distintas opciones de CMS que hay actualmente y justificaremos la elección obtenida.
- Analizaremos los servicios de hostings VPS actuales y un certificado SSL necesario para la instalación del proyecto y justificaremos las opciones escogidas.
- Instalación del CMS en el servicio hosting y configuración del servidor.
- Desarrollo de la configuración del sitio, aplicación de un diseño, personalización de la aplicación web e inserción de productos y categorías, se aplicará un diseño “responsive”, es decir, un diseño adaptativo a distintas resoluciones y dispositivos.
- Desarrollo de pruebas con métodos de pago aplicados

En definitiva, nuestro objetivo es el desarrollo de una tienda online de artículos de cerámica mediante el uso de un CMS y lanzarla en internet con las herramientas necesarias.

CAPITULO 2. ESTADO DEL ARTE

2.1 Introducción a los CMS

En la actualidad es muy importante la Ciencia de la información, todo es información, uno de los factores por los que nació esta ciencia es el interés por las Tecnologías de la información y la aparición de internet. De hecho, si nos remontamos al inicio del internet podemos comprobar el nacimiento del CMS, por Tim Berners-Lee en 1990, que se inspiró en un sistema de hipertexto basado en Internet HTML, que representaba solo texto y enlaces.

Figura 2. Historia del CMS

Hasta mediados de los 90, la red World Wide Web (Internet) aumenta su popularidad, esto hace que el número de sitios web en la red aumente y a su vez necesitan una actualización para optimizar estos sitios. Gracias a esto comenzaron a salir al mercado los primeros CMS, como FileNet, StoryBuilder de Vignette, Documentum y muchos otros, sin embargo, estos eran de código privado y patentados.

No fue hasta la década de los 2000 cuando empezamos a ver los primeros CMS de código abierto y gratuitos con servicios de plugin extensibles y varias plantillas de HTML y CSS disponibles para cambiar el diseño del sitio, estos primeros CMS fueron algunos como WordPress, Drupal, Joomla o PHPNuke. Estos CMS ya implementaban conexiones a una base de datos como MySQL o MariaDB.

La aparición de los CMS de código abierto inspiró a la creación de una infraestructura LAMP (Linux, Apache, MySQL/MariaDB y PHP/Perl/Python). Esta nueva estructura representó el inicio del desarrollo web que permitió la creación de sitios web dinámicos que utilizan consultas a la base de datos para ofrecer contenidos únicos a los diferentes usuarios finales. En este momento, el modelo anterior de sitios estáticos ubicados en un servidor empezó a desaparecer.

¿Y por qué los CMS adquirieron esta gran popularidad? A parte de por las ventajas que tiene el uso de estos, que las analizaremos más adelante, es porque a día de hoy el manejo y el análisis de la información es muy importante en un negocio y no todo el mundo tiene conocimientos de programación para desarrollar un sitio web desde cero o costearse un programador, por lo que un CMS aun que necesita unos conocimientos mínimos para la instalación, la gestión de este y el mantenimiento lo puede hacer cualquier persona y por eso

a día de hoy la mayoría de las páginas webs de comercios están creadas bajo un CMS. (10) (4)

2.2 Definición

CMS son las siglas de Content Management System, si traducimos las siglas podemos decir que un CMS es un Sistema de Gestión de Contenidos, explicado de mejor forma es un programa informático que permite crear una aplicación web completa sin necesidad de codificar desde cero todo el proyecto. Un CMS te permite gestionar un sitio web de forma ágil ya que no sirve solo parar crearlo si no que a través del CMS y conectado con una base de datos se introducirá nueva información y se actualizará la existente de forma más fácil y rápida.

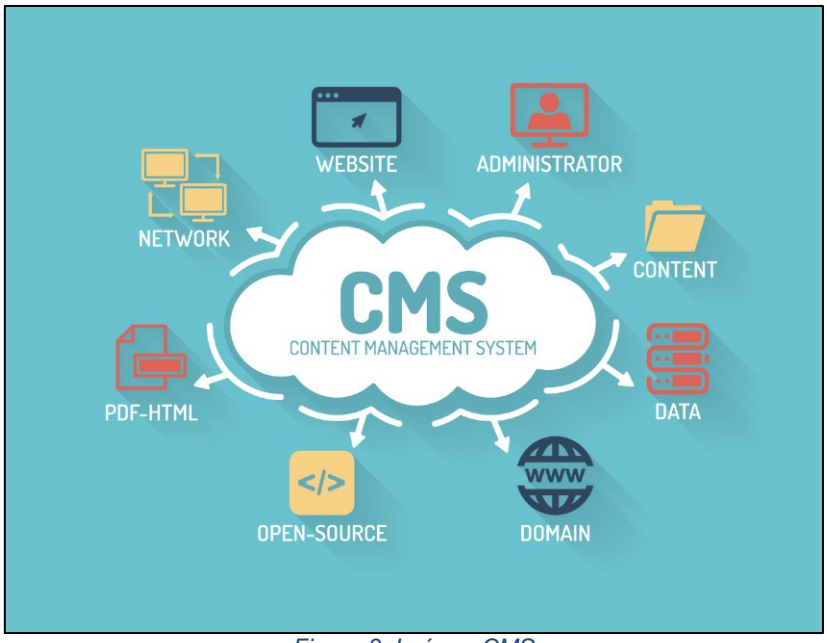


Figura 3: Imágen CMS

Actualmente son muy utilizados, ya que como hemos dicho anteriormente permite mantener el contenido de una manera ágil y a la vez permite que cualquier persona pueda realizar tareas de mantenimiento y actualización del sitio sin tener grandes conocimientos técnicos. (1) (2)

2.3 Características de un CMS

Debido a que el uso de un CMS es muy popular actualmente vamos a explicar sus características principales:

- Separan la funcionalidad y código del sitio web de la gestión de contenidos y usuarios del sitio, esto hace que la persona que gestión el sitio no necesite grandes conocimientos de informática una vez ya instalado el CMS.
- Tienen una gran variedad de diseños y son fáciles de cambiar a través el sistema de administración del CMS.

- Control de versiones, los CMS nos ofrecen la posibilidad de versionar el contenido para poder recuperar información de una versión en un momento concreto.
- Multi-Idioma, los CMS te dan la posibilidad de poner el sitio en varios idiomas para hacer un portal más internacional, además pueden funcionar en distintas monedas para los CMS eCommerce.
- Normalmente cada CMS dispone de recursos y extensiones, algunas de pago y otras gratuitas, para mejorar tu sitio web o enfocarlo a alguna especialidad concreta. (3)

2.4 Ventajas de un CMS

- **Económicos en su instalación**, los CMS más populares suelen ser gratuitos o tener una versión gratuita, aun que puedes mejorarlos con versiones de pago, extras de pago o incluso diseños específicos de pago para tu sitio web.
- **Edición del contenido ágil**, es fácil, cómodo de actualizar y de añadir información nueva ya que incorporan un panel de administración donde gestionar todo el sitio web, además la información es independiente del diseño, por lo que al cambiar el diseño no afectará al contenido del sitio.
- **SEO**, los CMS son amigables con el SEO, ya que están ideados para aplicaciones web de negocios como tiendas online o simplemente una web informativa para una empresa, por lo que son fácil de posicionar incluyendo títulos de página personalizados, así como metadatos y URL ajustables.
- **Desarrollo y tiempo de implementación rápidos**, debido a que la instalación de un CMS ya viene con muchas funciones preinstaladas como una gestión de usuarios, panel de administración, etc. Lo que conlleva también a un fácil mantenimiento. (3)

2.5 Inconvenientes de un CMS

- **Recursos del servidor**, algunos CMS al ser demasiado complejos y estar preparados para proyectos con distinta temática requieren muchos recursos del servidor lo que con un incremento de peticiones podría producir la caída del servidor.
- **No es del todo personalizable**, es decir, puedes aplicar plantillas y diseños específicos, pero pueden no ser del todo ajustables a un proyecto, lo mismo pasaría con extras o la misma gestión del CMS, todo esto podría modificarse a través del código, pero hacer esto induce a vulnerabilidades e incompatibilidades con actualizaciones del CMS.
- **Costosa implementación**, lo mismo que son económicos en su instalación mejorarlos según que CMS puede ser costoso ya que necesitan plugins o diseños que requerirán de un coste extra. (3)

2.6 CMS más populares

Dentro de la gran cantidad de CMS que hay hoy en la actualidad hemos hecho un estudio y análisis sobre los mejores enfocados al comercio digital (eCommerce), de esta forma podremos elegir el mejor y más adecuado para nuestro proyecto. (6) (7)

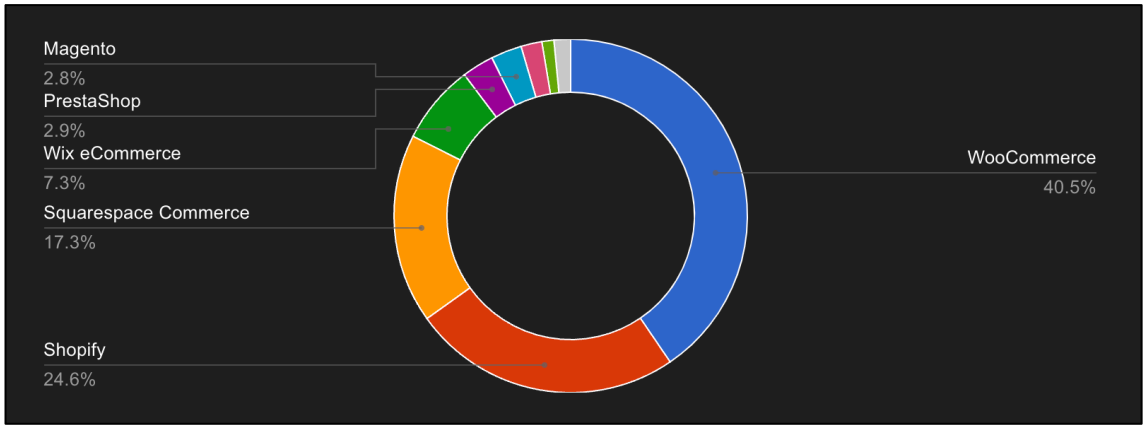


Figura 4. Top eCommerce technologies based on market share in 2022

#	Technology	Websites tracked	Market share
1	WooCommerce	1.490.000	40%
2	Shopify	935.000	25%
3	Squarespace Commerce	640.000	17%
4	Wix eCommerce	275.000	7%
5	PrestaShop	110.000	3%
6	Magento	107.000	3%
7	1C-Bitrix	71.000	2%
8	BigCommerce	40.500	1%
9	OpenCart	28.700	0.8%
10	Square Online	27.100	0.7%

Figura 5. Tabla con los CMS eCommerce Top en 2022

- **WooCommerce (Wordpress)**

  WooCommerce es un plugin gratuito creado por la empresa WooThemes que permite convertir una instalación de WordPress en una tienda online, por lo que el CMS utilizado en este caso es Wordpress.

  Wordpress es un CMS lanzado en mayo de 2003, enfocado en la creación de cualquier tipo de página web, al principio tuvo una gran popularidad por la creación de blogs para luego más adelante convertirse en la principal herramienta para la creación de páginas webs comerciales.

WooCommerce tiene las ventajas de ser gratuito, flexible, y sencillo, muy customizable sabiendo CSS y con la mayor comunidad para un CMS ya que es el más utilizado.

En cuanto a la parte negativa, la versión gratuita es muy básica y si quieres avanzar o mejorar tu sitio ya tienes que añadir plugins o diseños de pago que hace que el costo del sitio aumente. Y el otro punto negativo es que no es el mejor CMS para manejar una tienda con muchas referencias ya que no es la que más rápido gestiona una tienda con una gran cantidad de artículos.

- **Shopify**

Este CMS fue creado en 2004, aunque la puesta en marcha de la plataforma no llegaría hasta 2006, es el segundo más popular después de Wordpress y está más extendido en países como EEUU. Shopify es un CMS creado con la intención de que una persona sin nociones de programación sea capaz de diseñar su tienda online a su gusto, debido a que es de lo más intuitivo.

Es diferente a otros CMS ya que este no es gratuito y funciona como un SAAS (Software as service) al que te conectas sin necesidad de descargar ningún programa por lo que no es necesario la contratación de hosting ya que cuenta con distintos planes y ofertas para cubrir las necesidades tanto de pequeñas empresas y autónomos como de grandes empresas. Es una plataforma enfocada totalmente al eCommerce que permite vender productos físicos y virtuales.

Para un usuario sin conocimientos de programación sería una plataforma ideal ya que es del todo sencillo en su creación y con una fácil gestión de tu sitio, tiene el hosting incluido en el plan. Y como se ha comentado anteriormente está totalmente enfocado al comercio electrónico por lo que tiene facilidad de pagos con aceptación de hasta 70 monedas diferentes, sistema de impuestos automático por país, productos ilimitados y muchas más facilidades que van variando según el plan que se contrate.

Por el contrario, no es un CMS de código abierto ya que se gestiona a través de su web y si la persona que lo gestiona tiene nociones de programación no podrá modificar el sitio a su gusto. No es gratuito ya que dispone de distintos planes con distintas características.

- **Magento**

Magento es un CMS que publicó su primera versión de 2008, esta creado en PHP y tiene dos versiones, una de pago más completa sostenida por Adobe desde que está adquirió Magento en en 2018, aunque antes también se ofrecía una versión de pago similar, y otra versión open source de código abierto la cual puedes personalizar. Magento utiliza la

programación orientada a objetos y el modelo vista controlador de forma similar a los frameworks que existen actualmente para el desarrollo web.

Magento, además de ser un CMS muy completo, posee una gran escalabilidad y una personalización total, su versión de código abierto está disponible desde Github, tiene una gran comunidad al ser uno de los CMS más populares, además de que es personalizable por medio de diseños específicos y extensiones que se le pueden añadir a la versión de pago y a la versión gratuita.

Por el contrario, Magento es un CMS complicado, no ofrece un servicio de hosting, por lo que deberás proveer tú mismo este servicio además de la contratación de un dominio y herramientas adicionales como de un certificado SSL. En definitiva, es un CMS que requiere de conocimientos más avanzados, no solo para crearlo si no también para su personalización y mejora.

- **Wix eCommerce**

  La plataforma Wix se fundó en 2006 con el fin de la creación de páginas web de todo tipo de forma similar a Shopify, es decir, a través de su propia página web por medio de una plantilla. Wix eCommerce es un servicio que ofrece Wix para poder crear tu tienda online de forma intuitiva y fácil sin necesidad de tener conocimientos de programación. Los servicios de Wix son de pago y por un precio fijado mensualmente tienes a tu disposición un paquete de servicios completo que te permite crear y gestionar tu sitio web de comercio electrónico sin problemas.

  Los puntos fuertes de Wix eCommerce son la facilidad que tiene para crear un sitio web para una tienda online en muy poco tiempo, ofrece dos posibilidades para crear tu sitio, mediante un editor tradicional o por un editor automatizado Wix ADI (Artificial Design Intelligence), con el que tardará muy poco en tenerlo. Esto lo hace ser muy intuitivo y cómodo, y al tener que hacerlo todo a través de su propia página no tienes que preocuparte de servicios de hosting y otros servicios.

  Por parte negativa, es muy similar a Shopify, y es que no es un CMS de código abierto ni instalable, que esto puede ser una ventaja para una persona sin conocimientos avanzados de programación y una limitación para quien si los tiene. No es gratuito ni ofrece una versión gratuita, por lo que esto aumenta el coste de nuestro sitio.

- **Prestashop**

  Prestashop, lanzado en 2008, es un sistema de gestión de contenidos creado exclusivamente para la creación de tiendas online que se ha logrado posicionar como una de las mejores y de las plataformas más usadas. Es un CMS libre y de código abierto, que como Magento, puedes encontrar en GitHub, utiliza PHP y Smarty, a partir de 2016 añade a su código fuente el proyecto Symfony, un framework muy conocido de PHP.

Como positivo podemos destacar la facilidad de uso que tiene y la gran personalización con miles de diseños disponibles debido a la popularidad que tiene. Es un CMS que combina el poder de Magento con la facilidad que tiene Shopify, y al igual que estos tiene una gran comunidad por lo que permite que el proyecto sea estable a largo plazo.

Si tenemos que sacar puntos negativos podemos decir que no es tan flexible para realizar modificaciones avanzadas y requerirás de conocimientos avanzados para realizarlas. En tema SEO es mejorable ya que el modo en el que esta implementado no facilita al posicionamiento de tu tienda online. Y aunque sea gratuito para mejorar tu tienda tienes a disposición muchos módulos, pero estos son de pago, por lo que aumentaría el coste de tu sitio.

- **Squarespace Commerce**

  Squarespace nació como empresa en 2003, comenzó como un sitio de alojamiento de blogs pero poco a poco fue avanzo ofreciendo otros servicios. Squarespace te ofrece la posibilidad de crear el sitio y a través del plugin Squarespace Commerce puedes convertir el sitio en un negocio online.

  Squarespace, con las funciones de Commerce, dispone de un plan básico y un plan avanzado con más funciones y con un costo mayor. Como ocurre con Shopify, la propia plataforma te ofrece su hosting y también puedes tramitar a través de ellos servicios como dominio y certificado SSL.

  Por lo tanto, podemos apuntar que a la hora de tramitar tu tienda online Squarespace es de lo más cómodo, sencillo e intuitivo, ha crecido mucho en el negocio electrónico hasta el punto de estar entre los más populares y tener proyectos ambiciosos.

  Al ser de código cerrado tiene su lado positivo al ser más seguro y estable, pero esto también hace que no sea de lo más personalizable y que una persona con conocimientos avanzados en programación pueda modificarlo, aunque por medio de CSS se puede personalizar dentro de unos límites. (8)

2.7 Elección y justificación

Después de saber que es un CMS, y analizar los más populares, ¿Cuál elegiríamos? Para esta respuesta no hay una regla definitiva, ya que depende de muchos factores y necesidades.

Primero necesitamos priorizar una elección de bajo coste ya que nuestra clienta posee una tienda pequeña que quiere ir creciendo poco a poco, por lo que destacamos opciones que ofrezcan una versión gratuita que no sea demasiado limitable a mejorar la tienda a corto plazo.

También vamos a priorizar las opciones que para instalación y desarrollo requieran a una persona con conocimientos avanzados de programación, por lo que descartamos Shopify, Squarespace Commerce y Wix eCommerce.

Llegados a este punto aun que Wordpress con WooCommerce es más utilizado y tiene un gran potencial al igual que Prestashop, vamos a elegir como CMS a Magento, ya que esta desarrollado en PHP, que es un lenguaje conocido por el desarrollador, utilizaremos la versión de código abierto **Magento Open Source eCommerce 2.4.0**, ya que eso nos reducirá el coste del proyecto, Magento dispone de una gran comunidad y es más personalizable que otros CMS, por lo que a largo plazo será más fácil mejorar el sitio web.

Figura 6. Magento Open Source eCommerce

2.8 Análisis requerimientos

Después de haber analizado los CMS disponibles previamente y haber elegido Magento, este tiene unos requerimientos que deberemos cumplir para tener un buen rendimiento en nuestro servidor. (9)

Requerimientos Hardware

- 2GB de RAM, aunque se recomiendan 4GB de RAM.
- Un procesador de al menos 2 núcleos.
- Al menos 50GB de disco duro sólido (SSD)

Requerimientos Software

- Distribución Linux x86-64
- Servidor web Apache 2.4.
- PHP 7.4, para la instalación de Magento Open Source 2.4.0 necesitaremos las siguientes extensiones de PHP:
  ext-bcmath, ext-ctype, ext-curl, ext-dom, ext-fileinfo, ext-gd, ext-hash, ext-iconv, ext-intl, ext-json, ext-libxml, ext-mbstring, ext-openssl, ext-pcre, ext-pdo_mysql, ext-simplexml, ext-soap, ext-sockets, ext-sodium, ext-tokenizer, ext-xmlwriter, ext-xsl, ext-zip, lib-libxml, lib-openssl

  Más adelante en el proceso de instalación veremos como instalar todo.
- MySQL 5.7 o MariaDB 10.2, como gestor de bases de datos.
- Composer 2

- Certificado SSL, vamos a necesitar un certificado para encriptar la información y proporcionar seguridad a los clientes que accedan al sitio. Para ello hemos investigado

Al margen del servicio hosting, tenemos que analizar otro **requerimiento** como es el **certificado SSL**, este cubrirá el dominio, el cual es proporcionado por nuestro cliente que lo tenía anteriormente contratado con IONOS y por medio de una configuración DNS que más adelante veremos lo enlazaremos con nuestro hosting, por lo tanto, tenemos que analizar que es un certificado SSL y cuál sería el adecuado para nuestro proyecto.

Para empezar, que es un **certificado SSL** y por qué es necesario, Un certificado es un tipo de documento digital firmado por una entidad llamada Autoridad de certificación como puede ser Digicert, Sectigo, etc… que asocia una clave pública a unos datos que representan una organización o un dominio individual que posee la clave privada asociada a dicha clave pública.

Y es tan importante porque es el encargado de cifrar y así proteger el intercambio de información entre el servidor y el usuario, esta información puede ser los datos de su tarjeta de crédito, contraseña del usuario o algún otro dato de carácter sensible. Además, el tener un certificado para el sitio web aumenta la confianza del usuario en el sitio, mejoran el posicionamiento de la página. En caso de no tener un certificado SSL, el navegador web avisará al usuario de que esta en un sitio no confiable y que sus datos pueden no estar seguros, por lo que esto influiría negativamente en nuestro negocio online. (14) (15)

Hay distintos tipos de certificados SSL según la clasificación que tengan:

- **Según su validación**:
    - **SSL EV**, un certificado de este tipo se diferencia en que la empresa que expide el certificado tiene que realizar un proceso de inspección de los datos de la empresa que solicita el certificado, ya que es un certificado de Validación Extendida creado para garantizar que el certificado y el dominio están asociados a una empresa que está legalmente dada de alta.
    - **SSL OV**, en este tipo de certificados se comprueba el derecho del solicitante a usar un nombre de dominio específico. No se inspeccionan los datos de la empresa y únicamente se muestra la información encriptada al hacer clic sobre el Sello de Página Segura.
    - **SSL DV**, este certificado se considera como un certificado estándar, ya que no se asocia con ninguna organización o empresa, simplemente se comprueba el derecho del solicitante a usar el nombre del dominio en cuestión.
- **Según el número de dominios**:

- **Un solo dominio**, esta sería la opción más simple y más económica, ya que el certificado solo cubre un solo dominio, muy útil cuando queremos tener un certificado de forma rápida y sencilla, siempre que no sea EV y tenga que pasar por un proceso de inspección.

- **SAN (Múltiples dominios),** los certificados SAN (Subject Alternative Name) te permiten proteger varios dominios bajo el mismo certificado, por lo que si tienes varios proyectos con distintos dominios y solo quieres utilizar un certificado elegir un certificado SAN sería lo ideal ya que te ayudaría a reducir costes y tener una mejor organización y gestión de tus proyectos.

- **Wildcard**, estos certificados se utilizan para proteger subdominios ilimitados que provienen de un dominio principal, por ejemplo tenemos el dominio “thenuconcept.com”, y tendríamos un servicio de API con el dominio “api.thenuconcept.com”, con un certificado estándar no protegería este segundo dominio, por lo que lo ideal sería tener un certificado Wildcard que proteja todos los subdominios sin la necesidad de tener un certificado por cada subdominio.

Analizando lo que es un certificado SSL y los tipos que hay según su clasificación podemos concretar cual sería nuestra necesidad y que tipo de certificado necesitamos.

Puesto que para nuestro proyecto no tenemos servicios secundarios ni más dominios de otros proyectos que securizar necesitamos un dominio que cubra un solo dominio, además la tienda de nuestra cliente nace desde cero y al principio tendrá menos solicitudes, por lo tanto no necesitamos un certificado que valide una organización, ya que en lo que se diferencian los certificados EV, OV y DV no es en el cifrado si no en la garantía, la garantía en los certificados SSL es la cobertura que ofrece el emisor del certificado para sus clientes en caso de sufrir pérdidas económicas como consecuencia directa de confiar en un certificado emitido por negligencia suya. Esta garantía también varía según el proveedor que emita el certificado, aun que suelen ser parecidas según el tipo de certificado que se emita.

Existen diversas marcas emisoras de certificados SSL, algunas de nombre como DigiCert, que posee las marcas de RapidSSL, Thawte y Symantec, y otras marcas como Sectigo y Comodo que se unieron para vender sus productos, sin embargo nosotros hemos optado por darle la oportunidad a una empresa española conocida como “DonDominio”, más popular por ser una empresa de registro de dominios, pero que también tiene servicios de emisión de certificados SSL emitidos bajo el nombre de una empresa internacional con más experiencia como es Comodo, puesto que son muy económicos y tienen unas características similares a otras marcas nos hemos decidido por emitir el certificado SSL con ellos.

Figura 7. Datos certificado SSL

Figura 8. Características certificado SSL elegido

Para la validación de este certificado utilizamos una validación por DNS, ya que ya teníamos nuestro dominio enlazado con nuestro servidor (más adelante se mostrará este paso), para realizar esto había que crear un registro CNAME en el DNS del dominio dentro del panel de IONOS para validar el certificado sobre el dominio proporcionado por nuestra cliente.

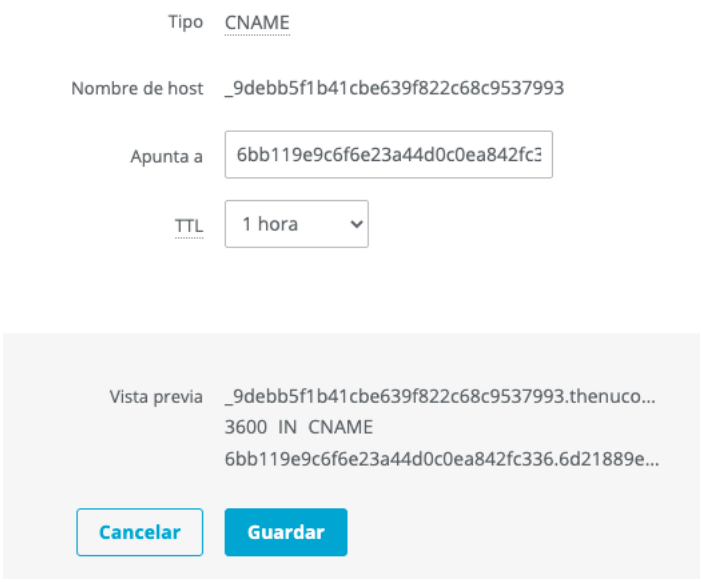


Figura 9. Creación registro CNAME para validar el certificado

Una vez propagado el registro CNAME, el certificado es validado automáticamente por la entidad emisora, en este caso aun que el dominio lo hemos sacado de DonDominio, el proveedor de este es Comodo. Con esto ya tendríamos el certificado validado y listo para descargarlo e instalarlo en nuestro servidor como veremos más adelante en el apartado de instalación.

2.9 Análisis de hosting

Lo más importante para comenzar nuestro proyecto es tener un hosting, un servidor donde instalar y alojar nuestro CMS. Nuestro hosting no puede ser un servicio normal porque necesitamos una instalación previa de unos servicios y extensiones requeridos para instalar **Magento Open Source eCommerce 2.4.0**, por lo que nuestro servidor debe ser un VPS, es decir un servidor virtual privado el cual nosotros dispongamos de todo el control para poder configurarlo como requiere nuestro CMS.

Para elegir un **servicio hosting** que cumpla estos requerimientos hemos realizado un análisis de los servicios actuales de distintos proveedores para elegir el más adecuado que se adapte a nuestro presupuesto, para ello mostramos el resultado de este análisis en la siguiente tabla. (11)

Servicio	Plan	Precio anual*	Precio mensual*	Observaciones
IONOS	VPS M	72,60	6,05 €	El primer año puede salir por 50,82 €, ya que tiene una promoción de que los primeros 6 meses se paga 2€/mes + IVA
Hostinger	VPS Server Plan 2	86,97 €		El primer año te rebajan un 54%, a partir del primer año se pasa a pagar 188,61€ anuales (IVA incluido)
PiensaSolutions	VPS BUSINESS	72,06 €	6 €	Después del primer año se paga el precio normal del servicio que serían 145,2 € anuales (IVA incluido)
Raiola Networks	VPS SSD 2	263,53 €	21,96 €	-
ProfesionalHosting	VPS Hosting SSD	274,66 €	22,88 €	-

Tabla 1. Análisis de precios para hostings VPS

* Los precios que se muestran son anuales y el precio mensual es si se pagase de forma anual, esto puede variar según el plan que se escoja, ya que tienen distintos descuentos dependiendo de la forma que se pague.

** Los precios, tanto anual como mensual, llevan el IVA incluido.

Con esta primera tabla haciendo referencia a una comparativa de precios, podemos destacar PiensaSolutions, ya que tienen un precio más bajo el primer año y además a partir del primer año también es el que ofrece un precio más económico, en la siguiente tabla analizaremos las características de los distintos proveedores para elegir definitivamente un hosting para nuestro proyecto.

Podemos ver 3 servicios que por medio de estrategias de marketing y descuentos los primeros meses o el primer año tienen un precio parecido, como son IONOS, Hostinger y PiensaSolutions, por el contrario, están Raiola Networks y ProfesionalHosting con un precio más elevado.

Servicio	Núcleos	RAM	SSD	Velocidad	Transferencia	Soporte
IONOS	2	2GB	80GB	400 MB/s	Ilimitada	Ubuntu 20.04, Ubuntu 18.04, CentOS 8, CentOS 7, Debian 10, Debian 9, OpenSuse Leap 15, Windows (+10€)
Hostinger	2	2GB	40GB	100 MB/s	2TB	CentOS, Ubuntu, Fedora, Debian, Suse
PiensaSolutions	2	4GB	80GB	400 MB/s	Ilimitada	Linux, Windows
Raiola Networks	2	2GB	20GB	100 MB/s	Ilimitada	Ubuntu 22.04, CentOS 7, Debian 11, AlmaLinux 8
ProfesionalHosting	2	3GB	30GB	100 MB/s	Ilimitada	CentOS

Tabla 2. Análisis de recursos para hostings VPS

En esta segunda tabla podemos analizar las características:

- **IONOS** ofrece un plan llamado VPS M, en el que disponemos de un procesador de 2 núcleos, 2GB de RAM y un disco duro de estado de 80GB, tenemos una velocidad de transferencia de 400 MB/s y disponemos de una gran variedad de sistemas operativos para instalar en el servidor, sin embargo, si queremos instalar Windows tendríamos que pagar un suplemento de 10€ al mes que es más de lo que cuesta el primer año, aun así, nuestra intención no es instalar un Windows Server. Tenemos que remarcar que tiene un buen soporte por medio de chat, teléfono y email. En definitiva, para el precio que tiene es una buena opción ya que es económico y sus características son similares a otros proveedores, IONOS es una empresa con recorrido en este tipo de servicios por lo que sería fiable, no obstante, ofrecen un periodo de prueba de 1 mes, por lo que si al mes has tenido problemas con este hosting VPS podrías darte de baja sin ningún coste.

- **Hostinger**, sigue un estilo de negocio parecido a IONOS, ofreciendo un precio bajo con una calidad aceptable con descuento en los primeros 12 meses. Las características son también muy parecidas a las de IONOS, podemos destacar como diferente que tiene la mitad de tamaño de disco duro, 40GB y que en velocidad de transferencia ofrece 100 MB/s con una transferencia mensual de 2TB, ofrece muchas distribuciones de los siguientes sistemas operativos: CentOS, Ubuntu, Fedora, Debian, Suse. Ofrecen un cPanel/WHM preinstalado y tienen un buen soporte de chat y tickets 24/7. Es otra buena opción por el precio ofrecido ya que tiene un descuento de mayor duración que IONOS que sería su competidor en cuanto a oferta y calidad.

- **PiensaSolutions**, esta empresa es más conocida por registros de dominios que por servicios hostings, aun así, tiene ofertas muy interesantes, como podemos observar en ambas tablas es el proveedor que ofrece mejores características y a más bajo coste, sin embargo, no es una empresa consolidada en el mercado de hosting profesional y algunas veces sus servidores no están bien optimizados. Como hemos dicho destaca por sus características ya que ofrece un procesador de 2 núcleos, RAM de 4GB y un disco duro de estado sólido de 80GB, con un servicio de 400 MB/s y una transferencia mensual ilimitada, en mi opinión es un servicio muy amplio de características de muy bajo coste con una calidad aceptable, aunque no es la mejor, pero resulta una opción muy interesante.

- **Raiola Networks**, es una empresa española con un recorrido joven en el mercado de hosting, pero debido a su calidad es de las mejores en cuanto al mercado español. Una de las mayores virtudes es el soporte que dan ya que disponen de un servicio de soporte telefónico y de tickets 24h al día, sorprende la rápida respuesta que tienen. En cuanto a las características, no destaca, de hecho, es el proveedor que ofrece peores características por un precio tan elevado, un procesador de 2 núcleos, 2GB de RAM y 20 GB de SSD con una velocidad de transferencia de 100 MB/s, sin embargo es de los proveedores que tiene una velocidad de respuesta más rápida, por lo tanto su calidad es indudable, está claro que es una opción buena aun que tiene un alto coste

- Por último, tenemos a **ProfesionalHosting**, una empresa con un largo recorrido en el mercado de hosting, llevan más de 19 años ofreciendo estos servicios en España. Tienen unas buenas características, se diferencia de los demás (Sin contar PiensaSolutions) en que su plan ofrece 3GB de RAM, el resto de características son similares, PrefesionalHosting, como Raiola Networks, destacan por su calidad y sus precios en hosting compartidos, ya que disponen de precios mas bajos y competitivos, sin embargo, en cuanto a servicio hosting de VPS tienen una muy buena calidad, pero unos precios muy altos para las características del servidor que ofrecen.

En conclusión, después de analizar una selección de los mejores proveedores de servicios de hosting VPS, nos quedamos con la opción de PiensaSolutions, ofrece un servicio con un coste muy bajo con las mejores características de la selección y cumple con los requerimientos de Magento Open Source eCommerce 2.4.0, y aun que hay opiniones que no lo ven como el mejor servicio nosotros ya hemos utilizado antes un servicio de hosting de PiensaSolutions y no hemos tenido ningún problema, además, la tienda de nuestro cliente ahora mismo es un negocio pequeño en el que debemos priorizar la mejor relación calidad-precio para ajustarnos al presupuesto.

Figura 10. PiensaSolutions logo

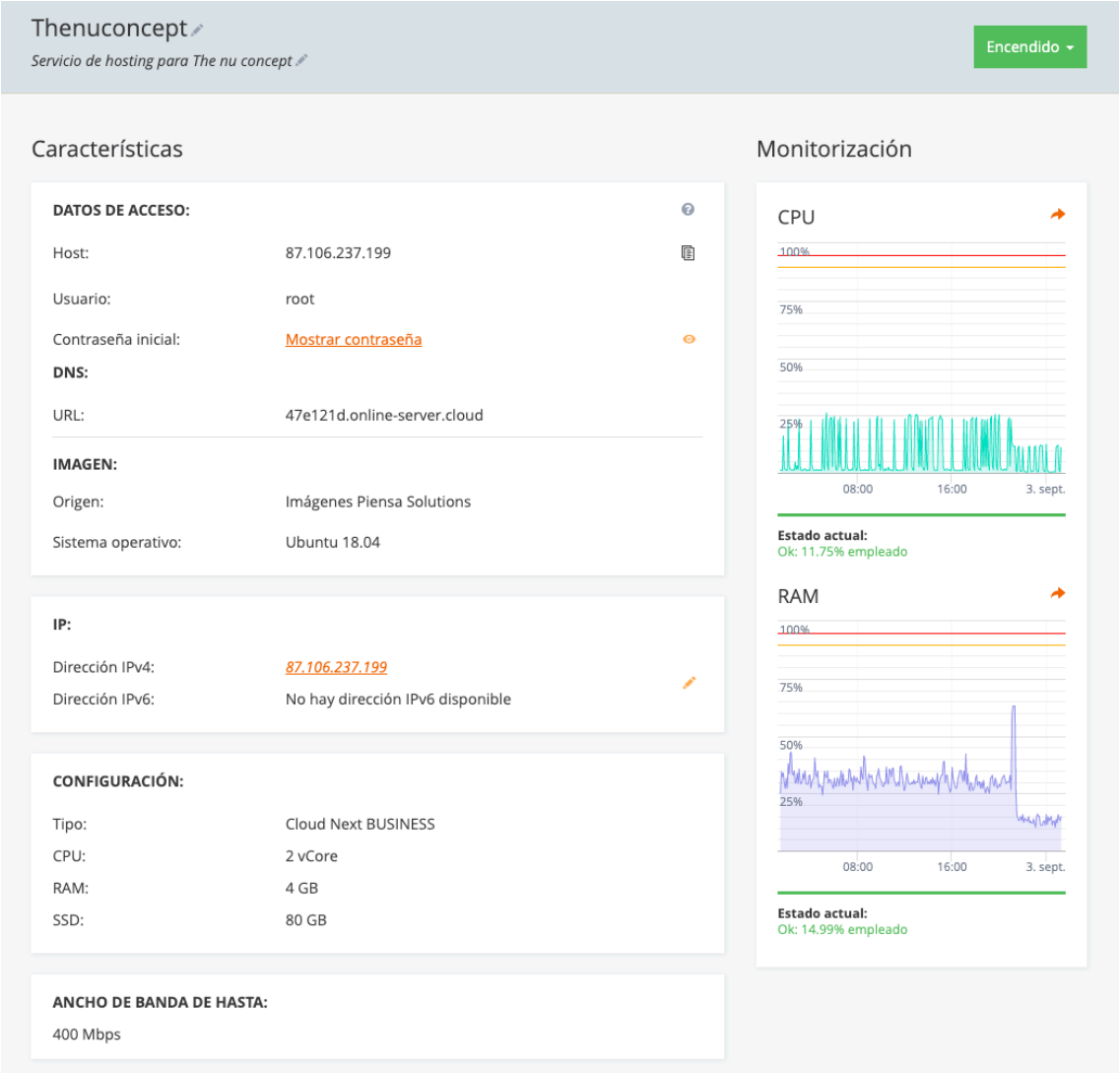


Figura 11. Vista de configuración del servidor por PiensaSolutions

CAPITULO 3. INGENIERÍA DEL SOFTWARE

3.1 Introducción

En este capítulo vamos a analizar la ingeniería del software requerida para realizar el proyecto, desde un análisis y elección de la metodología del software que se ha seguido, un análisis y diseño de los requisitos como una planificación, presupuesto, instalación y el proceso de implementación y pruebas.

Una buena planificación y un buen análisis de los requisitos nos hará escoger mejor la tecnología a utilizar, además de seguir una metodología adecuada, haciendo así que la relación entre presupuesto y calidad sea la idónea para desarrollar un proyecto en las condiciones óptimas.

3.2 Metodología de la ingeniería del software

3.2.1 Definición

Una metodología software es un conjunto de técnicas y métodos organizativos a seguir para desarrollar un proyecto de software informático. El objetivo de una metodología es organizar los equipos de trabajen para que trabajen de la misma forma optimizando el desarrollo de manera óptima.

Tener una metodología permite organizar el desarrollo de un proyecto de manera eficiente y ordenada, si un proyecto software no tiene una metodología a seguir puede desembocar en un desarrollo aún más complejo, que conducirá a problemas, retrasos, errores y, en definitiva, un mal resultado final. (24)

3.2.2 Metodologías

Existes distintas metodologías, aquí vamos a presentar algunas y explicaremos su funcionamiento y características.

Metodología Ágil

Esta metodología permite a la empresa desarrollar un proyecto de manera rápida y dinámica agilizando la toma de decisiones y enfocándolo a la producción, es decir, permiten darle flexibilidad al proyecto y gestionarlo de forma eficaz.

Esta metodología se caracteriza por entregar rápidas y tempranas, por considerar al cliente como parte del equipo de producción, se realiza una comunicación directa que estimula las actitudes y estructuras del equipo. (25)

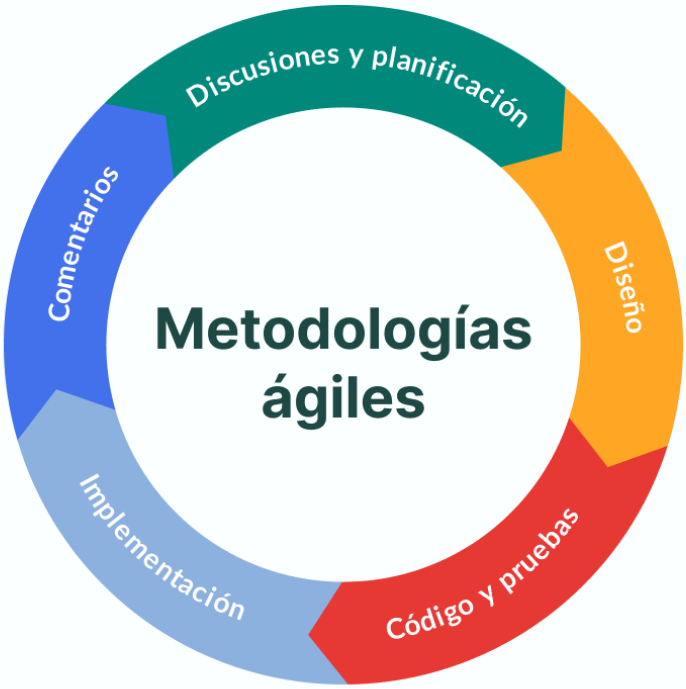


Figura 12. Esquema metodología ágil

Metodología iterativa

Esta metodología se creó en respuesta a las debilidades del método en cascada. La idea principal es desarrollar un sistema de manera incremental que favorezca al desarrollador y le permita sacar ventaja de lo aprendido en la versión anterior, así pues, se verían incrementadas las versiones entregables del sistema.

En resumen, esta metodología comienza por una versión inicial del software a desarrollar y se van lanzando versiones nuevas hasta que el proyecto esté terminado, en cada versión puede haber mejoras notables del software, mejoras estructurales o mejoras de diseño, en definitiva, la meta consiste en crear un sistema de soporte para que el usuario retroalimente el proyecto así mejorando en cada versión.

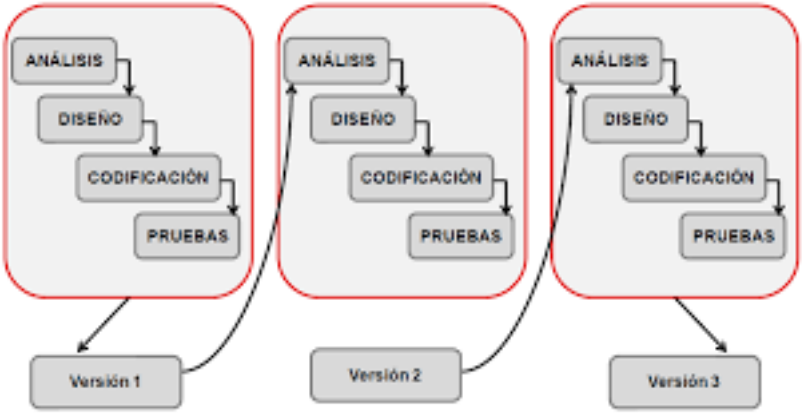


Figura 13. Esquema metodología iterativa

Metodología en cascada

La metodología en cascada a diferencia de otros métodos trata de definir bien las fases del proyecto, es decir, las fases establecidas no se ejecutan varias veces como en otros modelos, se establecen y se ejecutan y cada fase depende de la anterior.

Es un proceso lineal, que en la práctica puede verse aplicado con diferentes etapas, pero la más común son 5 etapas fijas:

- **Análisis**, se realiza una planificación y análisis de los requisitos
- **Diseño**, diseño del sistema
- **Implementación**, instalación y desarrollo del sistema
- **Verificación**, mediante la realización de pruebas
- **Mantenimiento**, entrega del proyecto, mejoras y mantenimiento

Figura 14. Esquema metodología en cascada

3.2.3 Elección de la metodología

Después de analizar varias metodologías y ver su funcionamiento creemos que para nuestro proyecto la mejor metodología seria la **metodología en cascada** ya que desde un inicio se han establecido unos requisitos y objetivos fijos con el cliente y por la elección previa de un CMS como Magento que ha establecido el tipo de tienda previamente, y aun que sea un proceso más largo que con otras metodologías, es un proceso más seguro y con un trabajo de mayor calidad debido a la preparación que este requiere.

3.3 Análisis y diseño

3.3.1 Definición de requisitos

Una buena definición de los requisitos nos ayudará a hacer un proyecto completo, el cumplimiento de estos nos llevará a un proyecto realizado con éxito.

Destacamos dos tipos de requerimientos, requerimientos funcionales y requerimientos no funcionales.

3.3.2 Requisitos funcionales

Estos requisitos establecerán la acción del software y que servicios debe proporcionar, como eje central de nuestros requisitos funcionales tendremos al usuario que interactúa en la tienda, y este además puede ser un usuario registrado, usuario no registrado o usuario administrador. Podremos ver estos requisitos gráficamente en el apartado de casos de uso. (12) (13)

Usuario

- **Navegar por la web**

  El usuario tiene la posibilidad de navegar a través del sitio web accediendo a vistas como “Sobre mí”, para obtener información de la marca, acceder a la vista de “Contacto”, para obtener información de contacto, además de obtener información legal a través del footer.

- **Consulta de categorías**

  El usuario puede consultar las categorías existentes accediendo a través del menú a “Productos” y así obtendría los productos de todas las categorías.

- **Consulta detalles producto**

  El usuario puede consultar los detalles de cada producto, desde una breve descripción, como una descripción más detalladas, fotos, precio, etc.

- **Búsqueda de productos**

  El usuario puede realizar una búsqueda de productos por medio de la barra de búsqueda situada en la barra de navegación utilizando palabra/s clave.

- **Envío review de producto**

El usuario puede realizar una reseña del producto desde la propia pantalla de detalles del producto, opinando sobre el producto para que otros usuarios puedan revisar su reseña.

- **Cambiar de idioma**

El usuario puede cambiar el idioma del sitio web, eligiendo idioma en el selector que hay en la esquina superior derecha, las opciones son “ES” (español, por defecto) y “EN” (ingles).

- **Añadir productos al carro**

El usuario, sin necesidad de estar logueado o registrado, puede añadir productos al carro para más tarde registrarse o loguearse y realizar la compra.

Usuario registrado

- **Añadir productos a la lista de deseos**

Un usuario registrado podrá añadir productos a la lista de deseos, esta lista será accesible desde la cuenta del usuario y tiene otro caso de uso.

- **Iniciar sesión**

Un usuario, al estar registrado, tiene la posibilidad de iniciar sesión en el sitio web.

- **Modificar información de usuario**

Un usuario registrado y que ha iniciado sesión puede modificar información de su cuenta, como información de envíos, información de nombre o editar la contraseña.

- **Cerrar sesión**

Un usuario registrado y con sesión iniciada puede cerrar sesión y saldrá a la página principal con la sesión cerrada.

- **Acceder al historial de pedidos**

Un usuario registrado puede acceder al historial de sus pedidos para ver los detalles de estos y comprobar su estado.

- **Acceder a la lista de deseos**

Un usuario registrado puede acceder a la lista de deseos desde su panel de usuario y ver los productos que ha ido añadiendo a la lista.

Usuario no registrado

- **Crear cuenta de usuario**

Un usuario no registrado tiene la posibilidad de registrarse y crear una cuenta de usuario para poder disfrutar de los servicios de la tienda.

Usuario administrador

- **Iniciar sesión**

Un usuario administrador puede acceder al panel de administración de la tienda online iniciando sesión.

- **Cerrar sesión**

Un usuario administrador con sesión iniciada en el panel de administración puede cerrar su sesión.

- **Administrar categorías de productos**

Un usuario administrador puede administrar las categorías de los productos, acciones como añadir, editar y eliminar categorías.

- **Administrar productos**

Un usuario administrador puede administrar los productos, creando nuevos, editando o eliminado los ya existentes, además de crear su versión en inglés.

- **Control de pedidos y envíos**

Un usuario administrador puede controlar los pedidos y envíos, actualizando su estado, confirmando la recepción del pago o confirmando el envío del producto.

- **Generar facturas**

Un usuario administrador puede generar facturas sobre pedido realizados de clientes.

- **Administrar reviews**

Un usuario puede administrar las reseñas enviadas por los clientes sobre los productos de la tienda.

- **Control de estadísticas**

Un usuario administrador puede revisar las estadísticas y gráficas de la tienda, sobre ventas

3.3.3 Requisitos no funcionales

Los requisitos no funcionales se centran en la calidad del sistema, no en las funciones de este, es más 'como debe ser un sistema' que 'que debe hacer un sistema', esto último serían los requerimientos funcionales.

En su mayoría derivan de los requisitos funcionales, y además complementan a estos. Desarrollaremos algunos de los requisitos no funcionales de este proyecto. (12) (13)

- **Rendimiento**

Este requisito requiere que el servidor tenga un rendimiento aceptable para el correcto funcionamiento de la tienda, Magento necesita tener unos mínimos requerimientos que debemos de cumplir para que nuestro servidor, donde se alojará la tienda online, tenga un tiempo de respuesta aceptable y además permita que un número de usuarios considerado sea capaz de acceder y comprar productos al mismo tiempo.

- **Apariencia y adaptabilidad**

La tienda debe tener un diseño, por petición del cliente, sencillo, moderno e intuitivo, que no sea un diseño sobrecargado de información, que presente el producto de forma clara y le sea fácil comprar lo que le guste.
Además, la página se tiene que ofrecer en español por defecto, pero tener la posibilidad de cambiar de idioma a inglés, es decir ser una web multilenguaje.
Por último, en cuanto a adaptabilidad, el sitio tiene que ser "Responsive Design", es decir, debe poder adaptarse a la resolución de cualquier dispositivo y verse correctamente.

- **Software**

Desde el punto de vista de los usuarios, solo necesitaran un navegador para acceder al sitio web desde un dispositivo.

Desde el punto de vista del programador vamos a necesitar software para conectarnos con el servidor, como Filezilla, la terminal del dispositivo donde realizaremos las acciones, en este caso un MacBook Air 2020, Visual Studio Code, este nos permitirá conectarnos al servidor por medio de una extensión y podremos editar o desarrollar cualquier fichero del servidor, también necesitaremos Magento, este lo clonaremos desde el repositorio Open Source.

Además, para tareas de diseño hemos utilizado el Adobe Photoshop 2022.

- **Hardware**

Por parte del cliente no se requiere de ningún hardware concreto, ya que será accesible desde cualquier dispositivo electrónico que disponga de un navegador, ya sea un ordenador, teléfono móvil o tablet.
Por parte del desarrollador, vamos a necesitar un ordenador, y desde el propio navegador se puede simular la pantalla para dispositivos móviles y tablets, aun así, también comprobaremos el funcionamiento y la adaptabilidad del sitio. El ordenador también nos servirá como instrumento para configurar el servidor, instalar Magento y desarrollar el proyecto.

- **Seguridad**

El sitio web dispondrá de la seguridad suficiente para garantizar que la información del usuario es solo accesible por el y por el administrador del sistema, encriptando la información delicada como la contraseña o sistemas de pago. Para garantizar que el intercambio de información entre usuario y servidor es seguro se instalará un certificado SSL en el servidor que encripta la información y garantizará la seguridad al usuario, este certificado SSL estará asociado al dominio “thenuconcept.com” y estará instalado en el servidor.

- **Mantenimiento y mejoras**

Este requisito garantiza que se producirá un mantenimiento del sitio actualizando y añadiendo las categorías y productos de la tienda si se requiere. Además, se podrá planificar mejoras a corto plazo y a largo plazo si la producción y ventas aumentan para mejorar el sitio y ofrecer un mejor servicio al usuario.

3.3.4 Planificación

Este proyecto se desarrollará en tres etapas para tener una mayor organización

- **Etapa 1.** En esta etapa se realizará un análisis del problema, de los objetivos de este proyecto y unos requisitos que deberemos tener en cuenta. Además de esta introducción en esta etapa tendremos que realizar un estudio de los CMS actuales para elegir uno adecuado a nuestro proyecto, además de un estudio de los servicios de hosting actuales para elegir el más acorde con nuestro CMS y nuestras necesidades cumpliendo con un presupuesto

previamente establecido. Por último, en esta etapa, habrá que analizar que certificado SSL podemos tramitar teniendo en cuenta el presupuesto que llevamos y los requerimientos de nuestro negocio online para este certificado.
Duración: 24 horas

- **Etapa 2.** Para esta segunda etapa nuestro esfuerzo se va a centrar en que, una vez teniendo claro las herramientas a utilizar, instalar los requisitos necesarios en nuestro servidor para poder instalar Magento, y una vez instalado, configurarlo con la información, categorías y productos del negocio. Para terminar esta etapa nos centraremos en aportarle al sitio un diseño simple, moderno e intuitivo.
Duración: 32 horas

- **Etapa 3.** A continuación, configuraremos un método de pago actual, como tarjeta o PayPal y a través de un sistema de testeo realizaremos las pruebas necesarias para comprobar que el sitio funciona correctamente, además de añadirle una configuración SEO al sitio a partir de las herramientas que ofrece Magento para poder posicionar al sitio de la mejor forma posible.
Duración: 24 horas

- **Etapa 4.** Por último, desarrollaremos la memoria para este proyecto, cumplimentando la documentación requerida.
Duración: 40 horas

Duración total: 120 horas

Destacamos que las horas se distribuirían en jornadas laborales actuales de 8 horas y de lunes a viernes, por lo que la duración del proyecto en días adaptada a la jornada laboral serían de **15 días** que sin contar los fines de semana sería de **3 semanas**

En la siguiente tabla podemos observar específicamente los detalles de la planificación del proyecto.

Etapa	Tarea	Duración
1	Análisis del problema	6 horas
	Análisis y elección del CMS	6 horas
	Análisis y elección de servicio de hosting	6 horas
	Análisis y elección de certificado SSL	6 horas
2	Configuración del servidor	8 horas
	Instalación CMS	8 horas
	Desarrollo del proyecto	16 horas
3	Configurar métodos de pago	8 horas
	Pruebas y corrección de errores	16 horas
4	Desarrollo de memoria	40 horas
	Total	**120 horas**

Tabla 3. Planificación, duración del proyecto

Además de planificar la duración del proyecto debemos adaptarnos a un presupuesto, teniendo en cuanta los distintos recursos utilizados.

- **Servicio de hosting**, hemos analizado este servicio anteriormente para elegir como proveedor a PiensaSolutions por un primer precio anual de 72,06 €

- **Certificado SSL**, tenemos que incluir el coste del certificado que se pagará también de forma anual por un precio de 5,99 €, el cual se ha analizado anteriormente para su elección.

- **Coste de personal**, para calcular este coste hay que tener en cuenta varios factores, al ser un proyecto para una pequeña empresa y no requerir de una gran base de datos de productos ni cliente se puede utilizar a una persona que desarrolle varios roles que necesitamos.

  Necesitaríamos a un Analista, que realice los trabajos de investigación, análisis de requerimientos y requisitos, realice una planificación adecuada al proyecto y detalle los casos de uso.

  Por otra parte, necesitamos también un diseñador para modificar el diseño elegido para nuestra tienda y lo adapte a nuestra marca realizando los diseños necesarios para la personalización del sitio.

Por último, necesitaríamos un desarrollador web que realice la configuración del servidor, instalación del CMS y las pruebas necesarias para que el proyecto se pueda desplegar sin errores y con un correcto funcionamiento.

Como hemos dicho antes al ser un proyecto pequeño una persona realizara el trabajo de los roles especificados, por esa razón un trabajador IT de distintos roles puede tener un coste de media de 13,08€ / hora, por lo que, si nuestro proyecto tiene 120 horas, el coste total de personal será de 1570 €. (22)

Después de este análisis de costes necesarios vemos resumidos en la siguiente tabla el presupuesto necesario del proyecto.

Recurso	Coste
Servicio Hosting - PiensaSolutions	72,06 €
Certificado SSL - DonDominio	5,99 €
Coste personal	1570 €
Total	**1648,05 €**

Tabla 4. Planificación, presupuesto del proyecto

Una vez analizados los costes y la duración del proyecto podemos combinar estos dos factores mediante un **Diagrama de Gantt**, para observar de forma detallada la duración del proyecto y los costes medidos por día.

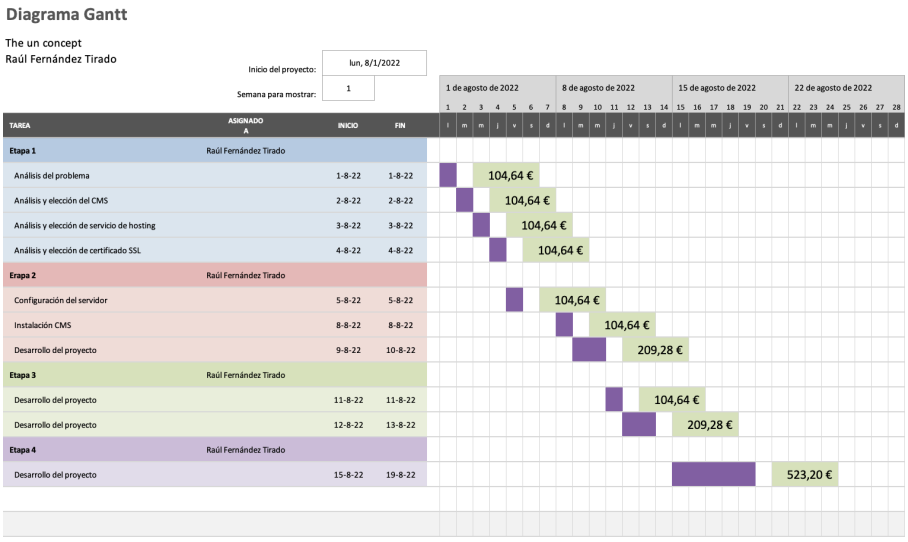
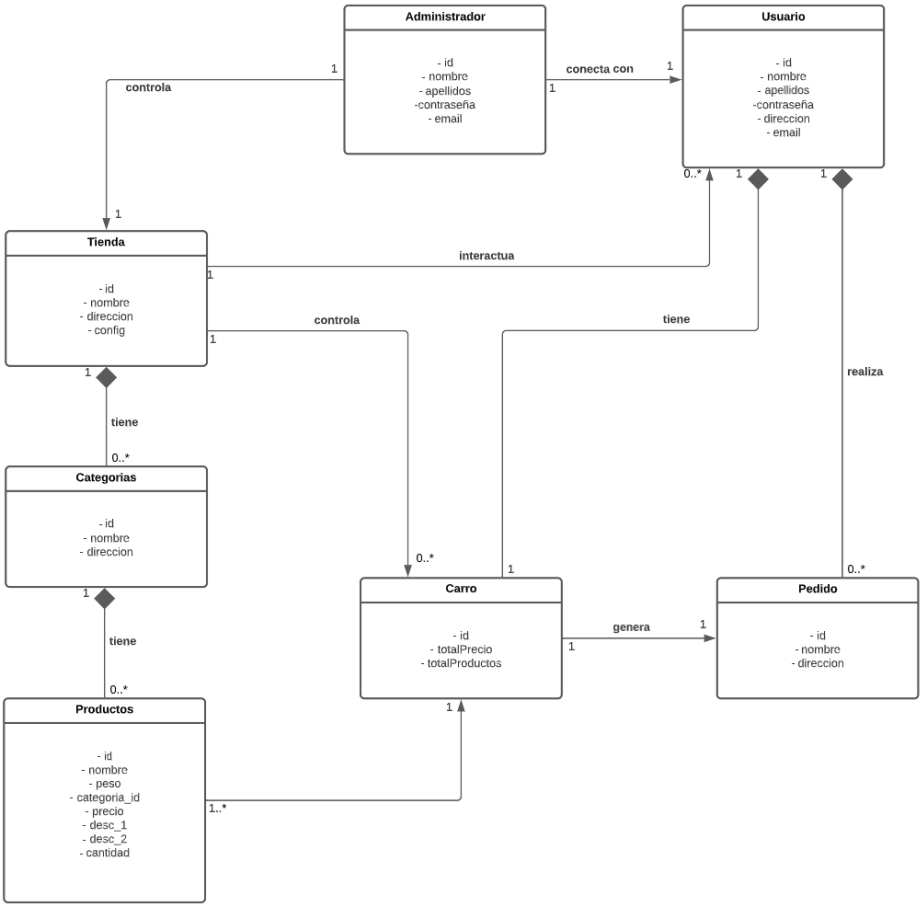


Figura 15. Diagrama de Gantt

3.3.5 Diagramas de clases

Un diagrama de clases es un tipo de diagrama de estructura estática, que muestra las entidades principales del sistema y sus relaciones entre ellas, se diseña mediante relaciones de agregación, composición y enlaces, mostrando en ella su cardinalidad para dar más detalle a una relación entre entidades.



3.3.6 Casos de uso

Los casos de uso muestran las diferentes relaciones que pueden tener un actor con el sistema.

Para nuestro proyecto tendremos los siguientes casos de uso según el usuario, el cual será nuestro actor, juntos con los diferentes tipos de usuario que puede ser (Usuario registrado, Usuario no registrado, Usuario administrador).

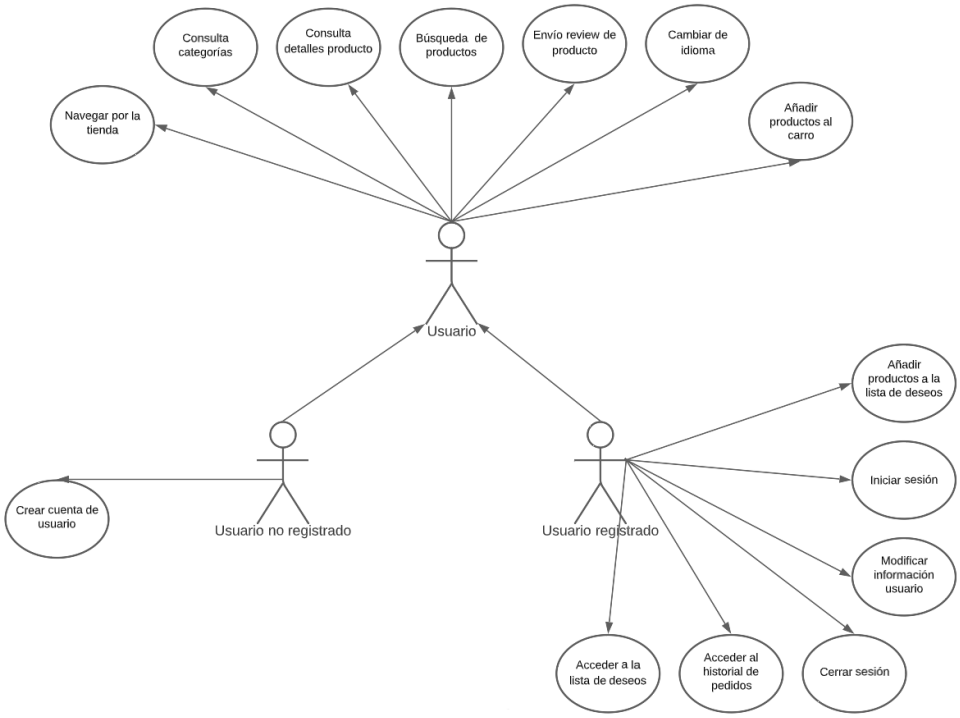


Figura 16. Casos de uso Usuario

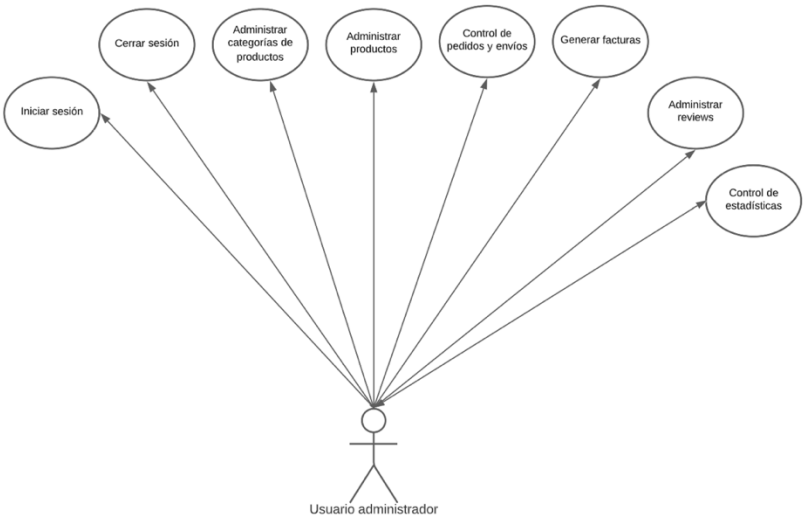


Figura 17. Casos de uso Usuario administrador

3.3.7 Diagramas de secuencias

Un diagrama de secuencias es un tipo de diagrama UML que podemos utilizar para mostrar la interacción de un actor con el sistema en un orden determinado, representa las solicitudes de un objeto (actor) que puede hacer al sistema y la respuesta de este, todo esto ilustrado en un diagrama UML. (23)

Representaremos los distintos diagramas de secuencia que pueden tener los actores, que teniendo en cuenta nuestro proyecto y los casos de uso los distinguiremos en Usuario, Usuario no registrado, Usuario registrado y Usuario administrador.

Usuario

Navegar por la tienda

En este caso el usuario puede realizar distintas acciones, como acceder a contacto y enviar un mensaje mediante el formulario de contacto, acceder a “Sobre mi” para obtener información sobre la marca, etc. Aquí vamos a mostrar un caso en el que un usuario navegando por la tienda envía un mensaje de contacto por el formulario de contacto.

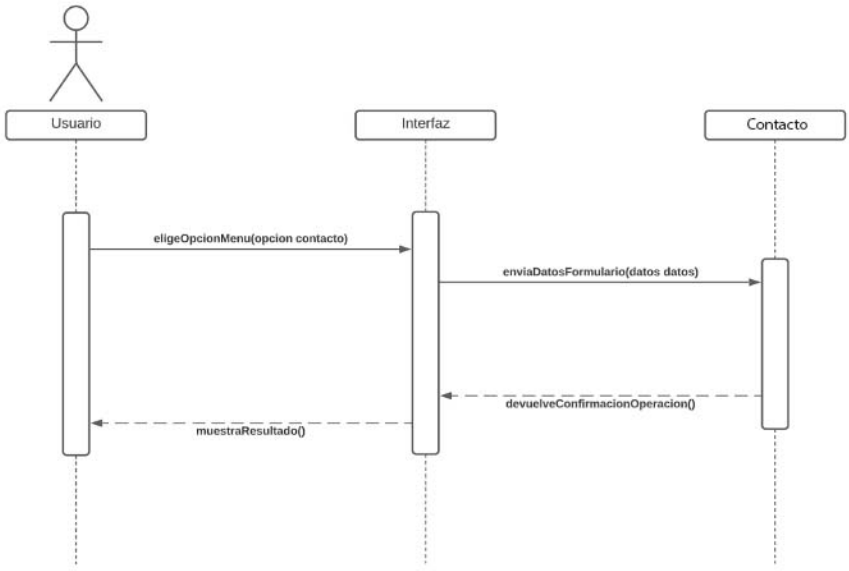


Figura 18. Diagrama de secuencia, navegar por la tienda

Consulta categorías

En este diagrama se muestra la opción que tiene el usuario para elegir la opción de la barra de navegación de “Productos” donde podrá ver un listado de todos los productos, sin un filtro por categoría.

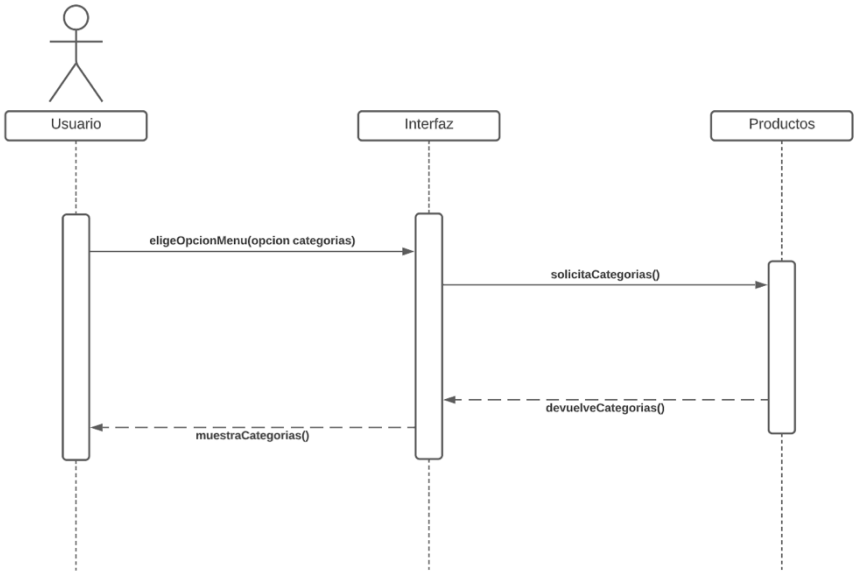


Figura 19. Diagrama de secuencias, consultar categorías

Consulta detalles producto

En este caso vamos a mostrar el diagrama de secuencia en el que el usuario accede las categorías y selecciona un producto, de este obtiene los detalles de este en una vista, detalles como el precio, medidas, descripción, fotos, etc.

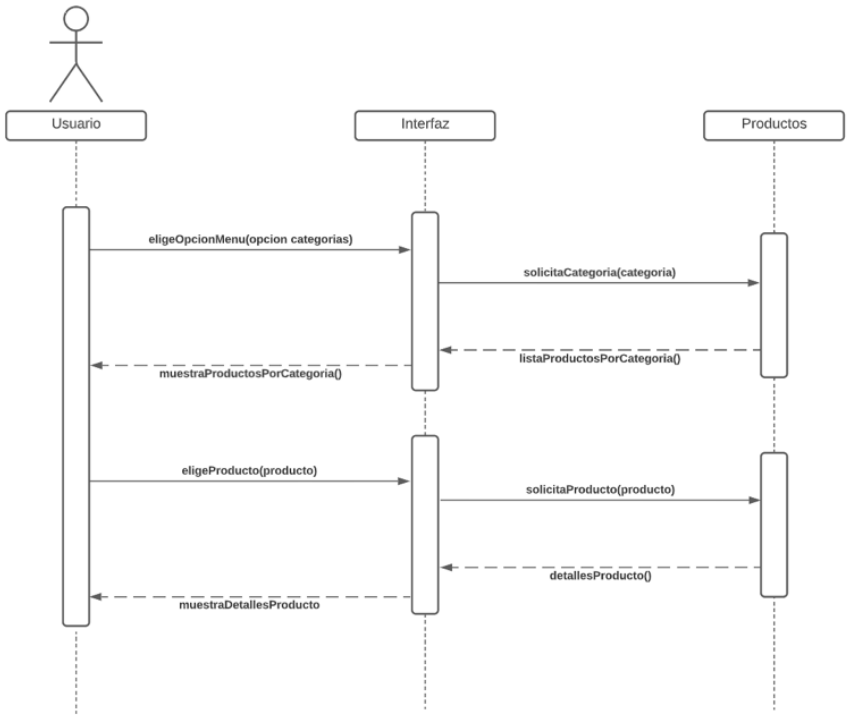


Figura 20. Diagrama de secuencias, consultar detalles producto

Búsqueda de productos

Para este caso a través la barra de búsqueda situada en la barra de navegación enviamos una cadena de texto que nos devolverá un listado de productos que concuerden con esa cadena de texto y de ahí seleccionaremos el deseado.

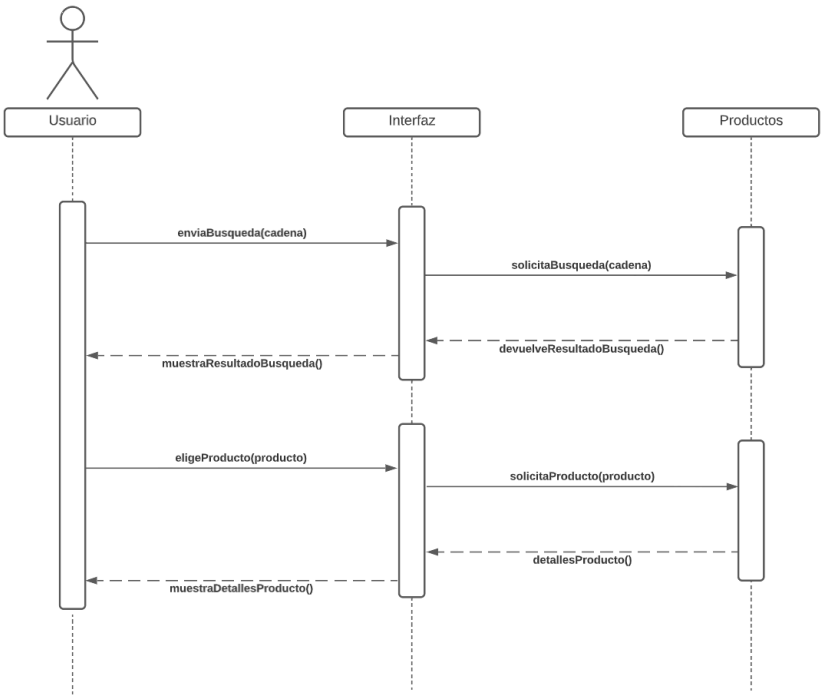


Figura 21. Diagrama de secuencia, búsqueda de productos

Envío review de producto

En este caso el usuario puede enviar una reseña, una opinión sobre el producto, para ello primero tiene acceder al producto y luego enviar la reseña escrita.

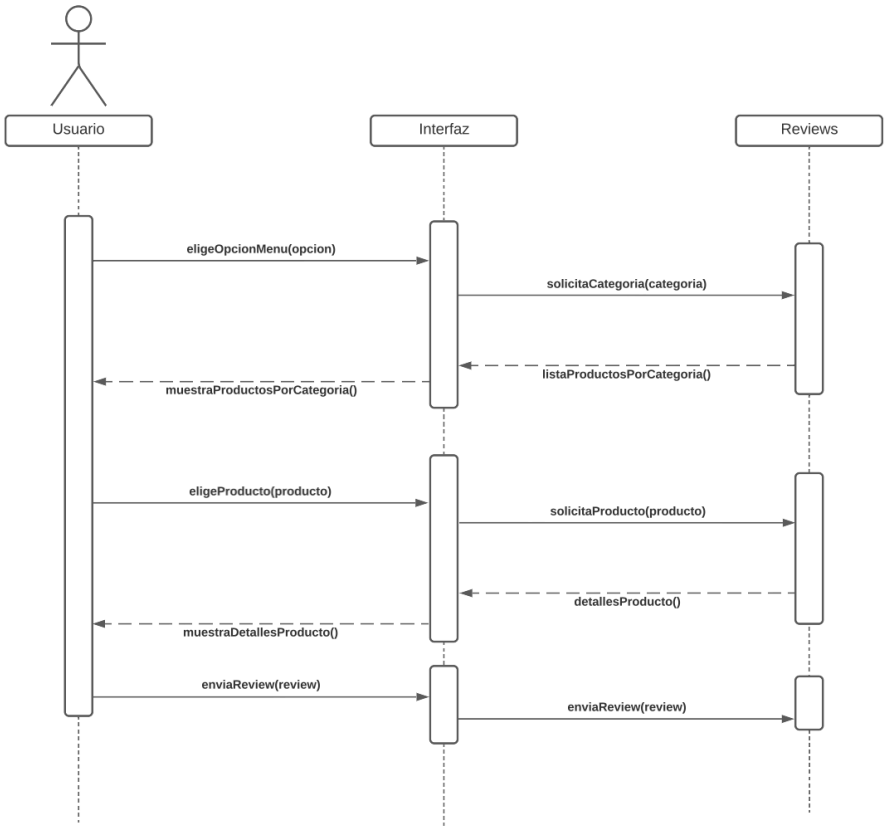


Figura 22. Diagrama de secuencia, envío de review

Cambiar de idioma

En este caso el usuario puede cambiar el idioma del sitio web cambiando la opción del selector situado en la esquina superior derecha, por defecto este se encuentra en ES (español) pero hay posibilidad de cambiar a EN (inglés).

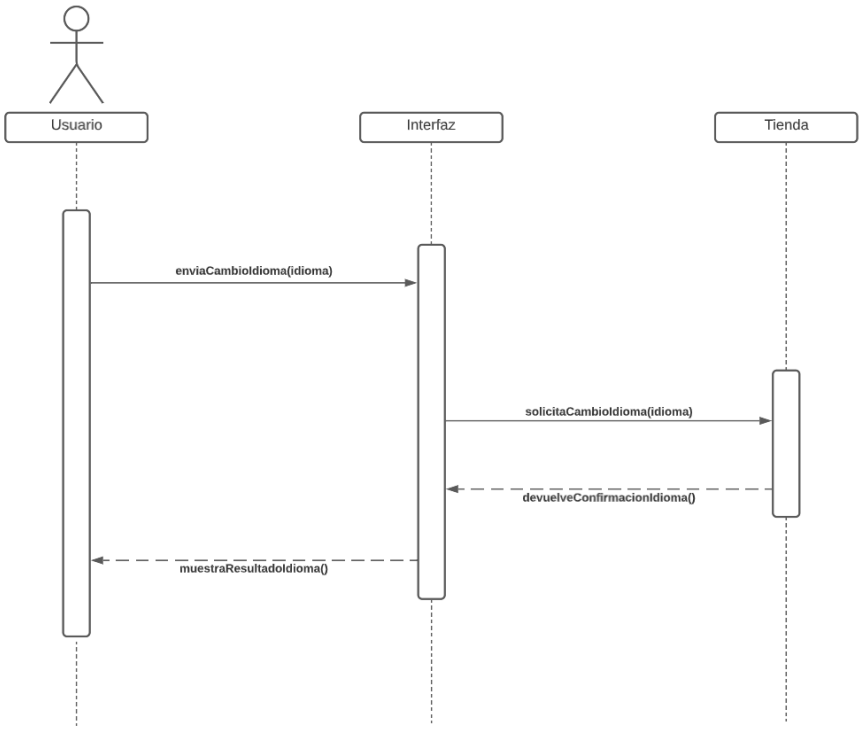


Figura 23. Diagrama de secuencia, cambiar idioma

Añadir productos al carro

En esta acción el usuario añadirá productos al carro, esta acción es válida para todo tipo de usuarios, el usuario tendrá que seleccionar una categoría, luego ver los detalles de un producto y añadir al carro el producto deseado si tiene existencias.

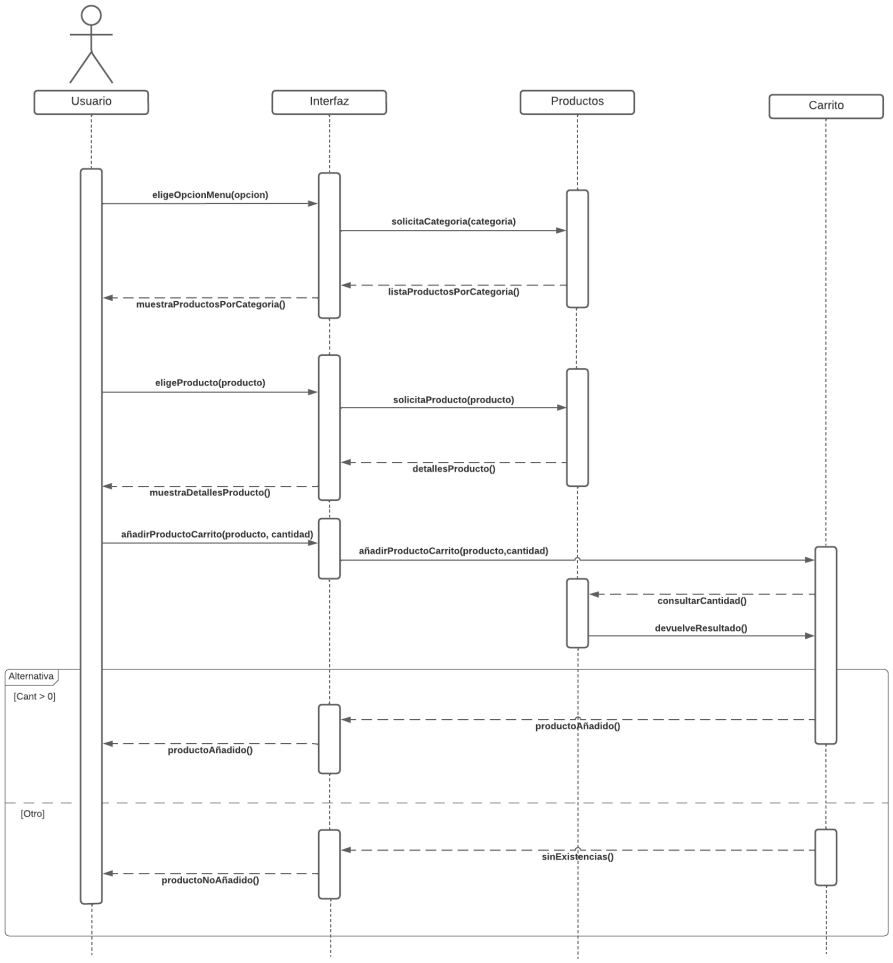


Figura 24. Diagrama de secuencia, añadir productos al carro

Usuario no registrado

Crear cuenta de usuario

En este caso un usuario no registrado puede realizar la acción de crear una cuenta de usuario. Para ello tendrá que acceder a la vista de registro, una vez allí tendrá que rellenar el formulario de registro y enviar este registro, en caso de que la validación de los campos de registro haya ido todo bien se le devuelve confirmación del registro, en caso de que haya habido algún problema de validación como email duplicado o contraseña incorrecta o débil, entonces devolverá un error de no registro.

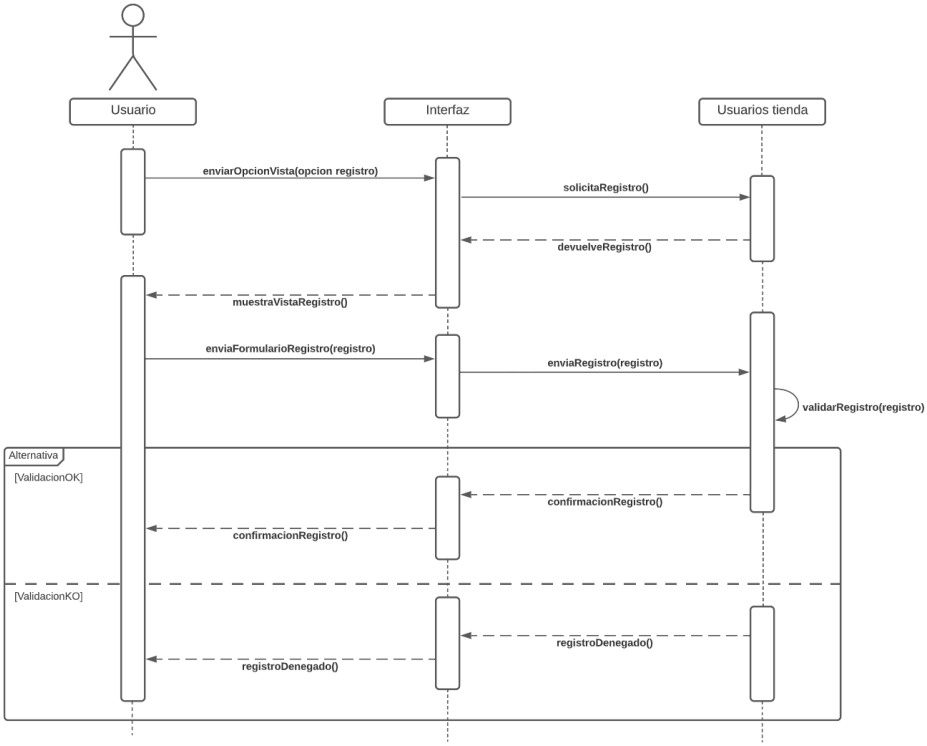


Figura 25. Diagrama de secuencia, registro de usuario

Usuarios registrados

Login

El proceso de login en el diagrama de secuencias es similar a la acción de registro, debido a que tienen que acceder a la vista de login en este caso y una vez allí rellenaran el formulario de login para enviarlo al servidor y este validará los datos enviados, en caso de que la cuenta de usuario coincida con el email y contraseña enviados devolverá una confirmación de que todo ha ido correctamente e iniciará sesión. En caso de que no exista usuario que coincida con el email y la contraseña enviado se devolverá un error.

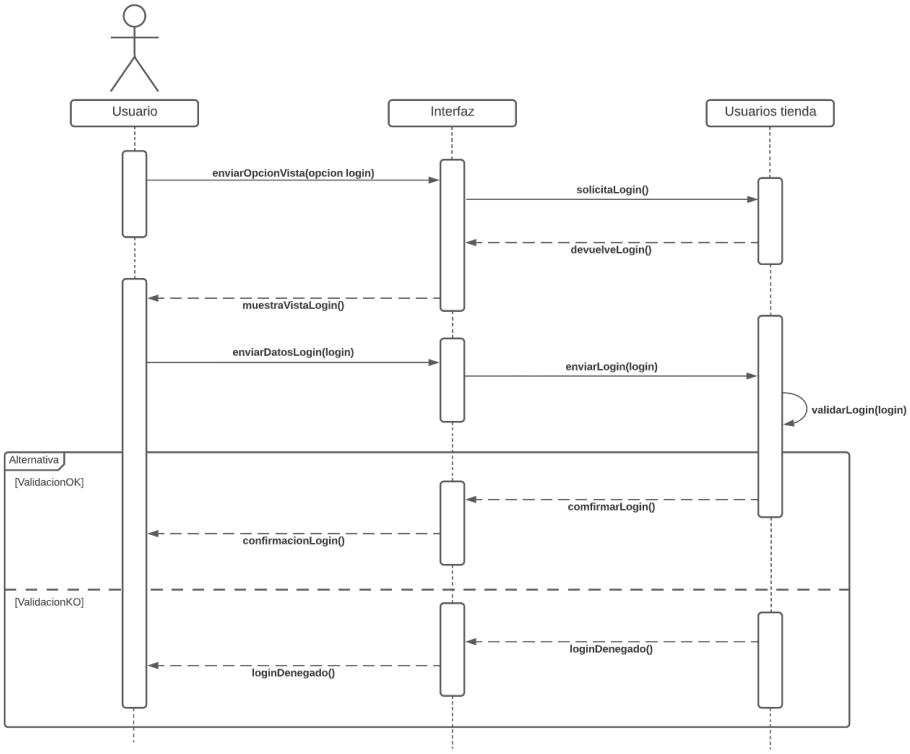


Figura 26. Diagrama de secuencia, login

Modificar información de usuario

Un usuario registrado previamente logueado podrá modificar la información de su usuario solicitando los camios por medio de un formulario que devolverá un mensaje de confirmación al actualizar los datos del usuario.

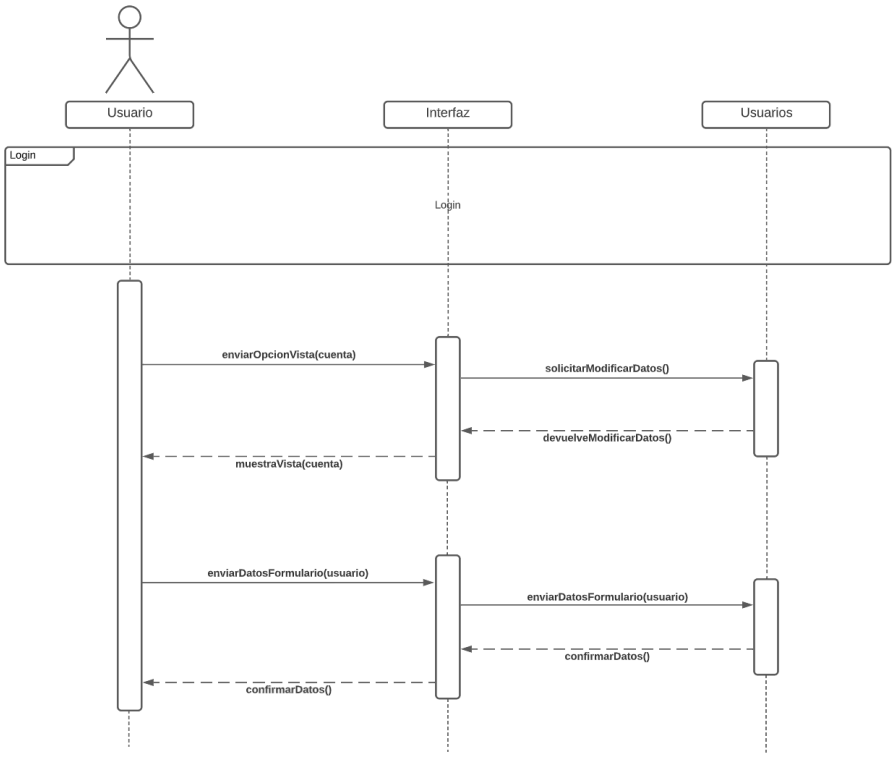


Figura 27. Diagrama secuencia, modificar información usuario

Cerrar sesión

Un usuario con sesión abierta tiene la posibilidad de cerrar la sesión con esta acción.

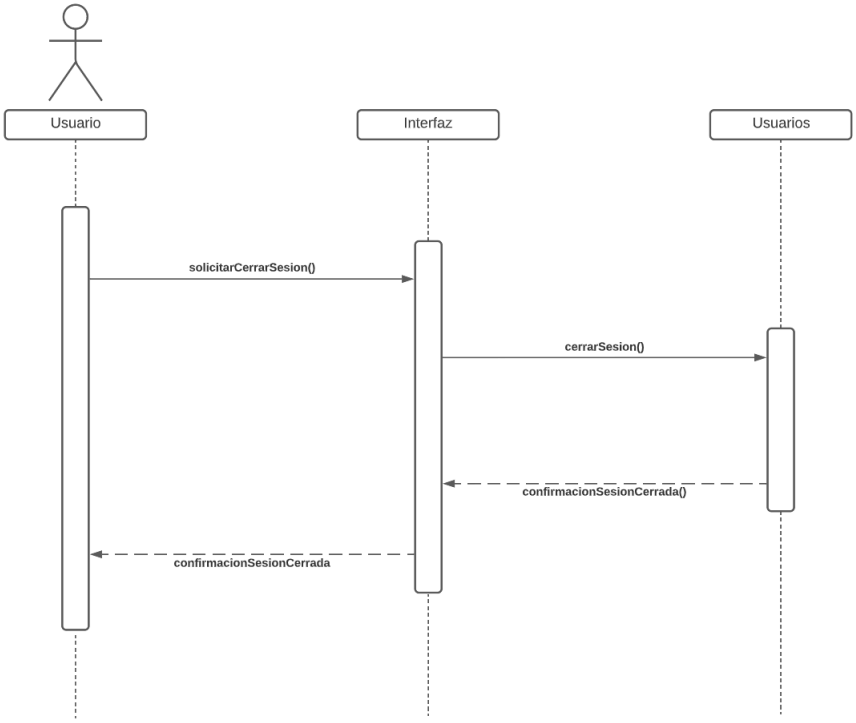


Figura 28. Diagrama secuencia, cerrar sesión

Acceder al historial de pedidos

Esta acción permitirá a un usuario previamente logueado a acceder al historial de sus pedidos para poder comprobar el estado de estos o acceder a los detalles de los pedidos.

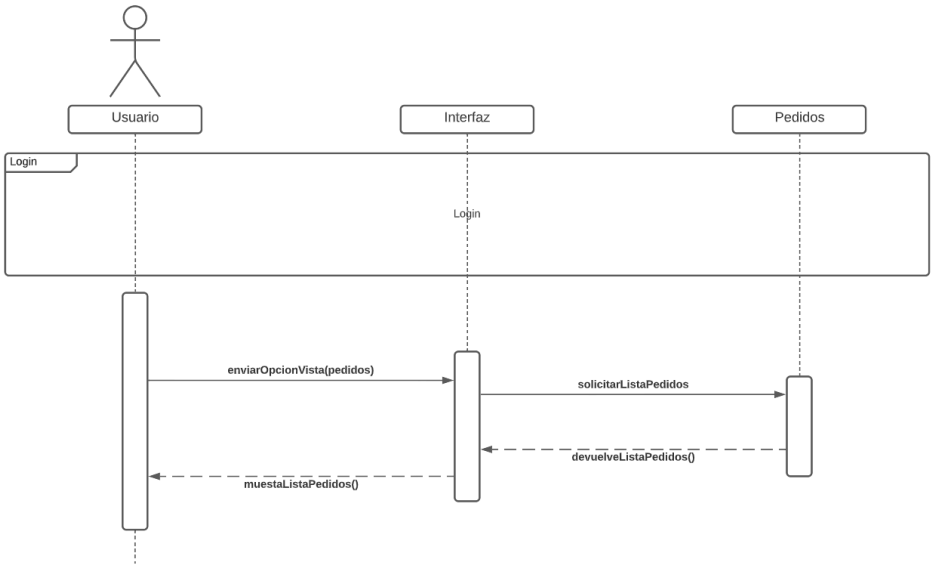


Figura 29. Diagrama secuencia, acceder historial pedidos

Añadir producto a la lista de deseos

En este caso un usuario previamente logueado, accede a los detalles de un producto para añadirlo a la lista de deseos.

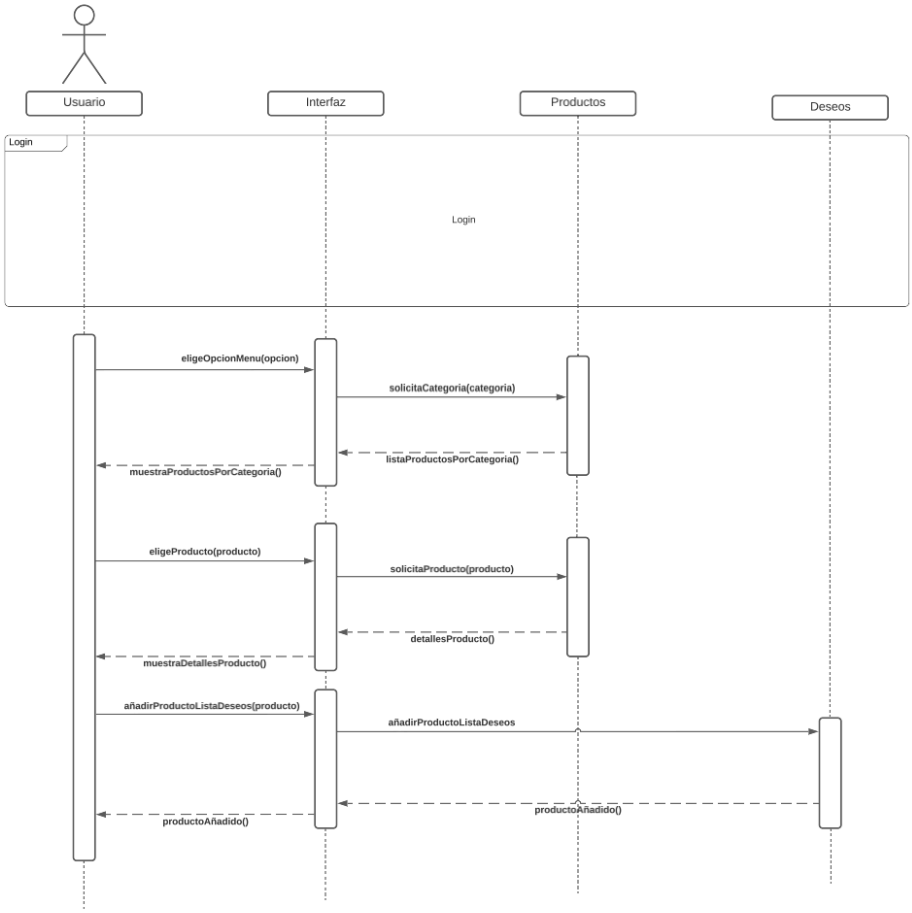


Figura 30. Diagrama secuencia, añadir producto lista deseos

Acceder a la lista de deseos

Este diagrama muestra la acción del usuario de acceder a la lista de deseos y comprobar los productos que tiene añadido si los hubiese.

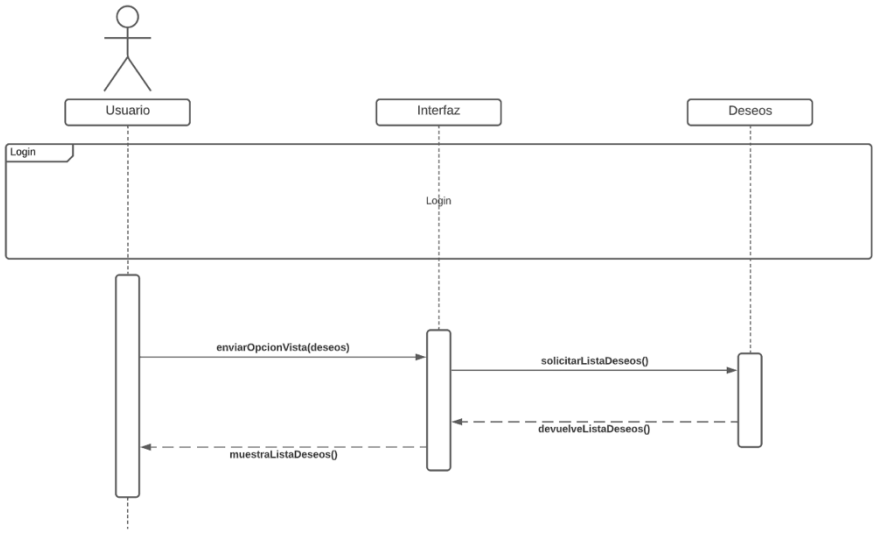


Figura 31. Diagrama secuencia, acceder lista deseos

Usuario administrador

Iniciar sesión

El inicio de sesión del usuario administrador es similar al del usuario normal, cambiando que el usuario administrador no tiene que entrar a una vista de login previamente ya que la pantalla inicial del panel de administración es el formulario de login.

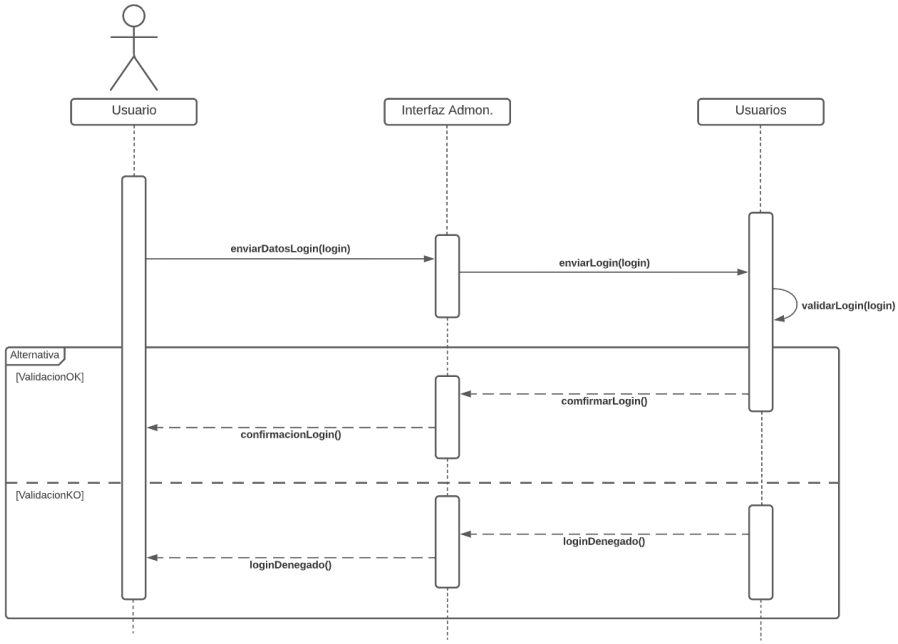


Figura 32. Diagrama de secuencia, inicio de sesión administrador

Cerrar sesión

Esta acción será similar al del usuario normal ya que lo único que tiene que hacer es cerrar la sesión actual y se le devolverá un mensaje de confirmación.

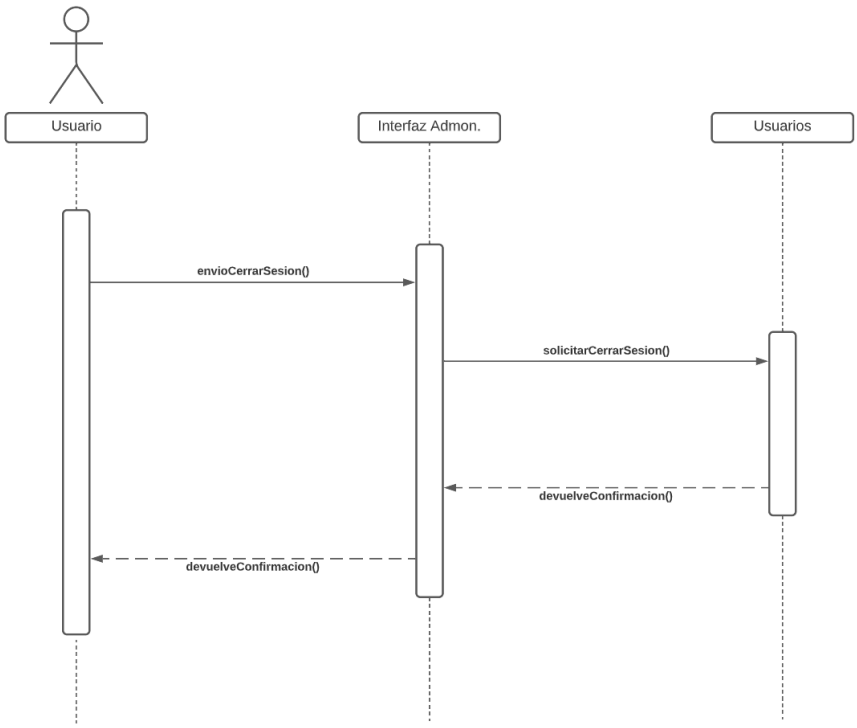


Figura 33. Diagrama de secuencia, cerrar sesión administrador

Administrar categorías de productos

En este diagrama se mostrará la acción donde el administrador estando logueado administra las categorías solicitando una acción sobre las categorías de los productos.

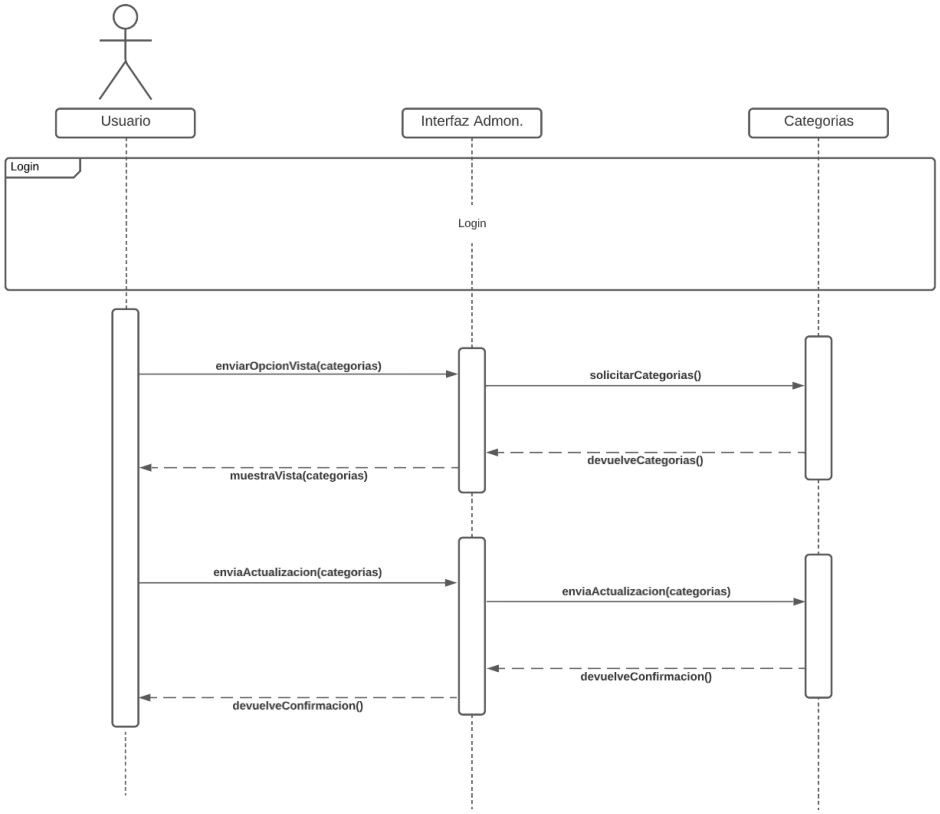
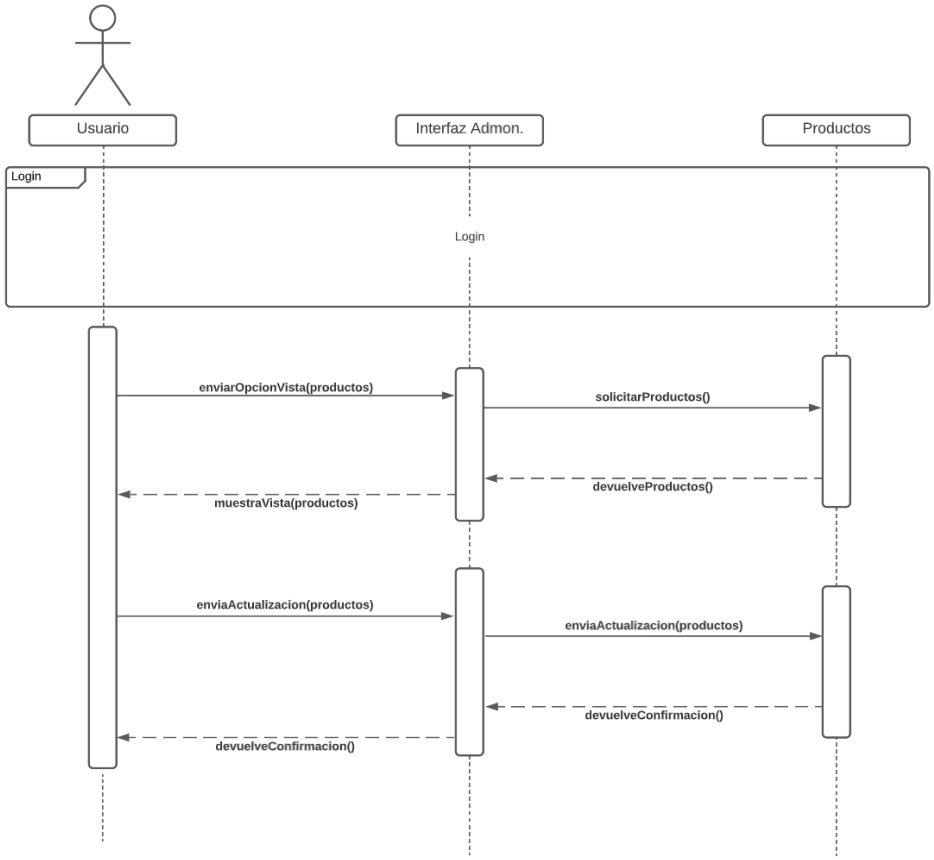


Figura 34. Diagrama de secuencia, administrar categorías

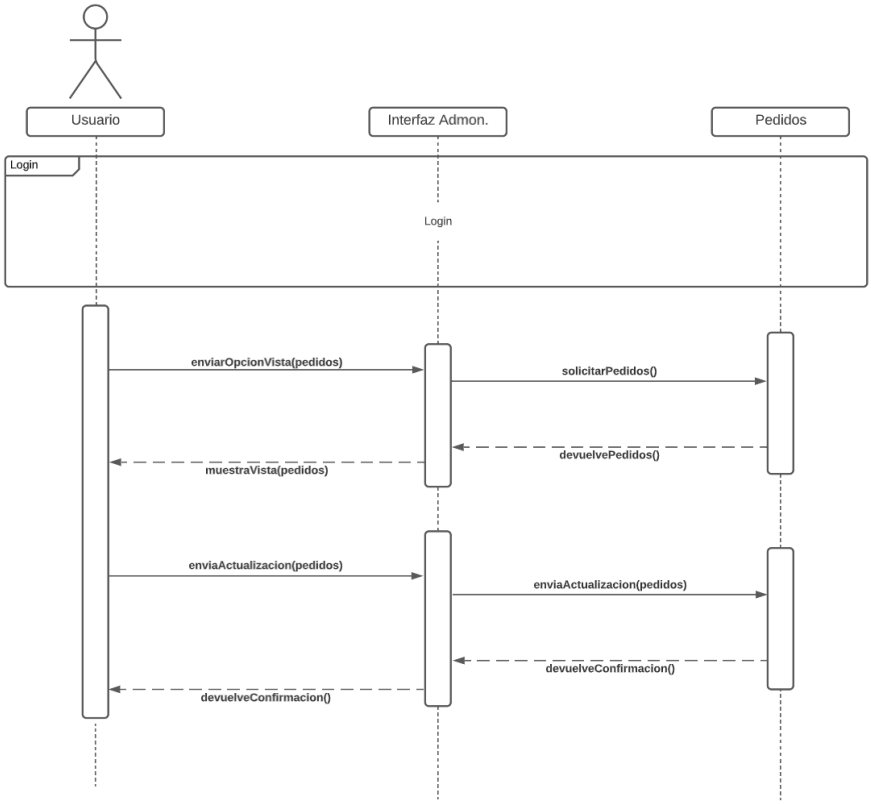
Administrar productos

Para administrar los productos, primero debe acceder a la vista de los productos donde tendrá una lista con todos los productos y luego podrá actualizar los detalles de los productos que desee.



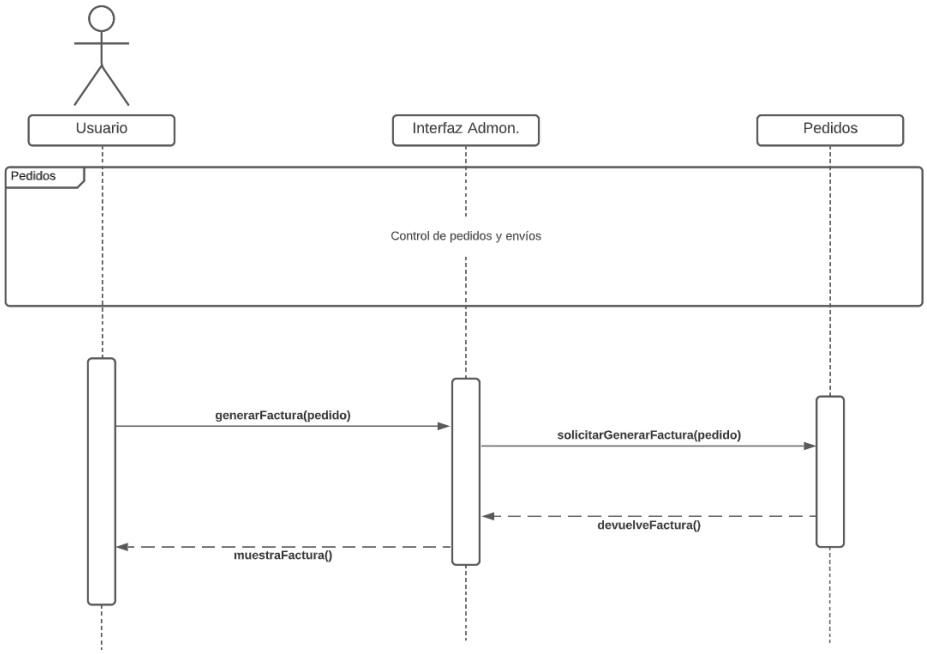
Control de pedidos y envíos

En este diagrama mostramos la acción de un usuario administrador que accede a la vista de pedidos para poder gestionarlos, aquí puede gestionar los pedidos y envíos, comprobar o actualizar estados y actualizar datos de los pedidos.



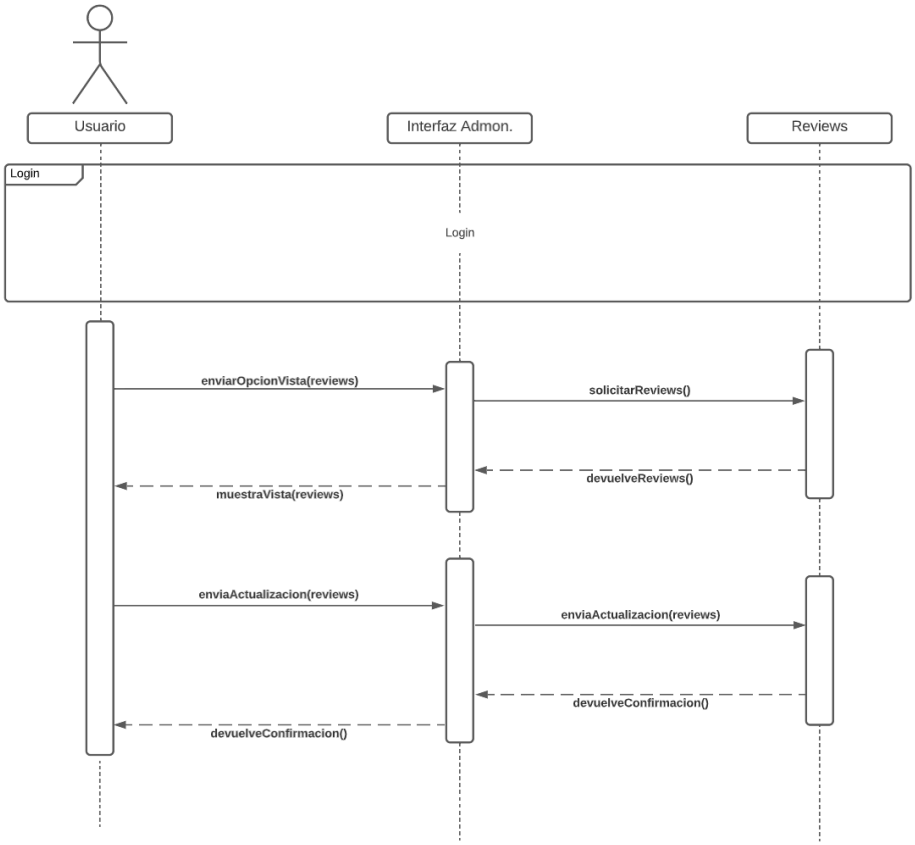
Generar facturas

Un usuario administrado una vez entrado al control de pedidos y envíos puede generar las facturas necesarias de un pedido concreto, solo tendrá que solicitar la generación de esta factura sobre el pedido deseado y se le mostrará la factura generada.



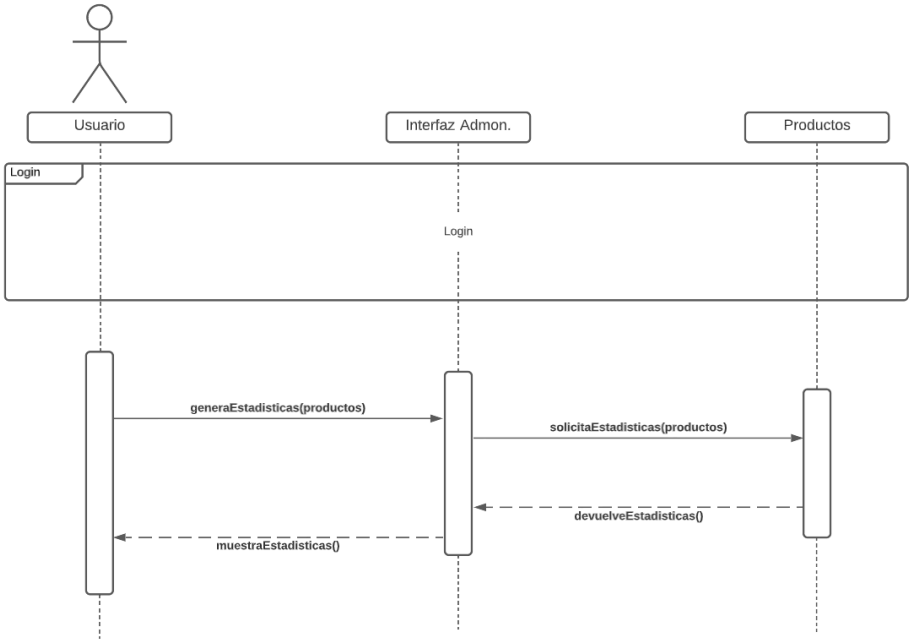
Administrar reviews

En este diagrama se observa la acción de un usuario administrador gestionando las reseñas, son los comentarios que envían los usuarios como opinión sobre productos, entrando en la sección de “reviews” se pueden observar las existentes y gestionarlas.



Control de estadísticas

El usuario administrador puede obtener las estadísticas referidas a los pedidos con estados completos, de esta manera se puede hacer un estudio del funcionamiento de la tienda y las ganancias obtenidas.



3.4 Instalación

Una vez analizado y elegido los requisitos previos tenernos que proceder con la instalación de Magento, esta instalación se puede dividir en dos etapas, en la primera etapa nos conectaremos a nuestro servidor mediante conexión SSH y configuraremos el servidor para la instalación de Magento con todos sus requerimientos, especificados en este apartado.

- **Configuración del servidor**

  Lo primero que tenemos que hacer es conectarnos por medio del protocolo SSH y el terminal de nuestro dispositivo a nuestro servidor de PiensaSolutions, en el panel de configuración podemos ver la IP (“87.106.237.199”) y un campo para mostrar una contraseña por defecto. (16)

Figura 35. Datos de acceso al servidor

Una vez tengamos los datos de acceso, procedemos a abrir el terminal y acceder al servidor por medio del comando “ssh” de la siguiente forma.

```
raulfdez@MacBook-Air-de-Raul ~ % ssh root@87.106.237.199
(root@87.106.237.199) Password:
```

Figura 36. Comando de acceso al servidor

Una vez estemos dentro del servidor, el primer paso dentro será actualizar los repositorios, los programas y las extensiones del servidor para trabajar con el servidor lo más actualizado posible para evitar errores futuros.

```
root@localhost:~# apt update && apt upgrade
```

Figura 37. Comando actualización del servidor

Proseguimos con la instalación de Apache2, que servirá de nuestro servidor web y MySQL Server, para gestionar nuestra base de datos.

```
root@localhost:~# apt install apache2
```

Figura 38. Comando instalación Apache2

```
root@localhost:~# apt install mysql-server
```

Figura 39. Comando instalación MySQL Server

Comprobamos que se han instalado correctamente y los servicios están activos.

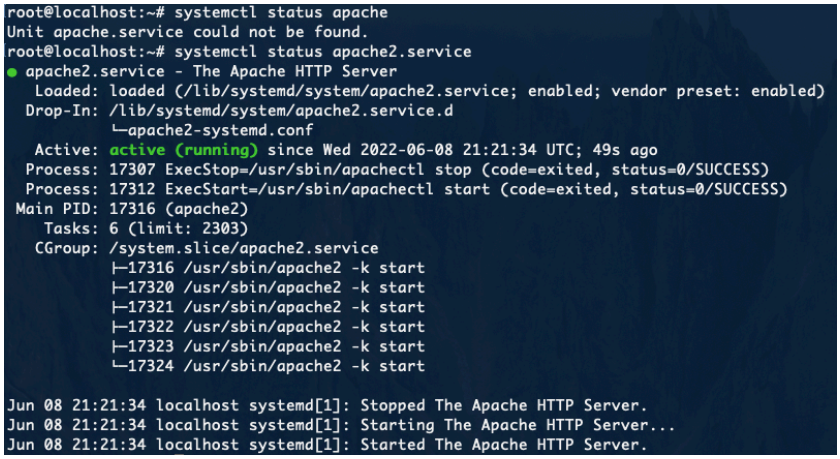


Figura 40. Comprobación estado Apache2

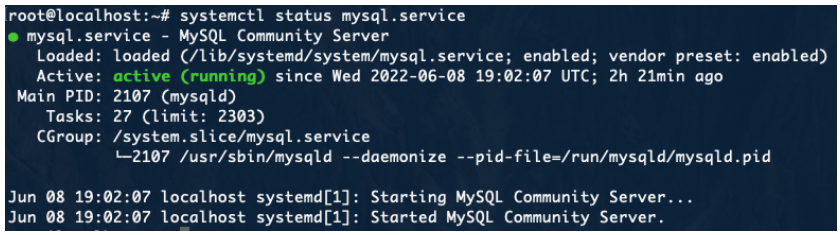


Figura 41. Comprobación del estado de MySQL

Ahora vamos a instalar PHP, es un requisito indispensable para Magento, aunque como vamos a instalar la versión 2.4.0 de Magento no podemos instalar la última versión de PHP, requiere la instalación de PHP 7.4, para ello tendremos que añadir un repositorio personal (PPA) donde se encuentra esta versión de PHP y lo haremos de la siguiente forma instalando también las extensiones de PHP que requiere Magento.

Figura 42. Comando instalación paquete para gestionar repositorios

Figura 43. Comando para añadir repositorio personal (PPA)

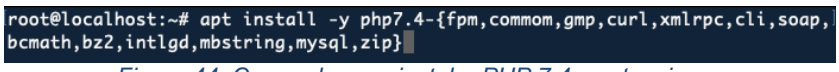


Figura 44. Comando para instalar PHP 7.4 y extensiones

Una vez instalado PHP, comprobaremos su versión para mostrar que está instalado correctamente con la versión 7,4

```
root@localhost:~# php -v
PHP 7.4.29 (cli) (built: Apr 21 2022 10:16:17) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.29, Copyright (c), by Zend Technologies
```

Figura 45. Comprobación de la versión de PHP

Antes de proseguir con la configuración e instalación de paquetes necesarios vamos a configurar el dominio de nuestra cliente para que apunte a nuestro servidor.

Para ellos nos vamos al panel de administración del dominio perteneciente a IONOS, por lo que realizaremos login en IONOS con los datos de acceso proporcionados por nuestra cliente e iremos a la configuración DNS del dominio, tenemos que comprobar que no existe ya un registro A, ya que este es el que determina qué dirección IPv4 se asigna al dominio.

Una vez comprobado este paso tenemos que crear un registro A que apunte hacia la dirección IP de nuestro servidor, para ello tenemos que realizar la siguiente configuración. (17)

Figura 46. Configuración registro A del dominio

Ahora continuamos con la configuración del certificado que adquirimos anteriormente y validamos, para ello tenemos el certificado descargado de DonDominio que viene los siguientes ficheros.

- Certificate.ca.crt → La autoridad de certificación (CA), es una empresa u organización que actúa para validar las identidades de las entidades (como sitios web, direcciones de correo electrónico, empresas o personas individuales) y vincularlas a claves criptográficas mediante la emisión de documentos electrónicos conocidos como Certificados digitales.

- Certificate.crt → Este fichero contiene nuestro certificado SSL, contiene el nombre del dominio, y acredita que efectivamente se está comunicando con quien dice ser.

- Certificate.key → Este fichero contiene la clave privada, digamos que la clave privada funciona como un sello, y con ella se acredita al servidor.

Una vez con estos ficheros, nos conectaremos a nuestro servidor a través del software Filezilla, este programa nos ayudara en la transmisión de ficheros con nuestro servidor, nos conectamos a nuestro servidor por medio del protocolo SFTP (SSH File Transfer Protocol).

Antes de subir los ficheros debemos hacer una pequeña modificación y es concatenar en un mismo fichero .crt el certificado nuestro (Certficate.crt) y el certificado CA, no es obligatorio hacerlo de esta forma. Los ficheros los llamaríamos “apache-certificate.crt” (como hemos dicho contiene nuestro certificado y el certificado CA) y “apache.key” que contiene la clave privada de nuestro certificado.

Lo siguiente será subir los ficheros al servidor a una carpeta que hemos creado llamada “certificate” dentro de la ruta “/etc/apache2” de nuestro servidor, por lo que quedaría así.

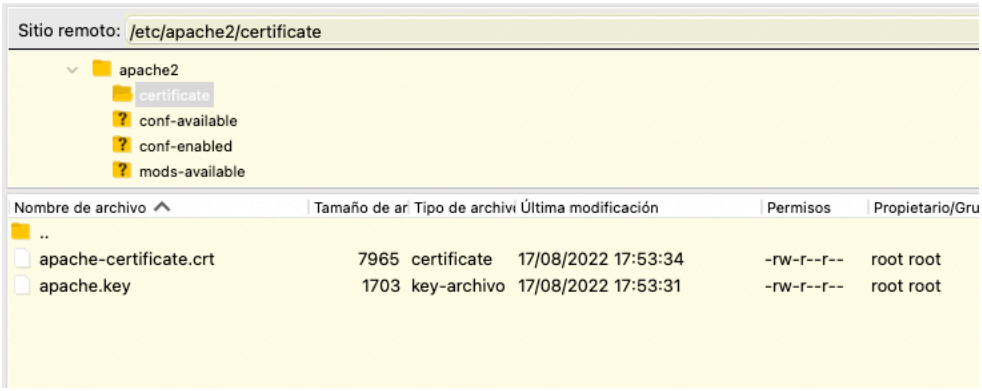


Figura 47. Ficheros del certificado en el servidor

Con los ficheros subidos al servidor ahora tenemos que seguir los siguientes pasos, primero vamos a activar el módulo SSL con los siguientes comandos.

```
root@localhost:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
```

Figura 48. Comando activación módulo SSL

```
root@localhost:~# systemctl reload apache2
```

Figura 49. Comando recargar el servidor Apache2

Ahora creamos el fichero de configuración a través del software Visual Studio Code, el cual mediante una extensión nos conectaremos a nuestro servidor desde el propio VSCode y crearemos el fichero en la ruta "/etc/apache2/sites-available", lo llamaremos "thenuconcept.conf" y tendrá la siguiente configuración.

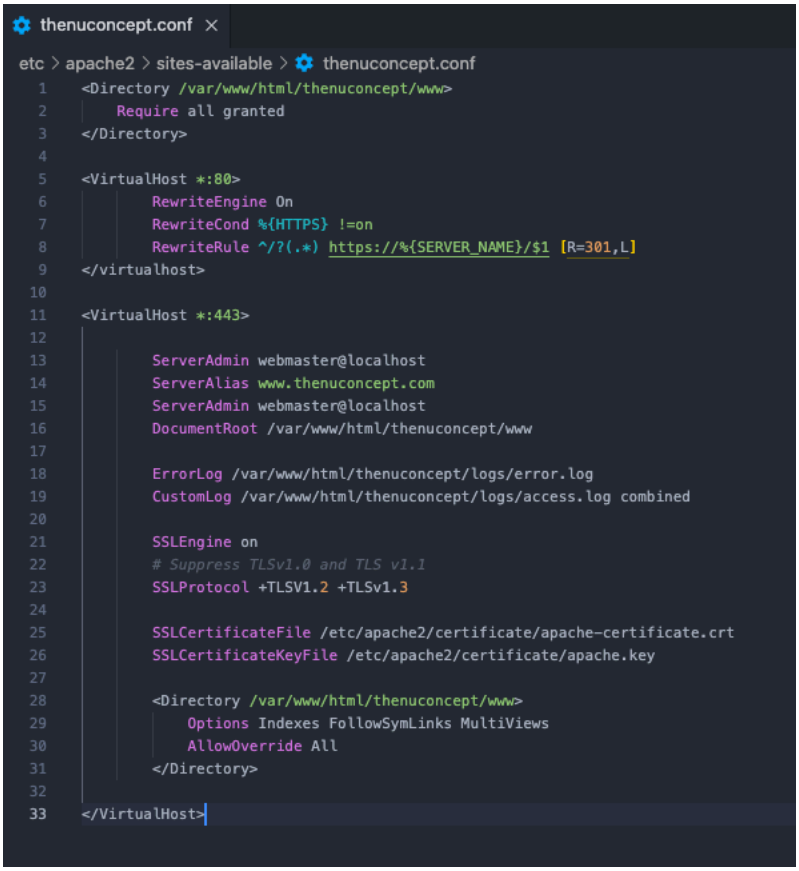


Figura 50. Fichero de configuración del certificado en el servidor

En este fichero mostrado en la Figura 27, podemos observar la configuración necesaria para los puertos accesibles desde el navegador (80:HTTP y 443:HTTPS), destacamos la redirección creada dentro del puerto 80, para que si se intenta entrar por el puerto HTTP redirija automáticamente al puerto HTTPS. Luego en la configuración del puerto 443 está configuración del dominio y alias, la ruta donde se encuentra nuestro proyecto y la configuración para nuestro certificado SSL especificando las rutas del certificado, también destacamos la supresión del acceso mediante los protocolos TLSv1.0 y TLSv.1.1, ya que son antiguos e inseguros dejando así los protocolos TLSv1.2 y TLSv1.3.

Ahora debemos activar el host virtual SSL y reiniciar el servidor Apache para que se aplique el cambio.

Figura 51. Comando activación host virtual SSL

```
root@localhost:~# systemctl reload apache2
```

Figura 52. Comando reinicio servidor Apache2

Para comprobar que el certificado ha sido instalado correctamente, aparte de comprobar que el sitio es seguro y se muestra el certificado instalado, hemos realizado un test SSL a través de la página SSL Labs

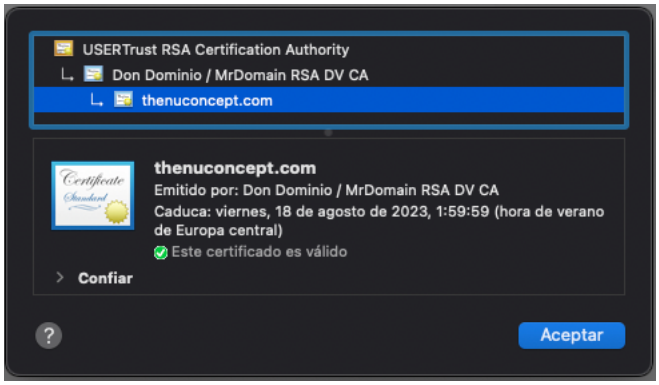


Figura 53. Comprobación de instalación de certificado

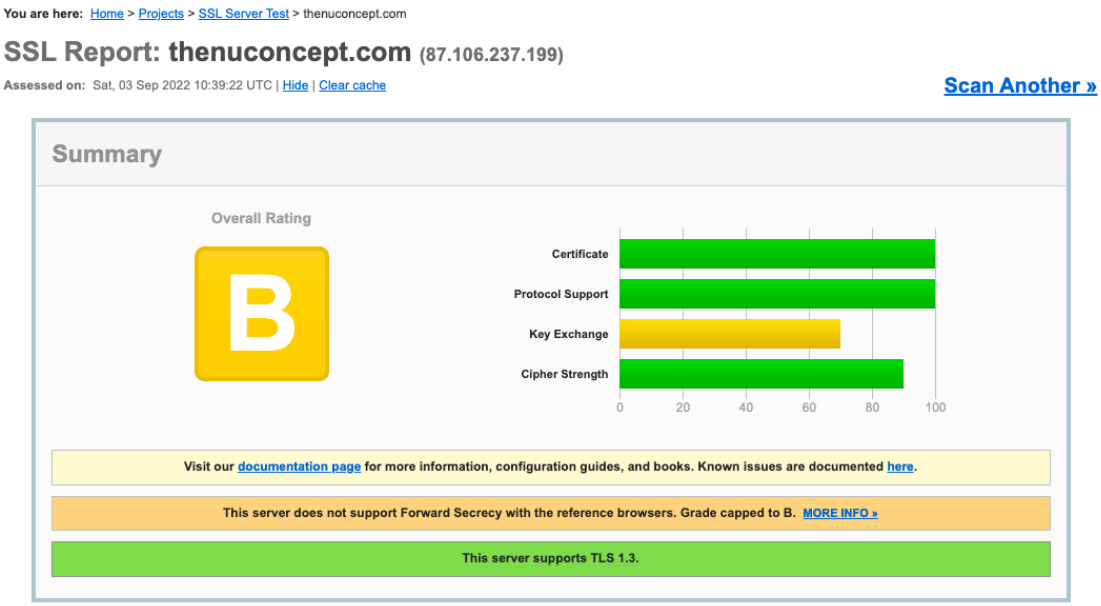
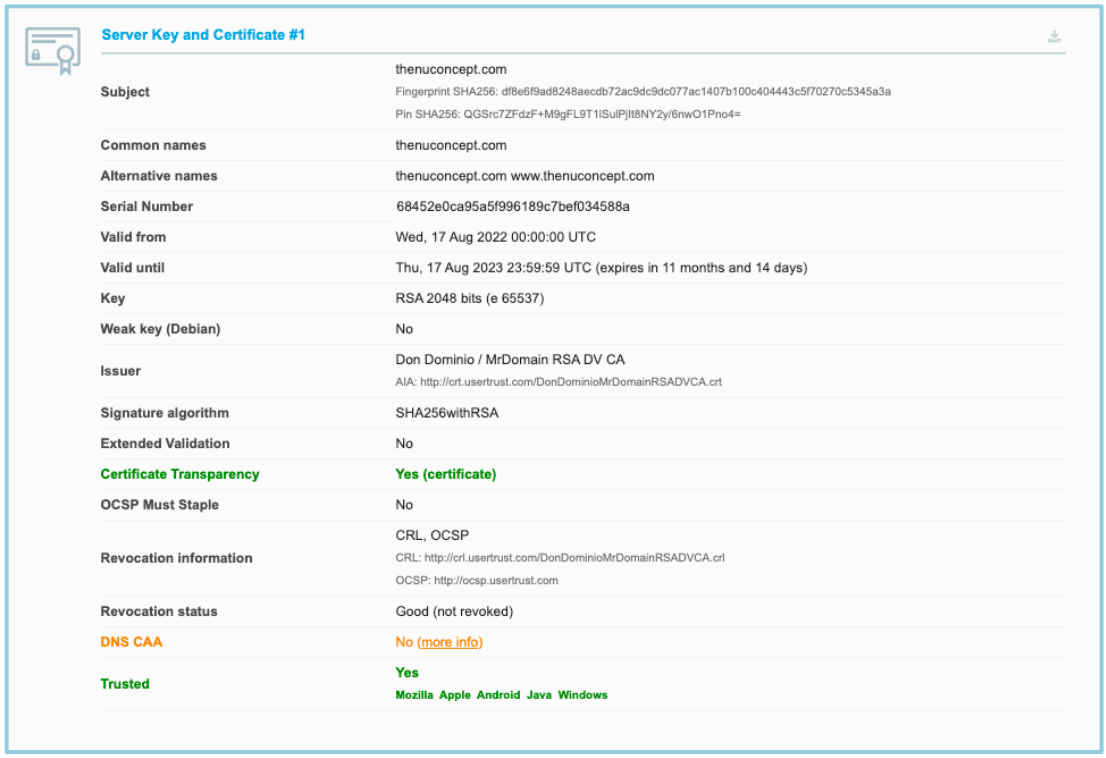


Figura 54. Evaluación del test SSL de SSL Labs



Server Key and Certificate #1

Subject	thenuconcept.com Fingerprint SHA256: df8e6f9ad8248aecdb72ac9dc9dc077ac1407b100c404443c5f70270c5345a3a Pin SHA256: QGSrc7ZFdzF+M9gFL9T1ISulPJlt8NY2y/6nwO1Pno4=
Common names	thenuconcept.com
Alternative names	thenuconcept.com www.thenuconcept.com
Serial Number	68452e0ca95a5f996189c7bef034588a
Valid from	Wed, 17 Aug 2022 00:00:00 UTC
Valid until	Thu, 17 Aug 2023 23:59:59 UTC (expires in 11 months and 14 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	Don Dominio / MrDomain RSA DV CA AIA: http://crt.usertrust.com/DonDominioMrDomainRSADVCA.crt
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	CRL, OCSP CRL: http://crl.usertrust.com/DonDominioMrDomainRSADVCA.crl OCSP: http://ocsp.usertrust.com
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows

Figura 55. Detalles del certificado instalado, SSL Labs

En las figuras 30 y 31 podemos observar que el certificado está instalado correctamente y que la configuración es correcta, en cuanto a certificado, protocolo y cifrado tenemos una buena evaluación, y un poco más bajo en Key Exchange debido a que no cumplimos con un sistema que tiene secreto hacía delante, un problema menor que se tendría que resolver en un futuro cercano, por lo que podemos seguir con la instalación. (18)

Continuamos instalando Elasticsearch, como requisito para instalar y actualizar Magento 2.4.x. Elasticsearch es un motor de base de datos muy potente, capaz de indexar, ordenar y responder con rapidez a consultas sobre millones de datos. Utilizaremos “wget” para obtener el paquete de instalación de Elasticsearch y lo instalaremos.

```
root@localhost:/opt# wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.1-amd64.deb
```

Figura 56. Comando para obtener el paquete de Elasticsearch

```
root@localhost:/opt# dpkg -i elasticsearch-7.6.1-amd64.deb
```

Figura 57. Comando instalación Elasticsearch

```
root@localhost:/opt# systemctl start elasticsearch.service
root@localhost:/opt# systemctl status elasticsearch.service
● elasticsearch.service - Elasticsearch
   Loaded: loaded (/usr/lib/systemd/system/elasticsearch.service; disabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-06-08 21:43:36 UTC; 28s ago
     Docs: http://www.elastic.co
 Main PID: 24881 (java)
    Tasks: 61 (limit: 2303)
   CGroup: /system.slice/elasticsearch.service
           ├─24881 /usr/share/elasticsearch/jdk/bin/java -Des.networkaddress.cache.ttl=60 -Des.networkaddress.cache.negative.ttl=10
           └─24959 /usr/share/elasticsearch/modules/x-pack-ml/platform/linux-x86_64/bin/controller

Jun 08 21:42:59 localhost systemd[1]: Starting Elasticsearch...
Jun 08 21:43:00 localhost elasticsearch[24881]: OpenJDK 64-Bit Server VM warning: Option UseConcMarkSweepGC was deprecated in versio
Jun 08 21:43:36 localhost systemd[1]: Started Elasticsearch.
lines 1-13/13 (END)
```

Figura 58. Comando inicio y mostrar estado del servicio de Elasticsearch

Llegados a este punto tenemos instalado todos los componentes para la instalación de Magento, pero antes de descargar e instalar Magento debemos configurar algunos componentes para una instalación correcta.

Empezaremos por configurar la base de datos, para ello debemos entrar como usuario “root” a través del terminar y crear una base de datos para el proyecto que llamaremos “thenuconcept” y crearemos un primer usuario llamado “soporte” con una contraseña que posteriormente cambiaremos para tener mayor seguridad, a este usuario le daremos todos los permisos sobre la base de datos creada.

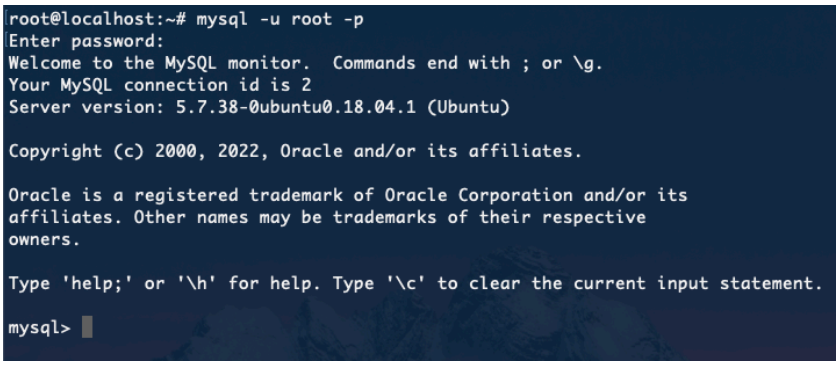


Figura 59. Comando acceso a la consola de la base de datos

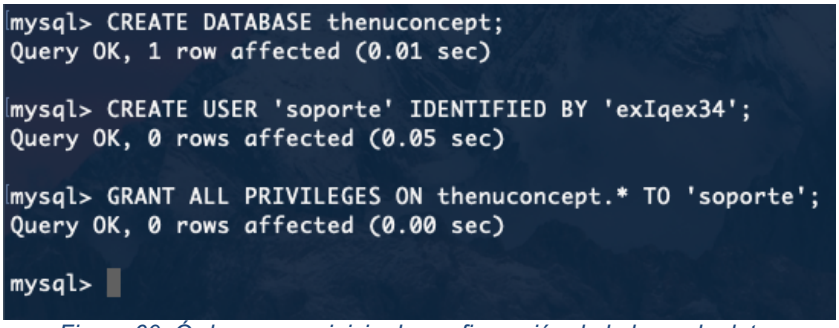


Figura 60. Órdenes para iniciar la configuración de la base de datos

Una vez configurada la base de datos tenemos que hacer algunas modificaciones en el fichero de configuración de PHP, editaremos los ficheros “/etc/php/7.4/cli/php.ini” y “/etc/php/7.4/apache2/php.ini” modificando los parámetros “memory_limit” y “date.timezone”, poniendo el primer valor con una memoria de “2G” y el segundo valor “Europe/Madrid”, con esto aumentaremos el límite de memoria PHP de memoria que cualquier script de tu sitio web que se puede utilizar, y con el segundo valor establecemos la zona de la hora donde se encuentra nuestro servidor para que las funciones de php de tiempo tengan por defecto la configuración de tiempo de nuestra situación.

- Una vez tengamos todo configurado y las extensiones requeridas instaladas pasaremos a la segunda etapa en la que procederemos a instalar Magento Open Source eCommerce 2.4.0.

Para ello primero crearemos los directorios donde tendremos alojado el CMS de Magento y una carpeta para guardar el log de errores.

```
root@localhost:~# mkdir -p /var/www/html/thenuconcept/{logs,www}
```

Figura 61. Comando creación de directorios base

Una vez creados los directorios, nos iremos al directorio "/var/www/html/thenuconcept/www" y clonaremos el repositorio de Magento Open Source. Para esto es necesario tener instalado "git" en el servidor.

```
root@localhost:/var/www/html/thenuconcept/www# git clone https://github.com/magento/magento2.git
```

Figura 62. Comando para clonar el repositorio de Magento Open Source eCommerce 2.4.0

Antes de instalar Magento con Composer tenemos que configurar un usuario "thenuconceptuser" para los ficheros del CMS y añadirlo al grupo de Apache ("www-data").

```
root@localhost:~# useradd thenuconceptuser
```

Figura 63. Comando para añadir un usuario

```
root@localhost:~# usermod -g www-data thenuconceptuser
```

Figura 64. Comando para añadir el usuario al grupo de Apache

Una vez creado el usuario y añadido al grupo, vamos a configurar los permisos de la siguiente manera.

```
/thenuconcept/www# find var vendor pub/static pub/media app/etc -type f -exec chmod g+w {} \;
```

Figura 65. Comando para dar permisos a las carpetas de static, media y etc

```
/thenuconcept/www# find var vendor pub/static pub/media app/etc -type d -exec chmod g+ws {} \;
```

Figura 66. Comando para dar permisos a las carpetas de static, media y etc

```
/thenuconcept/www# chown -R thenuconceptuser:www-data ../../thenuconcept
```

Figura 67. Comando para cambiar el propietario del directorio

```
root@localhost:/var/www/html/thenuconcept/www# chmod u+x bin/magento
```

Figura 68. Comando para dar permisos de ejecución a bin/magento

Después de configurar los permisos necesarios vamos a instalar Composer, herramienta que utilizaremos para instalar tanto Magento Open Source eCommerce 2.4.0 como para instalar extensiones y temas para el CMS.

```
root@localhost:/var/www/html/thenuconcept/www# curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/bin --filename=composer
All settings correct for using Composer
Downloading...

Composer (version 2.3.7) successfully installed to: /usr/bin/composer
Use it: php /usr/bin/composer

root@localhost:/var/www/html/thenuconcept/www# # composer install
```

Figura 69. Comando instalación de Composer

Para instalar Magento vamos a utilizar el comando "bin/magento setup:install" desde el directorio del proyecto con las siguientes variables de configuración:

- --base-url=https://thenuconcept.com → URL base del sitio web
- --db-host = localhost → Host de la base de datos
- --db-name = thenuconcept → Nombre de la base de datos
- --db-user = soporte → Usuario de la base de datos
- --db-password = exIqex34 → Contraseña del usuario
- --admin-firstname = Soporte → Nombre del usuario del panel de administración
- --admin-lastname = Thenuconcept → Apellidos para el usuario del panel de administración
- --admin-email = soporte@thenuconcept.com → Email del usuario del panel de administración
- --admin-user = soporte → Usuario del panel de administración
- --admin-password = exIqex34 → Contraseña del usuario del panel de administración
- --language=es_ES → Idioma del sitio web
- --currency = EUR → Moneda por defecto para la tienda
- --timezone=Europe/Madrid → Zona horaria del servidor

```
root@localhost:/var/www/html/thenuconcept/www# bin/magento setup:install --base-url=https://thenuconcept.com
--db-host=localhost --db-name=thenuconcept --db-user=soporte --db-password=exIqex34 --admin-firstname=Soporte
 --admin-lastname=Thenuconcept --admin-email=soporte@thenuconcept.com --admin-user=soporte --admin-password=e
xIqex34 --language=es_ES --currency=EUR --timezone=Europe/Madrid --use-rewrites=1
```

Figura 70. Comando instalación Magento 2.4.0

```
Starting Magento installation:
File permissions check...
[Progress: 1 / 867]
Required extensions check...
[Progress: 2 / 867]
Enabling Maintenance Mode...
[Progress: 3 / 867]
```

Figura 71. Inicio de proceso de instalación de Magento 2.4.0

```
[Progress: 861 / 863]
Post installation file permissions check...
For security, remove write permissions from these direc
[Progress: 862 / 863]
Write installation date...
[Progress: 863 / 863]
[SUCCESS]: Magento installation complete.
[SUCCESS]: Magento Admin URI: /admin_16bdbv
Nothing to import.
```

Figura 72. Fin instalación Magento 2.4.0

Ha terminado con éxito el proceso de instalación de Magento, la salida del proceso nos dice cuál será el URI de administración que tendremos que añadir a nuestra url para entrar al panel de administración, en este caso es **"/admin_16bdbv"**.

En la instalación pueden ocurrir distintos errores, normalmente suelen ser por extensiones de PHP que faltan o extensiones que no están instaladas

o bien configuradas, aunque haya terminado con éxito no hemos terminado aun, hay que realizar unos ajustes de configuración más.

Procedemos a poner como propietario del directorio base de nuestro sitio web al usuario de Apache (www-data)

```
root@localhost:~# chown -R www-data:www-data /var/www/html/thenuconcept/
```

Figura 73. Comando para asignar propietario www-data al directorio base

Al entrar en la dirección de nuestro sitio web https://thenuconcept.com nos aparece esta página por defecto de Magento con el tema “Luma” por defecto.

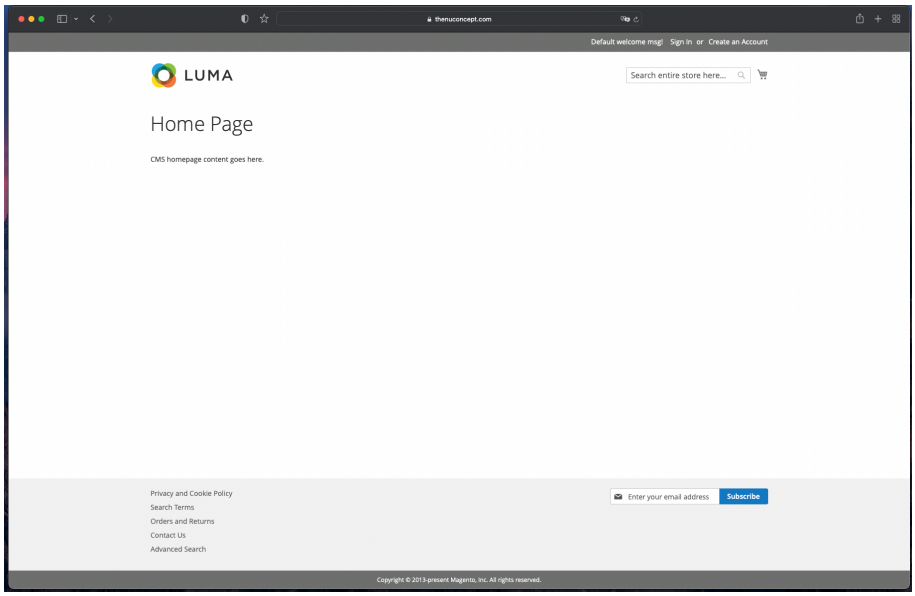


Figura 74. Página por defecto de Magento

Y si accedemos a la dirección de nuestro sitio con el URI generado en la instalación anteriormente mencionado entraríamos en el panel de administración. La dirección quedaría https://thenuconcept.com/admin_16bdbv y nos mostraría la siguiente pantalla de login (Figura 51) la cual accederíamos a la segunda pantalla (Figura 52) con la configuración establecida en el comando de instalación.

- Username: soporte
- Password: exIqex34

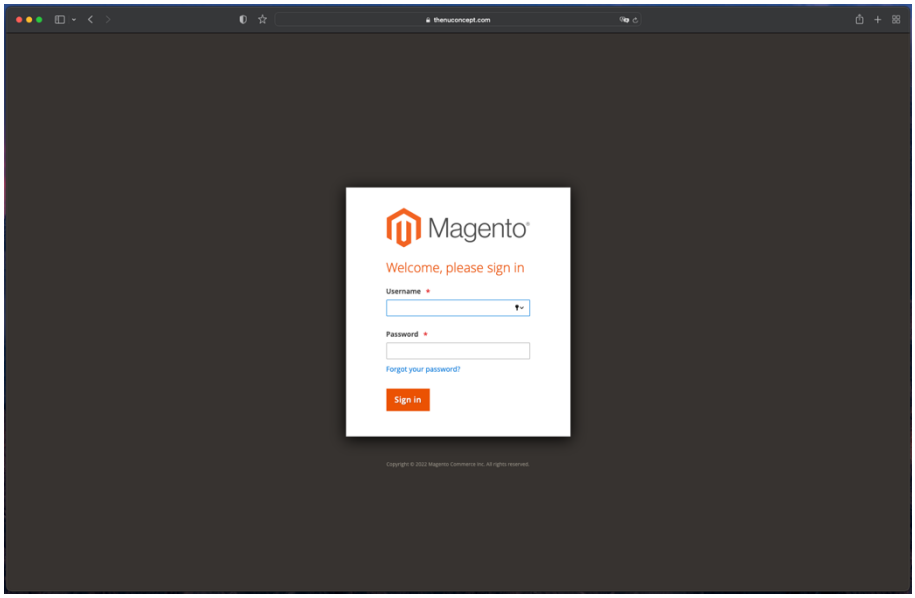


Figura 75. Página por defecto del panel de administración 1

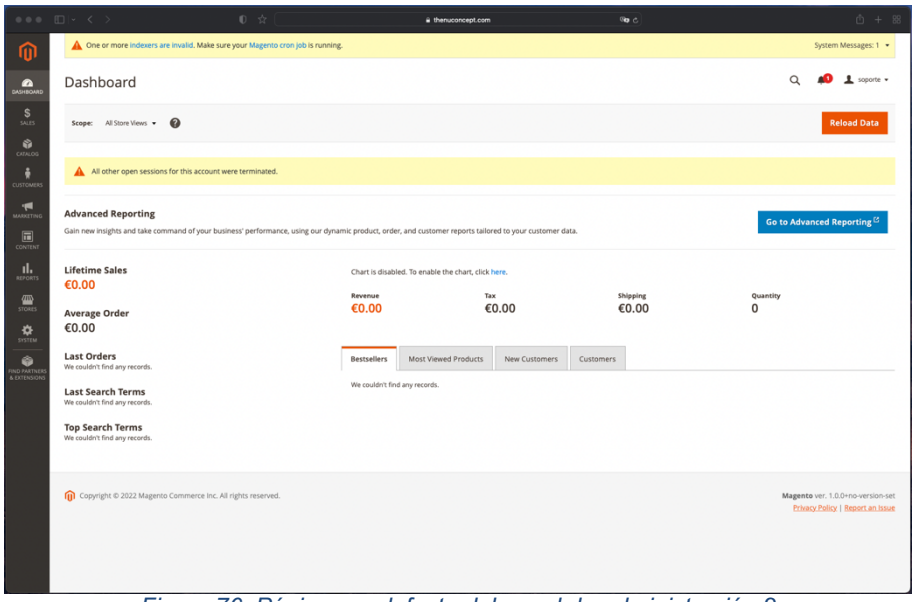


Figura 76. Página por defecto del panel de administración 2

Si nos fijamos en la Figura 52, nos damos cuenta de un aviso que se muestra en la parte de arriba debajo de la dirección de la página y muestra el mensaje de **"One or more indexers are invalid. Make sure your Magento cron job is running"**, este mensaje nos indica que uno o más índices son inválidos, porque varias funciones de Magento eCommerce requieren al menos un trabajo cron, que programa las actividades para que se produzcan en el futuro.

Para solucionar este aviso tendremos que ejecutar unos procesos repetitivos por medio de "crontab", que es un administrador regular de procesos en segundo plano que ejecuta procesos o guiones a intervalos regulares. Por lo que para el correcto funcionamiento tendremos que añadir al "crontab" las siguientes líneas.

```
* * * * * /usr/bin/php
/var/www/html/thenuconcept/www/bin/magento cron:run | grep
-v "Ran jobs by schedule" >>
/var/www/html/thenuconcept/www/var/log/magento.cron.log
```

```
* * * * * /usr/bin/php
/var/www/html/thenuconcept/www/update/cron.php >>
/var/www/html/thenuconcept/www/var/log/update.cron.log
```

```
* * * * * /usr/bin/php
/var/www/html/thenuconcept/www/bin/magento setup:cron:run
>> /var/www/html/thenuconcept/www/var/log/setup.cron.log
```

Con la primera línea ejecutamos el trabajo cron de indexación. La segunda y tercera línea son necesarias para el Administrador de componentes y actualización del sistema.

Con esto ya habremos terminado la instalación de Magento Open Source eCommerce 2.4.0 y tendremos accesible nuestro sitio web y nuestro panel de administración para empezar a configurar nuestra tienda online.

Por último, debemos saber los comandos básicos para lanzar el sitio web y realizar el mantenimiento de este.

- `php bin/magento setup:upgrade` → Actualiza la configuración
- `php bin/magento setup:upgrade –keep-generated` → Actualiza la configuración conservando los ficheros estáticos
- `php bin/magento setup:di:compile` → Ejecuta el compilador del sitio
- `php bin/magento setup:static-content:deploy -f` → Despliegue con el lenguaje “en_US” por defecto, con “-f” forzamos el despliegue
- `php bin/magento cache:clean` → Borra la cache por etiquetas
- `php bin/magento cache:flush` → Borra la caché completamente
- `php bin/magento cache:enable/disable` → Habilita o deshabilita la caché
- `php bin/magento indexer:status` → Comprueba el estado de los índices de búsqueda
- `php bin/magento indexer:show-mode` → Muestra el estado de los índices
- `php bin/magento module:status` → Comprueba el estado de los módulos (Activos y no)
- `php bin/magento module:enable/disable “Nombre_modulo”` → Habilitar o deshabilita un módulo. “`Nombre_modulo`” es el nombre del mismo como aparece en `module:status` (anterior)
- `php bin/magento module:disable “Nombre_modulo”` → Deshabilitar un módulo
- `php bin/magento module:uninstall “Nombre_modulo”` → Desinstalar un módulo

- `php bin/magento deploy:mode:show` → Consulta el modo de funcionamiento de Magento activo
- `php bin/magento deploy:mode:set developer` → Activa el modo desarrollo
- `php bin/magento deploy:mode:set production` → Activa el modo producción
- `php bin/magento maintenance:enable/disable` → Activa el modo mantenimiento
- `php bin/magento maintenance:status` → Estado del modo mantenimiento.
- `php bin/magento admin:user:unlock adminusername` → Desbloquear un usuario administrador.

3.5 Implementación y desarrollo

En esta sección del capítulo vamos a configurar y personalizar nuestra tienda online.

Para empezar vamos a instalar el paquete de idioma de español para el panel de administración, para ello debemos seguir los siguientes pasos desde el terminal mediante conexión SSH e instalando el pack de español desde Composer.

```
root@localhost:~# composer require mageplaza/magento-2-spanish-language-pack:dev-master mageplaza/module-smtp &&
> php bin/magento setup:static-content:deploy es_ES &&
> php bin/magento indexer:reindex &&
> php bin/magento cache:clean &&
> php bin/magento cache:flush &&
```

Figura 77. Instalación paquete español Magento 2

Con estos comandos instalaremos el pack de lenguaje para Magento 2 de español, desplegaremos el proyecto con el idioma de español, reindexaremos y limpiaremos la cache del sitio web.

Una vez hecho esto tenemos que cambiar el idioma en dos sitios del panel de administración.

La primera modificación la haremos desde Tiendas → Configuración →General Cambiando la opción de “Configuración regional” a español (España)

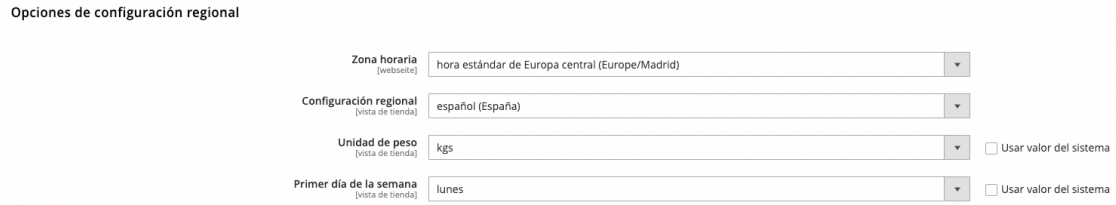


Figura 78. Cambio de idioma de la tienda online

Esta modificación debería haber sido automática después de desplegar el proyecto bajo el lenguaje por defecto de español, pero por si acaso la haremos.

La siguiente modificación la haremos para cambiar el idioma del panel de administración, para ello tendremos que ir a la configuración de la cuenta de administrador situado en la esquina superior derecha en un selector desplegable. Una vez allí modificamos el campo de "Interfaz de configuración regional" y ponemos el idioma de español.

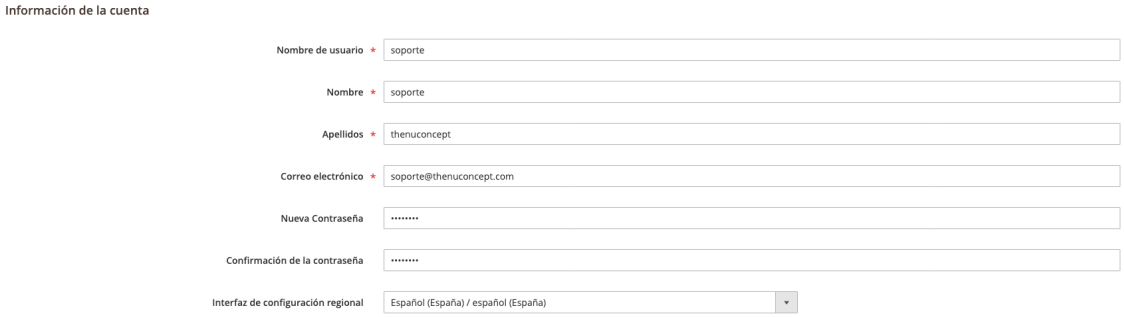


Figura 79. Cambio de idioma del panel de administración

Con esto ya hemos cambiado el idioma, el paso siguiente será realizar una configuración inicial del sitio con los datos de la tienda. Modificaremos los siguientes campos de configuración:

- Tiendas → Configuración → General:

Figura 80. Ajuste inicial de Magento 1

Figura 81. Ajuste inicial de Magento 2

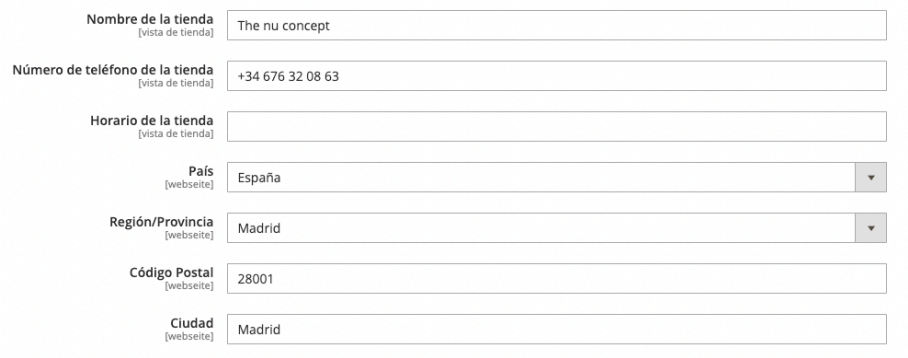


Figura 82. Ajuste inicial de Magento 3

Una vez hecho unos ajustes iniciales vamos a instalar un tema para cambiar el diseño de nuestro sitio web, nuestro diseño elegido será "**Breeze Evolution**", se trata de un tema simple y sencillo que proporcionará a la tienda una mejor experiencia de usuario.

Para ello seguiremos los siguientes pasos.

Primero instalaremos el módulo de Marketplace para ejecutar el instalador de temas después.

```
root@localhost:~# composer require swissup/module-marketplace &&\
> bin/magento setup:upgrade --safe-mode=1
```

Figura 83. Pre-instalación tema Breeze Evolution

Seguido de instalar el módulo de Marketplace instalaremos el tema "Breeze Evolution"

```
root@localhost:~# composer require swissup/breeze-evolution &&\
> bin/magento setup:upgrade --safe-mode=1 &&\
> bin/magento marketplace:package:install swissup/breeze-evolution
```

Figura 84. Instalación tema Breeze Evolution

Para seleccionar el tema en nuestro sitio web realizaremos la siguiente modificación en nuestro panel de administración en Contenido → Configuración, seleccionamos nuestro sitio, que tendremos dos ya que nuestra tienda online es multi idioma y tendremos "ES" y "EN", habría que cambiar el tema en las dos opciones de forma que una vez editado quede así.

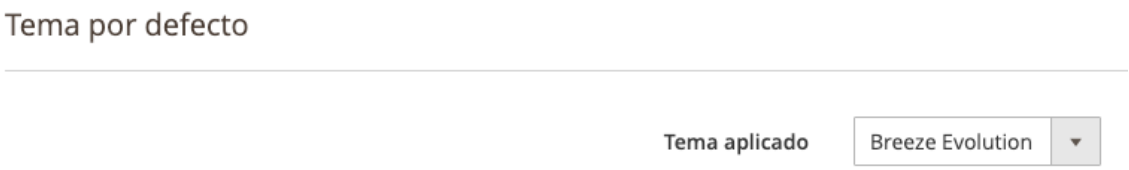
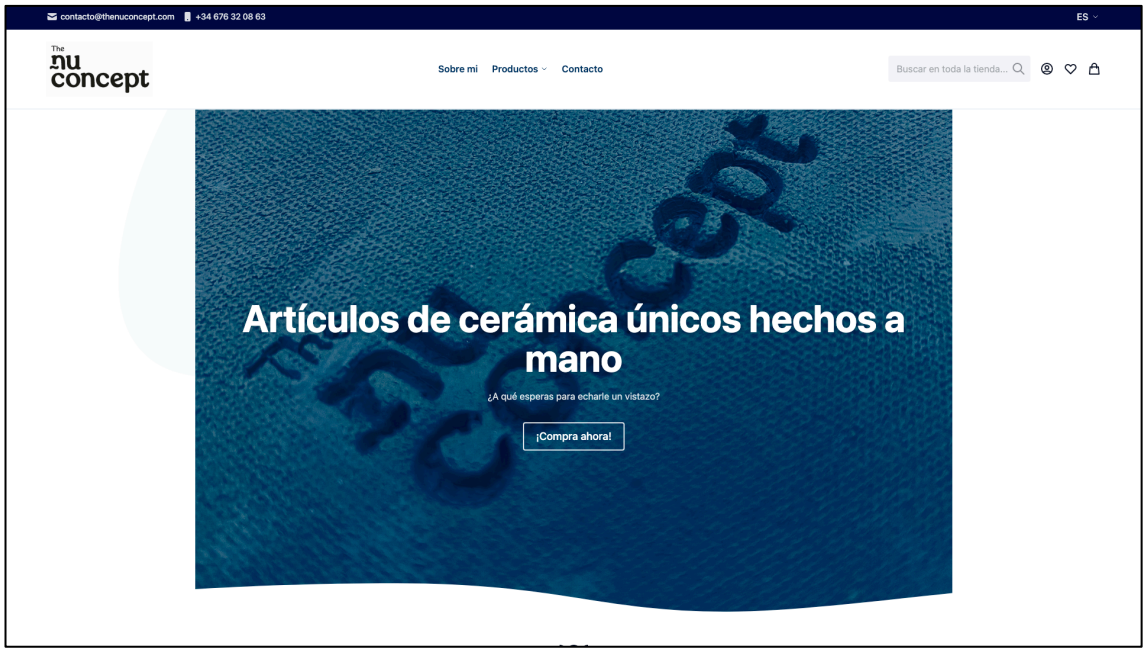


Figura 85. Aplicación del tema

Una vez aplicado el tema nos encontramos con varios errores de diseño, como por ejemplo que la imagen puesta en el banner de la pagina inicial no ocupa el ancho correctamente.



Para arreglar esto, nos hemos encontrado que un estilo de CSS limita el ancho de la imagen, por lo que hemos creado una estilo CSS asociado al id del div “maincontent” que permita un ancho de la imagen como máximo de 2400px, ya que la imagen de fondo tiene esta dimensión.

```css
.bg-image {
    background-image: url(https://i.ibb.co/njF5fK1/bg-header.png);
    background-repeat: no-repeat;
    background-size: auto;
}
```

Figura 86. Estilo CSS de la imagen de fondo en la página de inicio

```css
#maincontent {
    max-width: 2400px !important;
    padding: 0px;
}
```

Figura 87. Estilo CSS bug ancho de la imagen de fondo

Además, a la imagen de fondo en su clase CSS le hemos añadido el atributo “background-attachment: fixed” para añadir un efecto de profundidad en la página y a su vez hacer una apariencia más moderna.

```
#html-body [data-pb-style=DWTQMJT],
#html-body [data-pb-style=XQLN147] {
    background-size: cover;
    background-repeat: no-repeat;
    background-attachment: fixed;
}
```

Figura 88. Modificación de estilo CSS en la imagen de fondo

Una vez hecho estas modificaciones la página de inicio se muestra de esta forma.

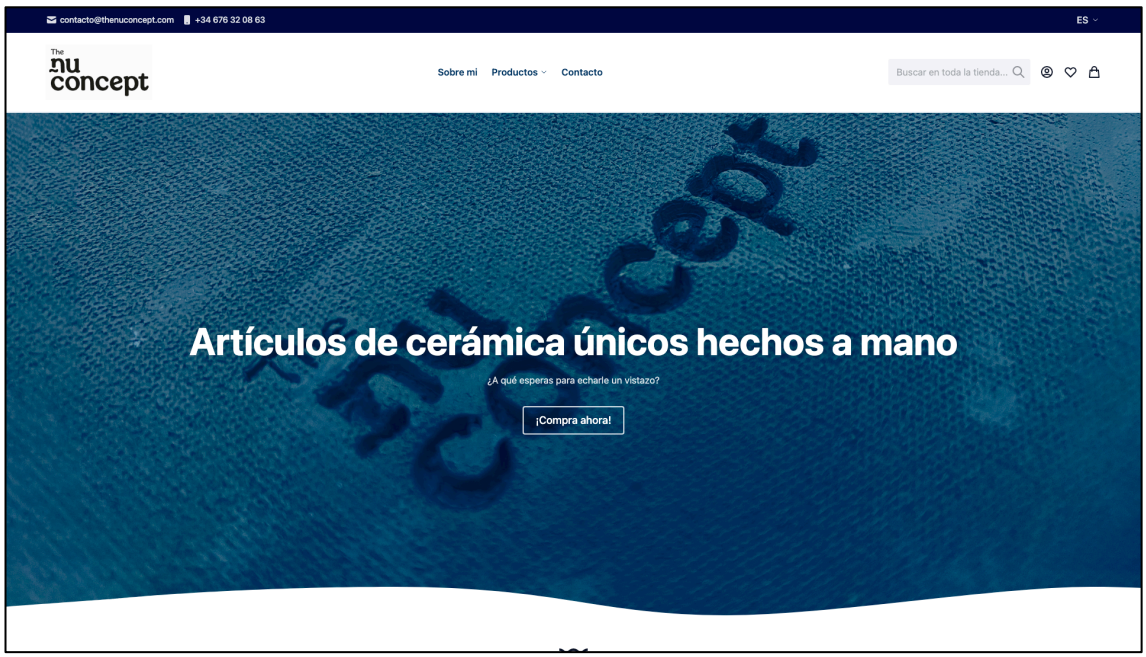
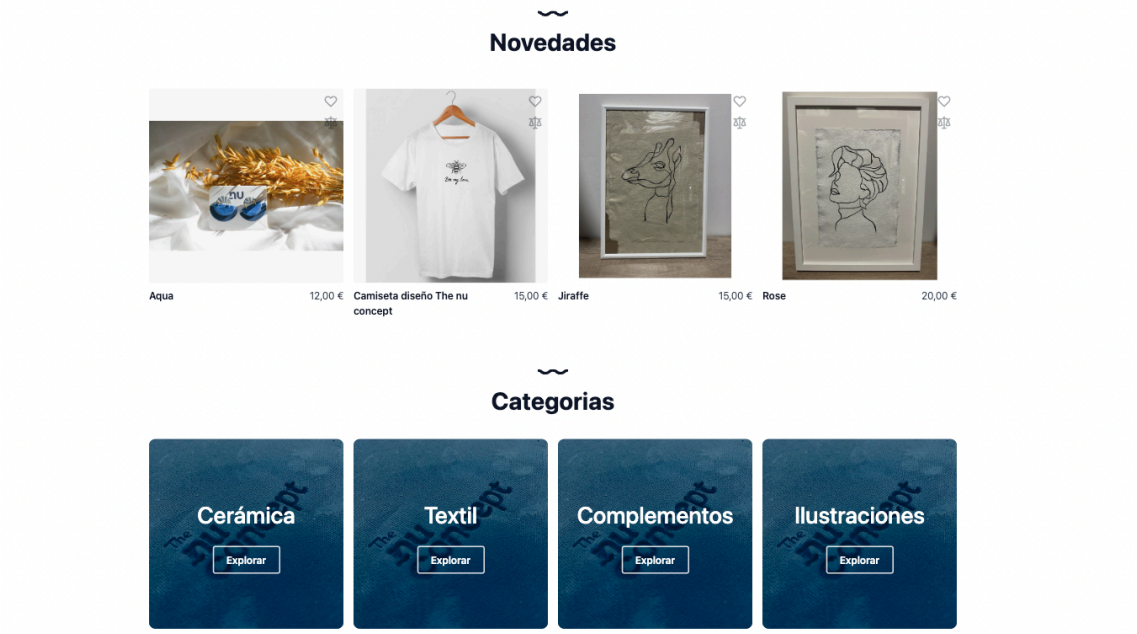


Figura 89. Página inicio con imagen arreglada

Además de arreglar la imagen de fondo, hemos añadido un bloque de “Novedades” donde se mostrarán los últimos artículos añadido y una fila con enlaces a las categorías existentes estiladas mediante CSS.



A parte de realizar ajustes de diseño se ha personalizado la página, a partir del panel de administración de Magento 2 se ha introducido distintos códigos HTML y CSS para personalizar el tema en distintos puntos.

Uno de ellos es el footer, es un bloque que se carga en el tema de Breeze Evolution y que se ha modificado el código HTML para que muestre los datos de nuestra tienda, además de realizar un bloque footer para el idioma de español y otro para el idioma de inglés.

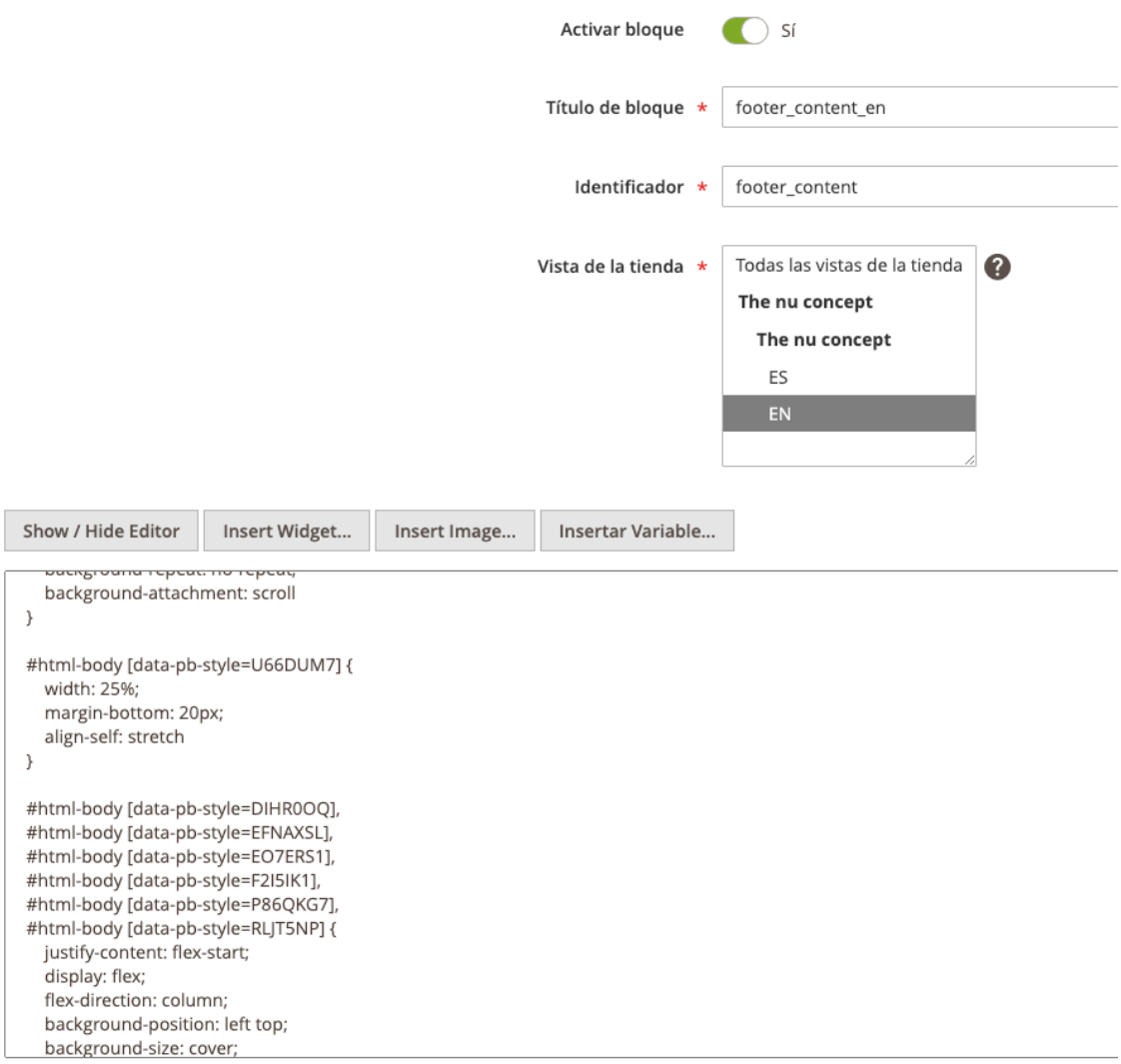


Figura 90. Inserción código footer

Figura 91. Footer personalizado

Seguimos con tareas de personalización, hemos creado un diseño personalizado con la marca para cuando un usuario entra en una página inexistente o que un link lleva a una página con error 404 de página no encontrada.

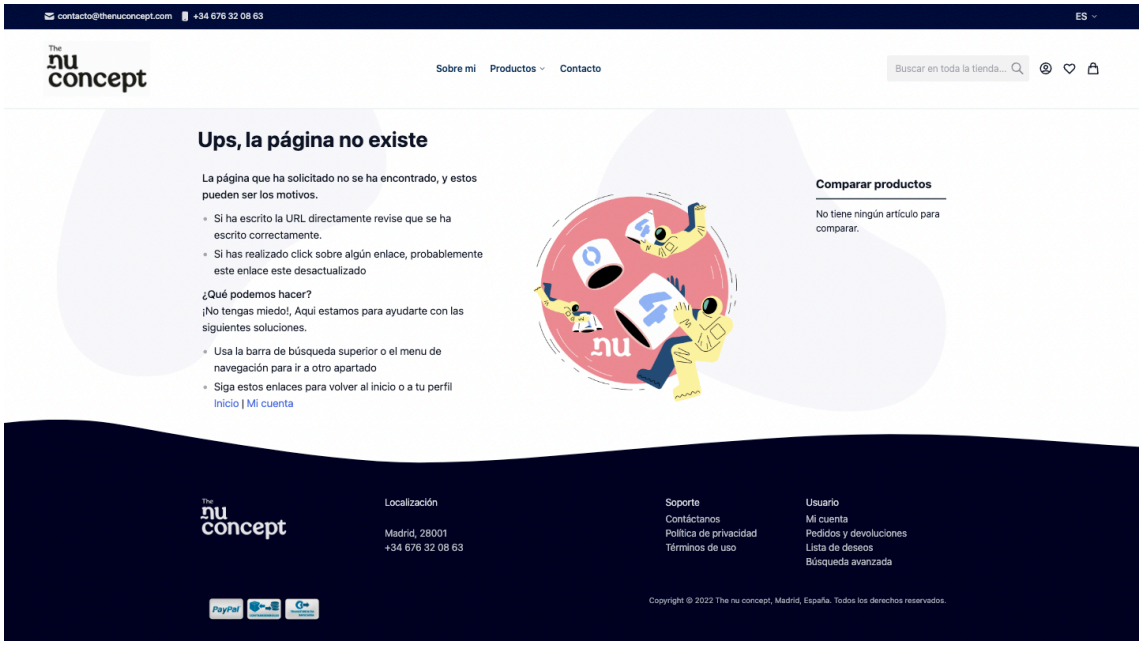


Figura 92. Diseño página de error 404

A parte esta página ha sido configurada también para mostrarla tanto cuando está el idioma en español como cuando está en inglés.

3.6 Pruebas

En este apartado vamos a realizar una serie de pruebas para comprobar que los objetivos fijados se han cumplido y el proyecto ha terminado de forma correcta.

Prueba de compra

Se va a realizar una compra con un usuario ficticio y se realizará el pago a través de un pago de PayPal sobre un sistema sandbox de este.

Para ello previamente debemos estar logueados y elegiremos un producto para realizar la compra, en este caso elegiremos unos pendientes “Calypso” pertenecientes a la categoría de complementos. Para empezar, debemos añadir el producto al carrito.

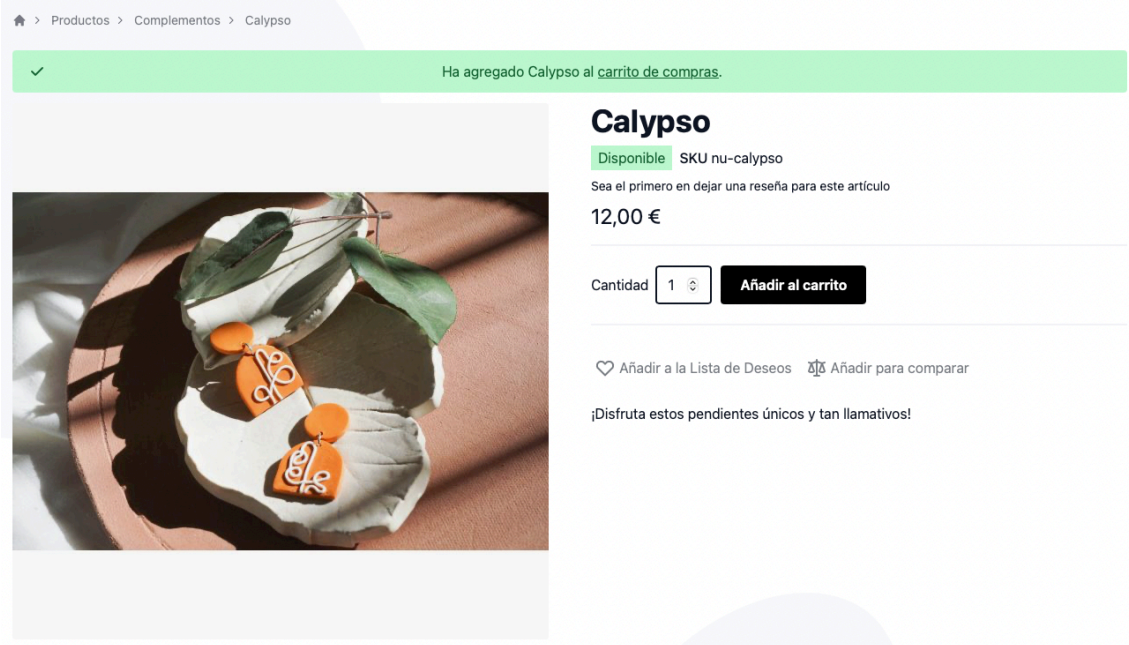


Figura 93. Prueba, añadir producto al carrito

Una vez añadido al carrito debemos abrir este y tramitar el pedido.

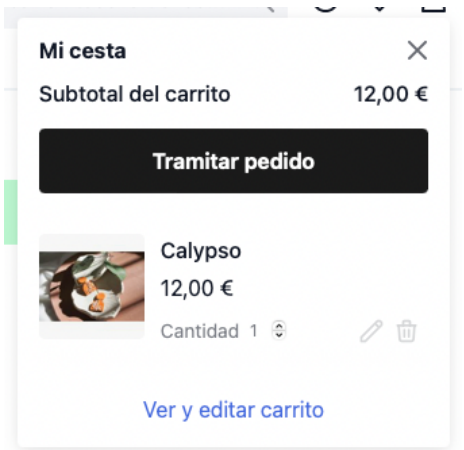


Figura 94. Prueba, tramitar pedido desde el carrito

Si previamente no hemos introducido ninguna dirección en nuestra cuenta nos saldrá una pantalla como esta en la que introduciremos los datos necesarios para continuar.

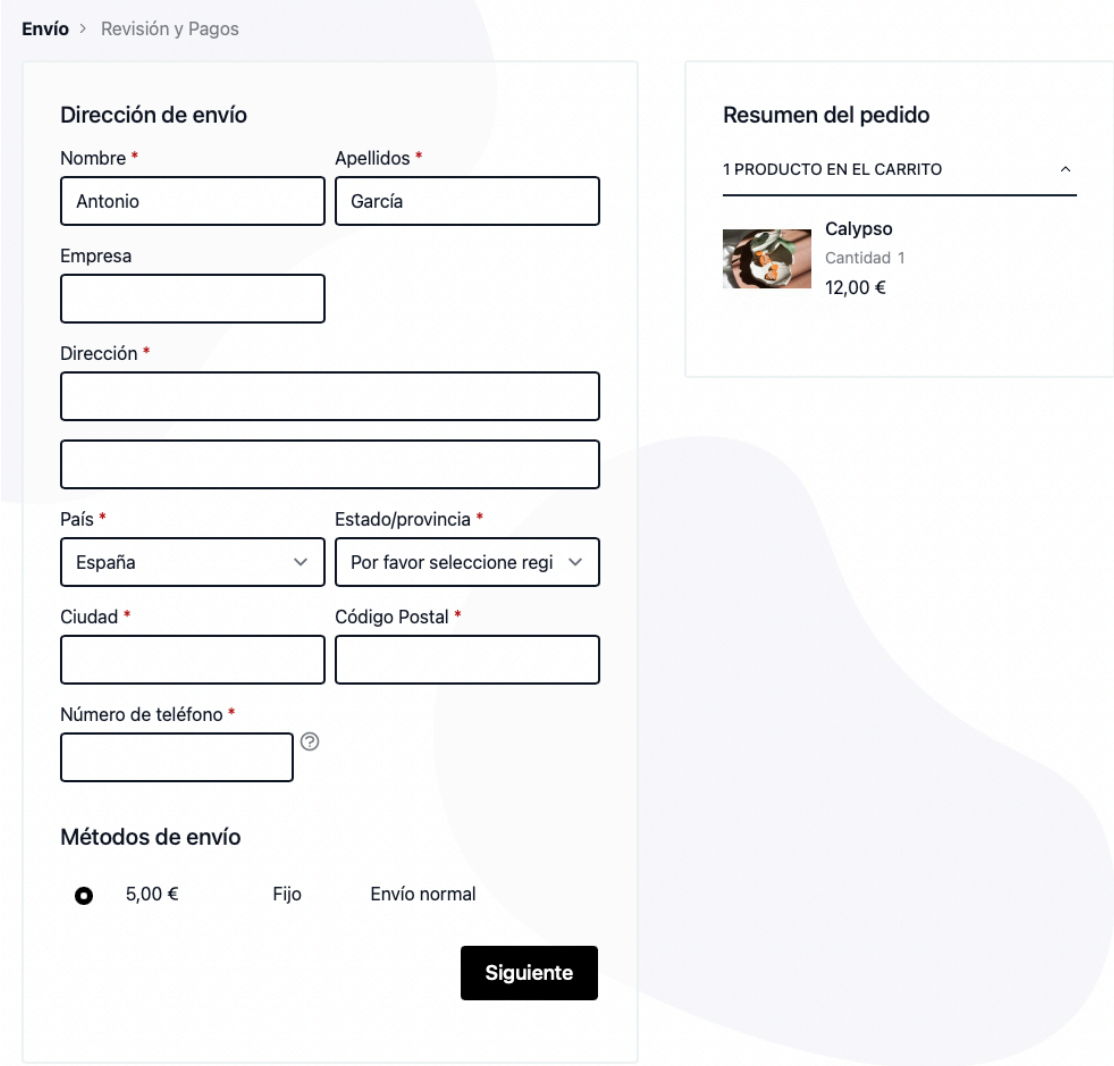


Figura 95. Prueba, formulario datos de dirección de envío

Una vez introducidos los datos nos llevará a una pantalla donde nos mostrará el importe desglosado y con los gastos de envío e impuestos aplicados.

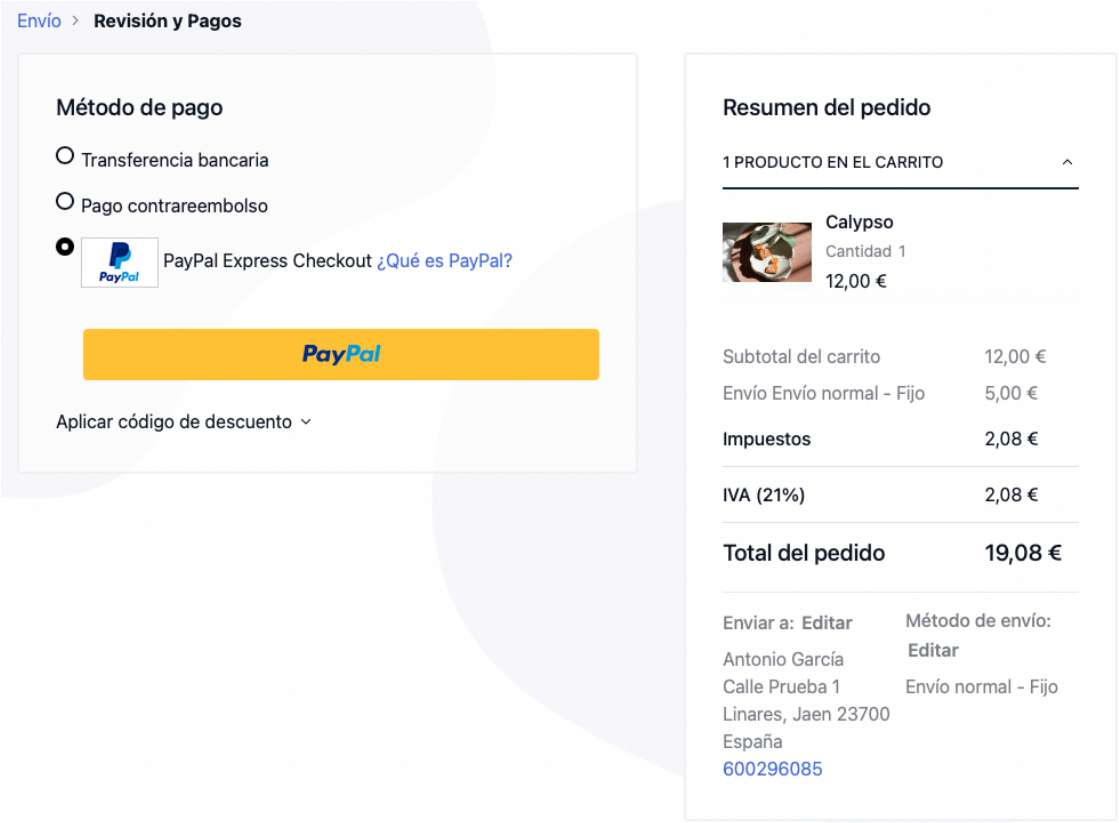


Figura 96. Prueba, resumen del pedido

Ahora haciendo “click” en el botón de PayPal nos llevará a una ventana de sandbox de PayPal donde introduciremos nuestros datos de prueba de usuario para realizar el pago y completar asi la compra.

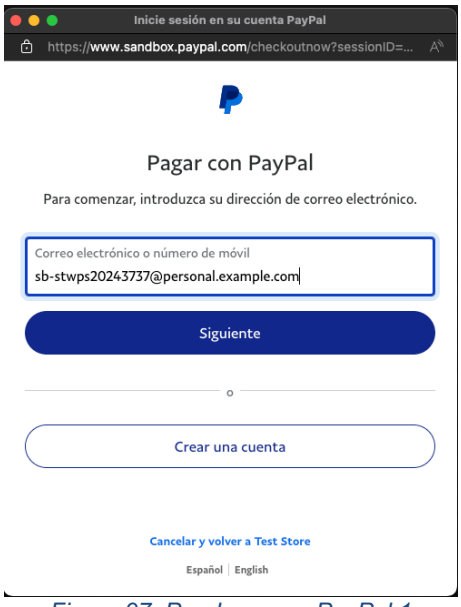


Figura 97. Prueba, pago PayPal 1

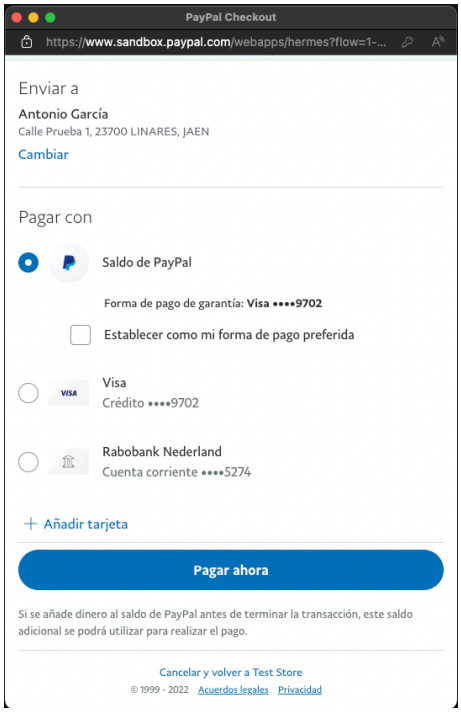


Figura 98. Prueba, pago PayPal 2

Una vez verificado el pago se completará la compra y nos mostrará este mensaje con la posibilidad de imprimir el recibo

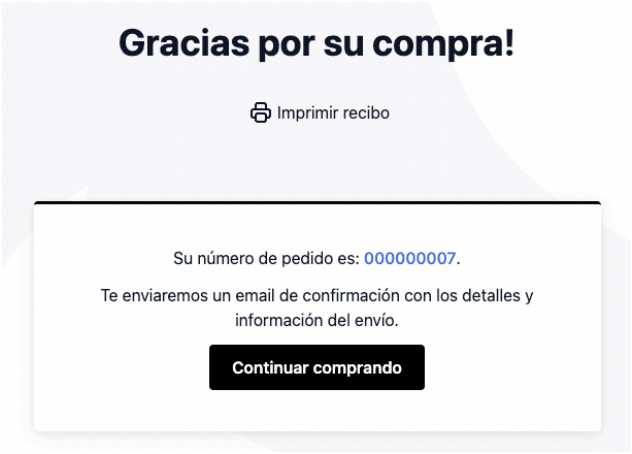


Figura 99. Prueba, comentario confirmación compra

Si accedemos al pedido podemos visualizar los detalles de este en formato factura.

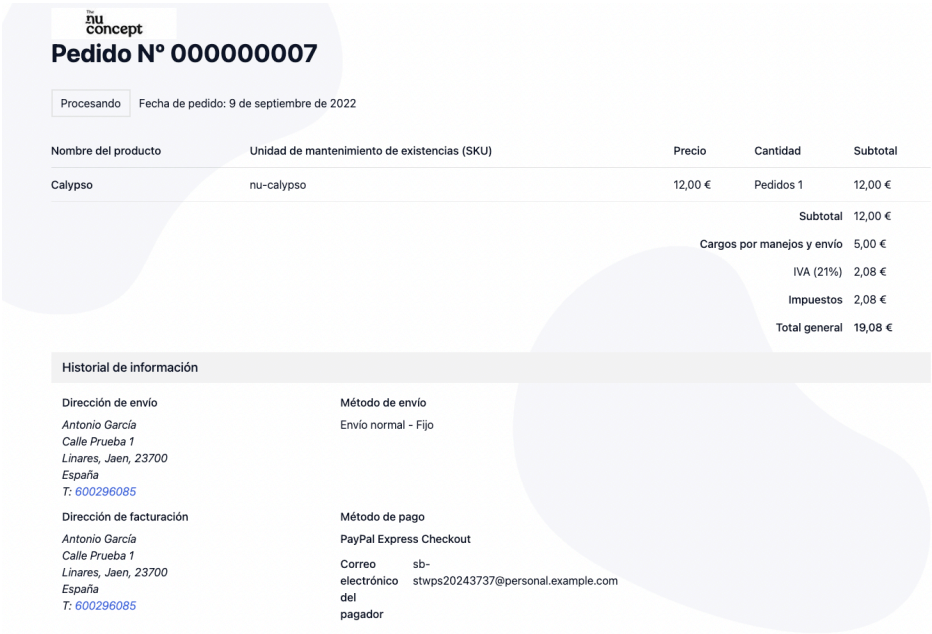


Figura 100. Prueba, impreso compra

Y para confirmar que el pedido se ha realizado correctamente, en nuestra cuenta podemos acceder a “Mis pedidos” y veremos el pedido con estado “Procesando”

Figura 101. Prueba, comprobación pedida.

Prueba responsive design

En esta prueba aplicaremos tres configuraciones de tres dispositivos con distintas dimensiones para comprobar que el sitio web se adapta a las dimensiones del dispositivo

Dimensiones de pantalla de ordenador, 1920x1080

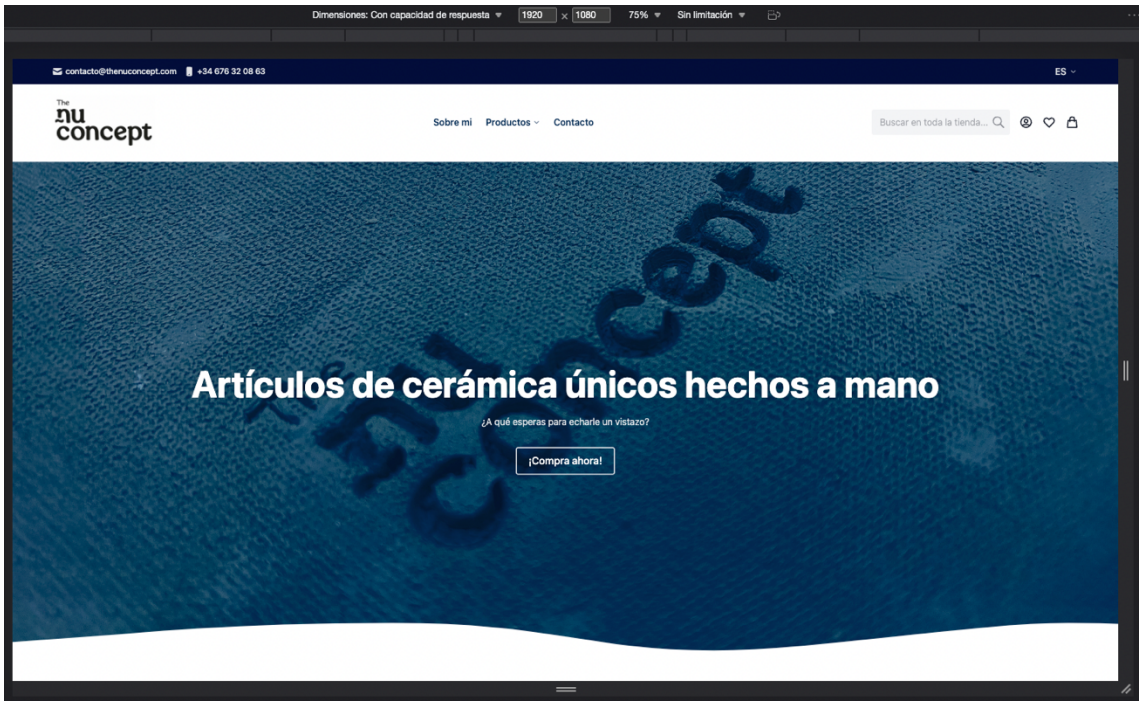


Figura 102. Prueba, dimensiones PC

Dimensiones smartphone iPhone 12 Pro, 390 x 844 (Con menú desplegado y sin el)

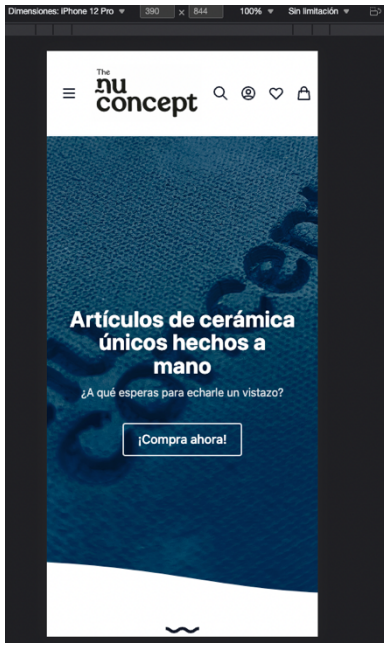


Figura 103. Prueba, dimensión smartphone sin menú

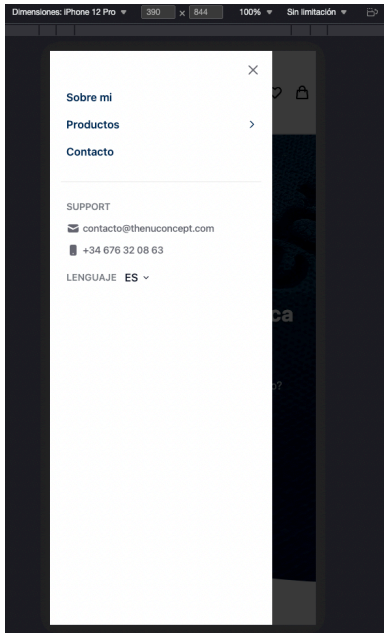
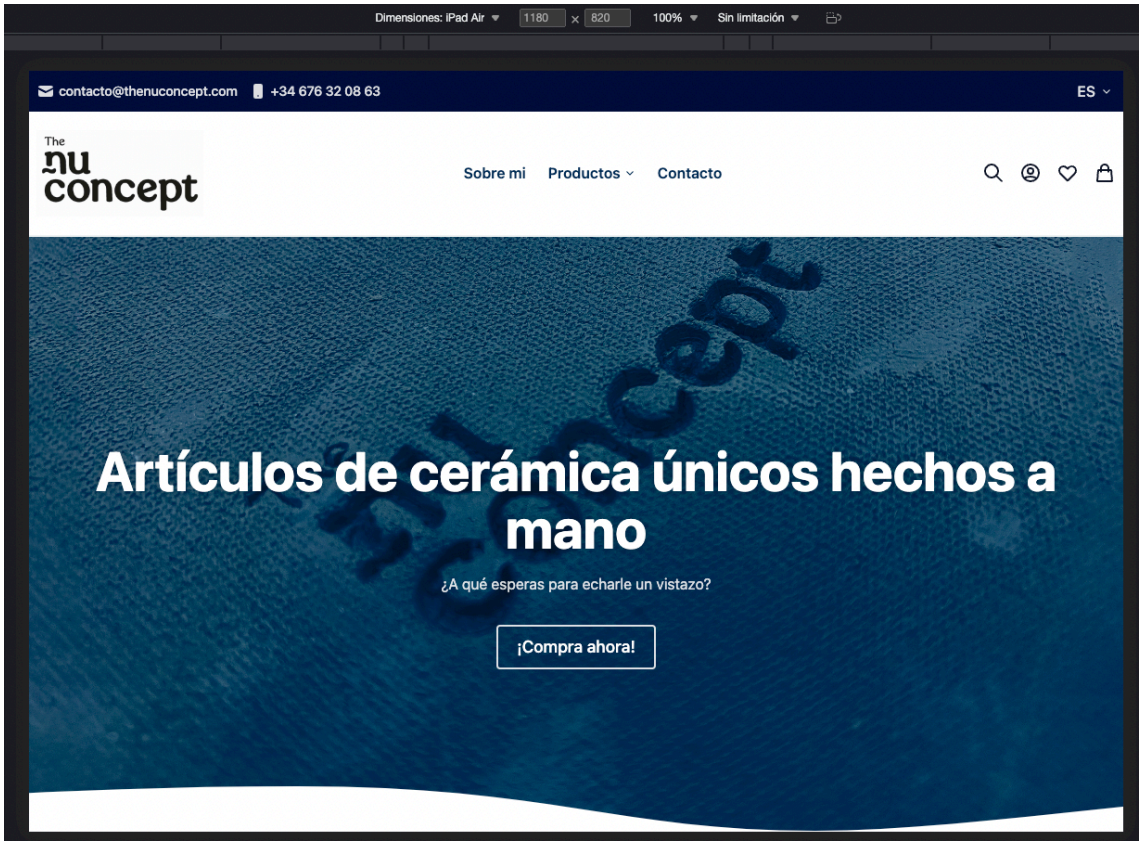


Figura 104. Prueba, dimensión smartphone con menu desplegado

Dimensiones iPad Air, 1180 x 820



Como se observa en las tres pruebas realizadas la página se adapata sin ningún problema a las dimensiones de los diferentes dispositivos, por lo que podemos concluir que posee un diseño adaptativo responsive design.

Prueba de idioma

En esta prueba mostraremos una vista con texto donde cambiando el idioma podemos confirmar que el sitio tiene multilenguaje de español e inglés.

Figura 105. Prueba, página con idioma español

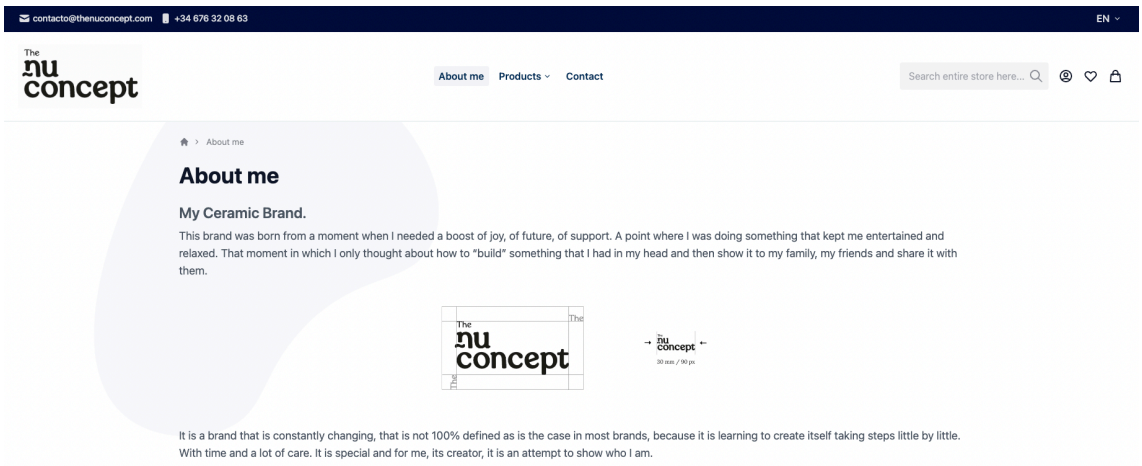


Figura 106. Prueba, página con idioma inglés

CAPITULO 4. CONCLUSIÓN

4.1 Conclusiones

A modo de cierre de este Trabajo de Fin de Grado, realizaremos una conclusión sobre el proyecto de desarrollar una tienda online de artículos de cerámica mediante el uso de un CMS, este proyecto es un proyecto real sobre una marca real y es accesible desde https://thenuconcept.com/

Al margen de esto, este proyecto tenía como objetivo principal desarrollar una tienda online que sea capaz de vender por internet productos de cerámica, con todo lo que ello conlleva.

Elegí esta temática por distintas razones, actualmente la venta online está en alza, no solo desde hace ya 10-15 años, si no que con la pandemia que ha habido estos últimos años la venta online se ha multiplicado de manera notoria, llegando hasta el punto de empresas como “El corte inglés” cerrar centros para aumentar su venta online perjudicando así la venta física y tradicional, es por eso una de las razones por las que quise desarrollar un proyecto de una tienda online desde cero y que mejor manera que mediante de un CMS como Magento que ya está preparado para ello.

Como especifico al principio de este proyecto yo me encuentro trabajando de Desarrollador PHP y solo en mis inicios en el mundo de la informática había conocido los CMS, siempre he programado y trabajado con proyectos que se han hecho desde cero a través de frameworks. Por este motivo pensaba que este proyecto me resultaría más fácil, y siendo honesto ha tenido un grado de dificultad que no esperaba.

Al ser un alumno de ingeniería informática no he querido elegir un servicio hosting que tiene un CMS preinstalado o un cPanel que te lo instala con varios “clicks”, he querido conocer desde cero el desarrollo de un proyecto haciendo una investigación de los CMS actuales, de los servicios de hostings y de las necesidades para desarrollar un proyecto así. Asociar el dominio proporcionado por el cliente al servicio de hosting contratado, configurar un servidor UNIX desde cero e instalar los recursos necesarios para la instalación de Magento, todo esto no era fácil, porque un servidor para un CMS concreto y una versión como es Magento Open Source eCommerce 2.4.0 requiere una versión de PHP concreta configurada de manera óptima, unos recursos software necesarios para la instalación de Magento y una vez instalado mantener el sistema y actualizar cuando sea necesario.

He comprobado de primera mano que el desarrollo de un proyecto así desde cero no es fácil, pero me ha enseñado una parte del desarrollo web que desconocía, ya que en la universidad había visto desarrollo web por medio del frameworks CodeIgniter, y estoy muy contento con el resultado del proyecto y con haber conocido los CMS, en la actualidad son una herramienta muy potente para cualquier empresa, ya sea pequeña, media o grande. Estos sistemas de gestión de contenido debidamente instalados y configurados puede gestionar

tanto tiendas pequeñas como grandes debido a la multitud de configuraciones que tienen, están perfectamente preparados para ello.

En conclusión, puedo confirmar que la tienda online cumple los objetivos previamente establecidos, se ha creado una tienda accesible con capacidad de aceptar nuevos usuarios, vender de forma online mediante tres tipos de pago (Transferencia bancaria, contrarrembolso y PayPal), con una gestión de categorías y productos, gestión de pedidos y envíos y ofreciendo el sitio en dos idiomas (español e inglés). Este trabajo me ha ayudado a completarme un poco más en el mundo del desarrollo web y seguro que me puede servir de cara a un futuro profesional.

4.2 Posibles desarrollos futuros

Un proyecto que acaba de ser creado nunca puede estar acabado, los proyectos tecnológicos deben estar en constante desarrollo actualizándose a la misma vez que va avanzando la tecnología, ya no solo por el avance de la tecnología, si no que no podemos pensar en todas las necesidades que puede tener un usuario, podemos intentar ponernos en su lugar y pensar todas las necesidades que puede tener, pero siempre habrá algo no cubierto.

Es por eso que también este proyecto necesitará desarrollos futuros para mejorar. Sería optimo añadir en un futuro próximo una pasarela de pago para tarjetas, ya que no todo el mundo dispone de PayPal y la transferencia bancaria no es un método de pago rápido.

Otra posible mejora seria introducir como tercer idioma disponible el francés, es un país vecino y nuestra cliente dispone de contactos de Francia por la que podría abrirse una vía de marketing.

Una posible mejora podría ser mejorar las condiciones del hosting, esta mejora va vinculada al número de usuarios y ventas que se produzcan, ya que incrementaría el presupuesto para mantener la tienda, y solo podría producirse si se tuvieran unos beneficios que cubriesen los gastos producidos de más.

ANEXOS

Información de acceso y pruebas

Se proporcionará información para realizar pruebas en la tienda online, para acceder al sistema de administración y datos de sandbox de PayPal para realizar pagos ficticios sobre el usuario de prueba.

Tienda online

- Dominio: https://thenuconcept.com/
- Usuario: usuarioprueba@thenuconcept.com
- Contraseña: PruebaTFG23

Panel de administración
- Dominio: https://thenuconcept.com/admin_16bdbv
- Usuario: soporte
- Contraseña: exlqex34

PayPal Sandbox
- Usuario: sb-stwps20243737@personal.example.com
- Contraseña: C7sf<S0h

Manual de usuario

En este manual de usuario se explicará las distintas funciones que tiene la página y como un usuario puede realizarlas.

Página principal

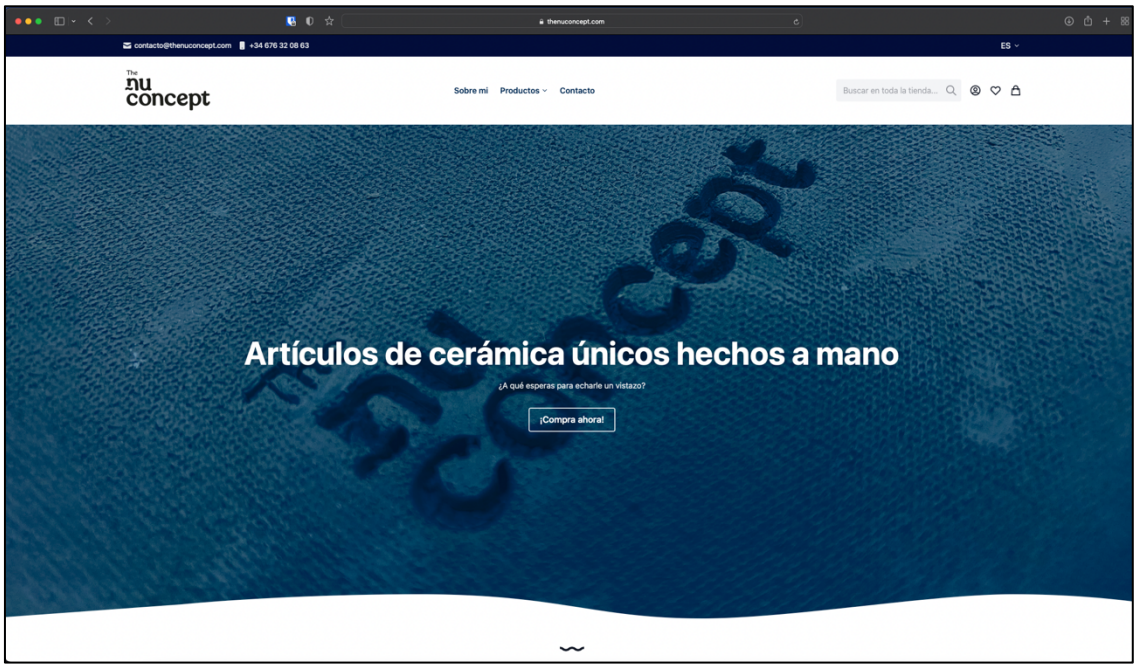


Figura 107. Manual, página principal

En la página inicial podemos encontrar una cabecera con distintas opciones, si desplazamos hacia abajo podemos ver los artículos más novedosos, enlaces a las categorías y enlaces a las redes sociales de la tienda.

Figura 108. Manual, cabecera y categorías

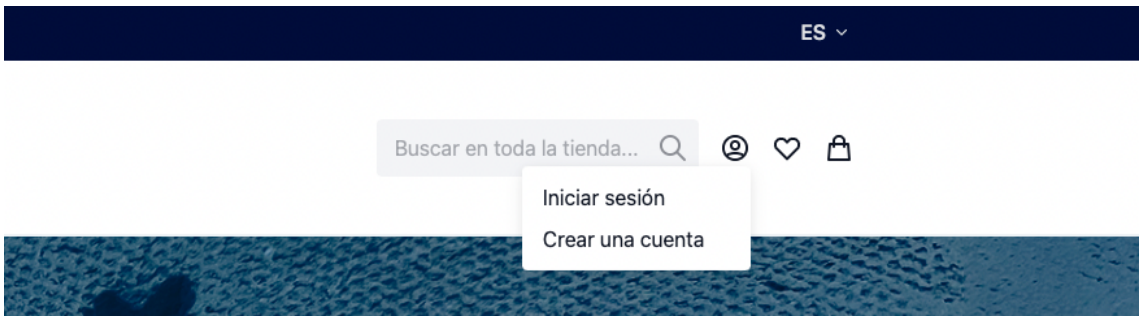


Figura 109. Manual, cabecera cuenta

En la figura 91 nos centramos en la parte derecha de la cabecera donde tenemos acceso al inicio de sesión, crear una cuenta de usuario, al buscador de productos y al carro que se mostrará con productos si tenemos añadidos alguno.

Si hemos realizado login previamente con una cuenta de usuario, también podremos ver la lista de deseos en el corazón, si tenemos algún producto añadido, además que el desplegable de “Mi cuenta” tendría opciones distintas, como acceder a la cuenta del usuario.

A parte también en la esquina superior derecha podemos observar el selector para cambiar de idioma la página.

Registro y Login

Si accedemos a la pantalla de login podemos ver un formulario como este en el que podremos acceder a nuestra cuenta introduciendo los datos requeridos de “Correo electrónico” y “Contraseña”.

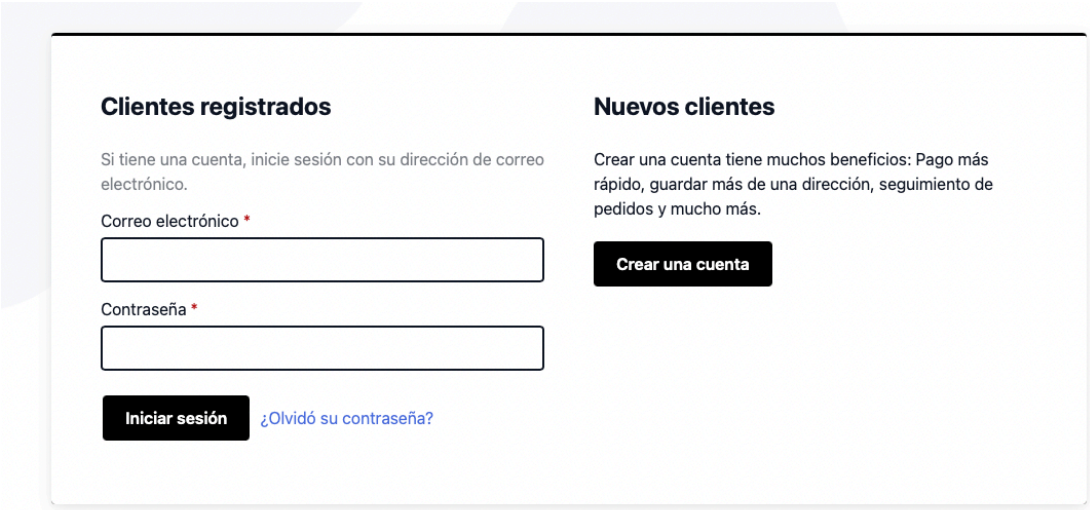


Figura 110. Manual, pantalla login

Si accedemos a la vista de registro veremos un formulario como este, deberemos introducir los datos requeridos y enviarlos para completar el registro de un nuevo usuario.

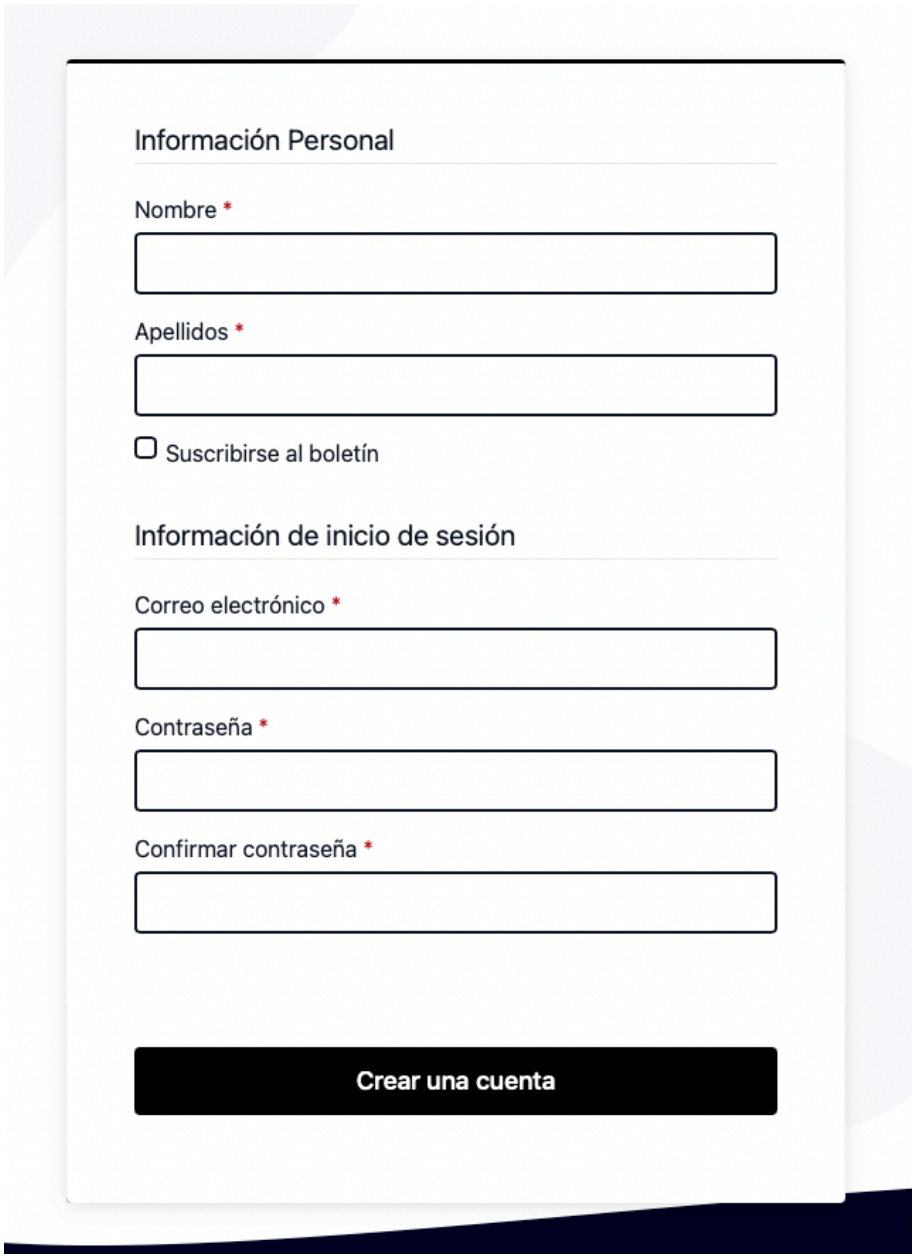


Figura 111. Manual, pantalla de registro

Contacto

La vista de contacto es la siguiente, en la que muestra información de contacto de la marca a través de sus redes sociales, teléfono o email. También dispone de un formulario de contacto si accede desde la opción de “Contáctanos” del footer como podemos ver en la figura 95.

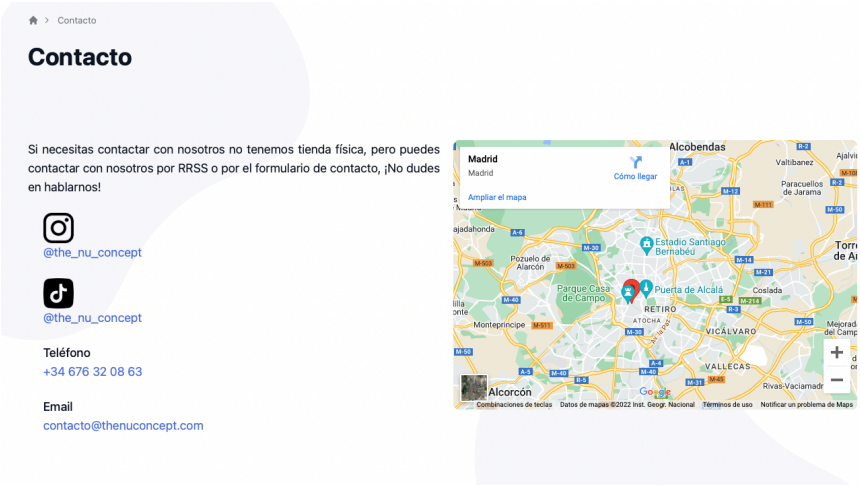


Figura 112. Manual, información de contacto

Figura 113. Manual, formulario de contacto

Búsqueda avanzada

Desde el footer tiene la opción de acceder al formulario de búsqueda avanzada donde podrá buscar un producto filtrándolo por distintos atributos para realizar una búsqueda más exhaustiva.

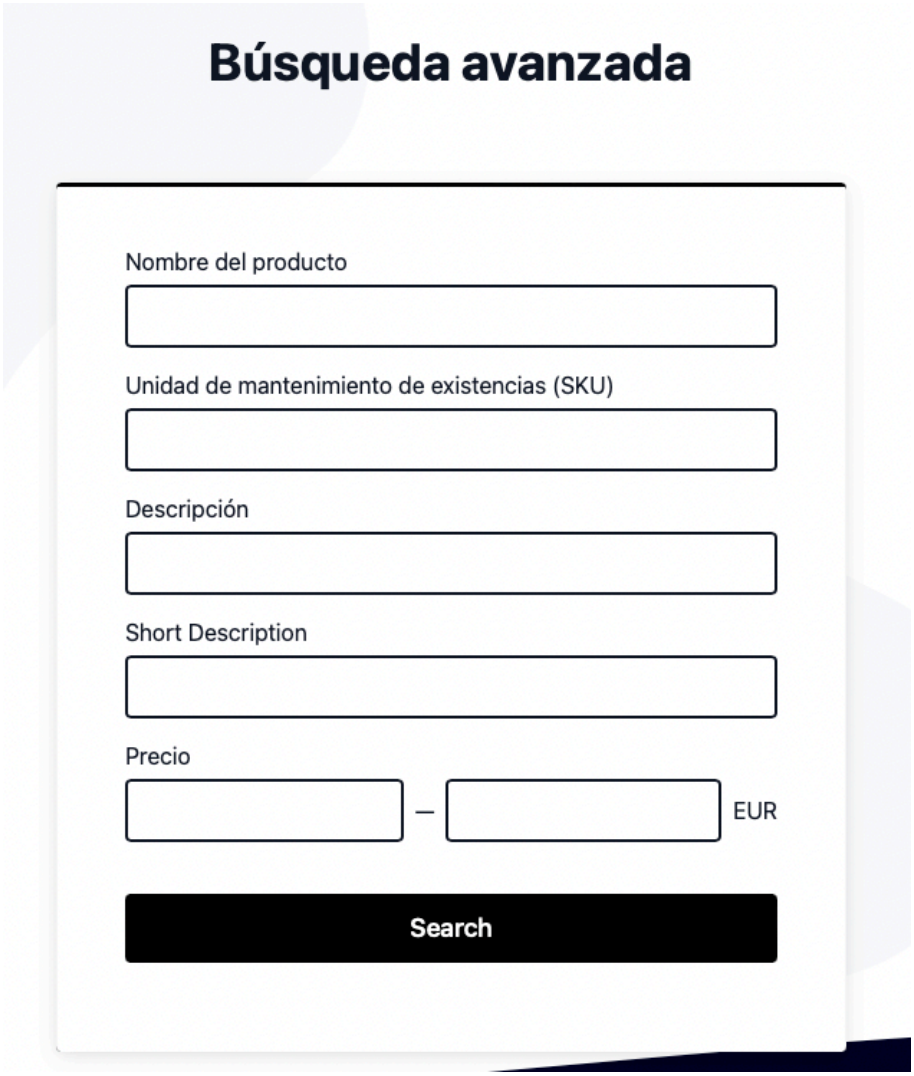


Figura 114. Manual, búsqueda avanzada

Pedidos y devoluciones

También desde el footer se puede acceder al formulario de pedidos y devoluciones, si tiene algún problema con un pedido o requiere realizar una devolución puede hacerlo desde este formulario introduciendo los campos requeridos.

Figura 115. Manual, formulario de pedidos y devoluciones

Sobre mi

En este apartado mostramos información sobre la marca, como nació, de donde sale y unas ilustraciones del lugar de trabajo donde se realizan los productos a mano.

Figura 116. Manual, pantalla "sobre mi"

Figura 117. Manual, ilustraciones del lugar de trabajo

Productos y categorías

Esta vista es similar para todos los productos como para categorías concretas en las que veremos los productos y con la barra lateral derecha podremos filtrar entre categorías o precio.

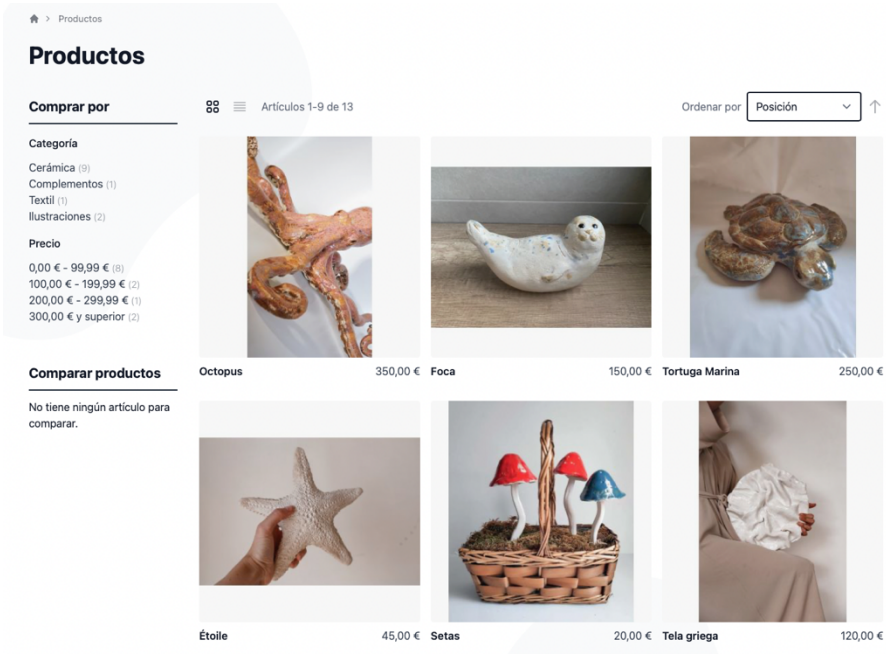


Figura 118. Manual, vista productos y categorías

Producto al detalle

En esta vista podemos ver un producto de forma detallada, su sku, imágenes de él, una descripción corta y otra más extensa, a parte de poder escribir una reseña del producto para poder opinar del producto y que otros usuarios puedan guiarse por su reseña.

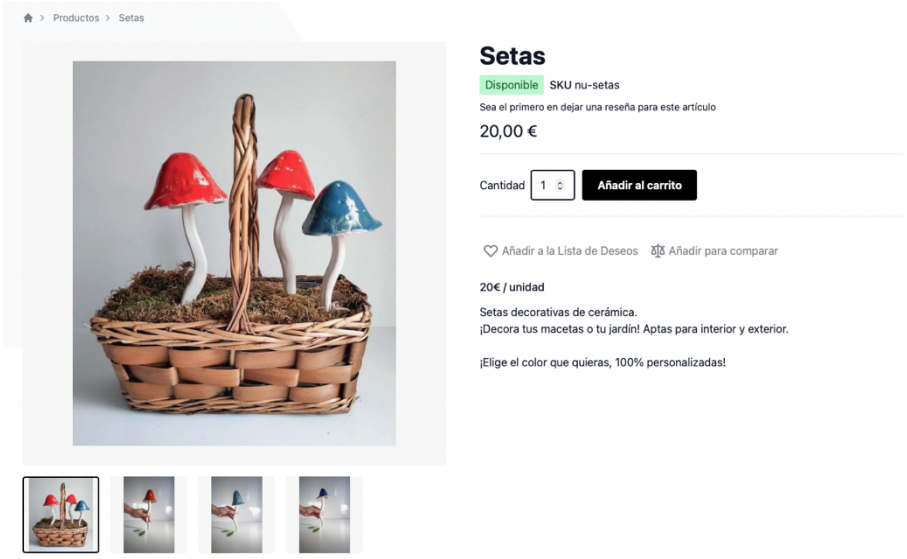


Figura 119. Manual, producto al detalle

Mi cuenta

En esta pantalla puedes acceder a los detalles de tu cuenta y modificarlos si lo necesitas, además de ver un listado de los pedidos y sus detalles, la lista de deseos del usuario y los datos relacionados con los pagos y envíos.

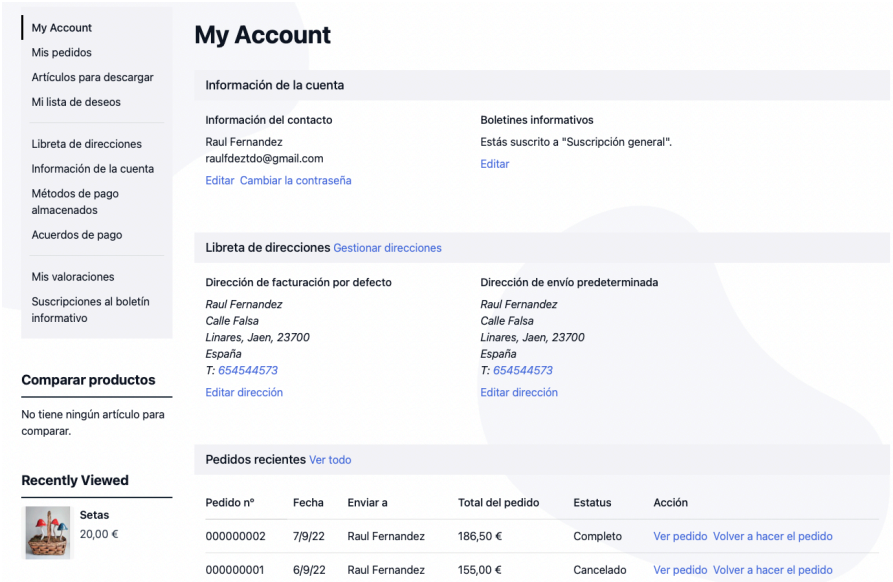


Figura 120. Manual, pantalla de "Mi cuenta"

Software utilizado

Terminal MAC, se ha utilizado el terminal que viene preinstalado en el MacBook Air M1, el cual nos conectábamos al servidor a través del protocolo SSH y se ha utilizado para instalar y configurar los recursos software necesarios para la realización del proyecto.

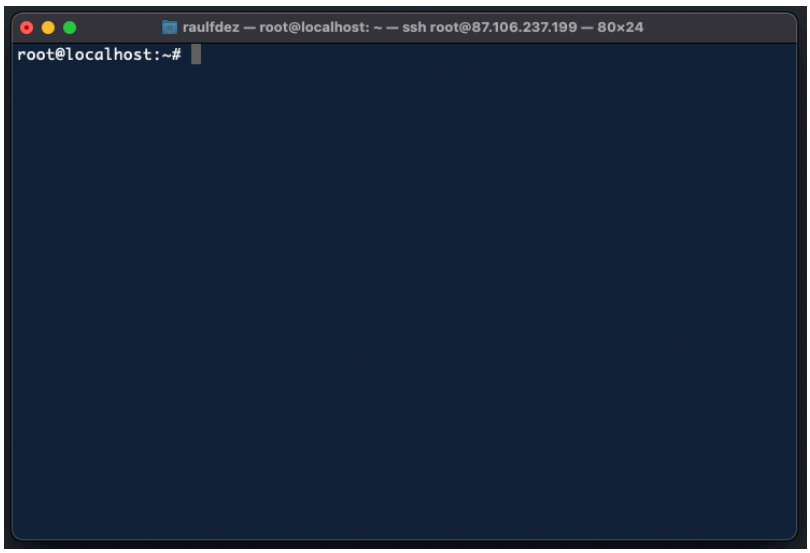


Figura 121. Terminal MacBook conectado al servidor

FileZilla

Este programa se ha utilizado para subir ficheros como los ficheros del certificado SSL o ficheros requeridos para el proceso de instalación o configuración, utiliza una conexión SFTP para conectarse al servidor.

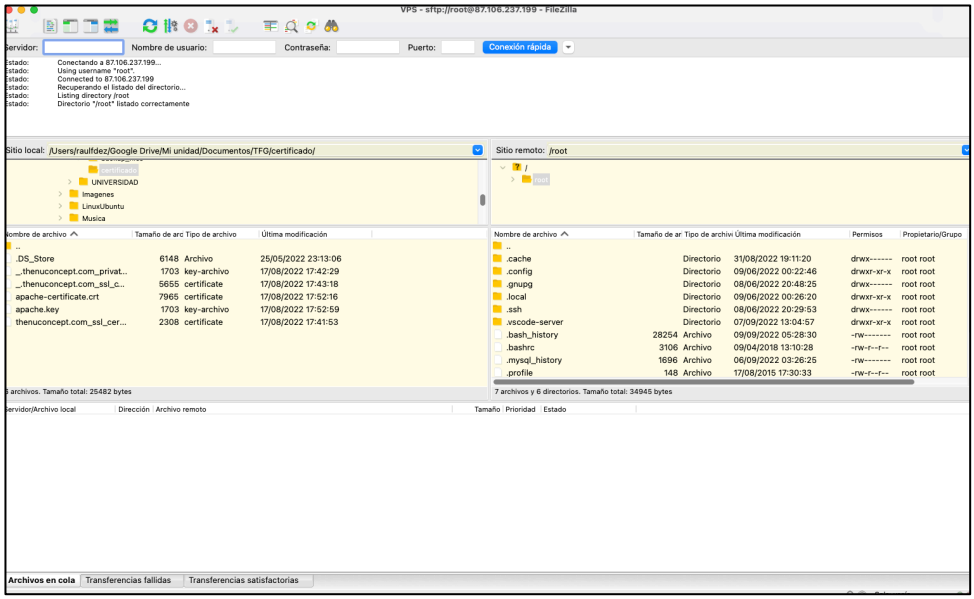


Figura 122. FileZilla software

Visual Studio Code

Este software lo hemos utilizado para el desarrollo de bloques y paginas antes de introducir el código a través del panel de administración de Magento. Además el VSCode permite conectarse, por medio del protocolo SSL, al servidor y así poder ver el sistema de ficheros de Magento y del servidor y realizar modificaciones de forma más fácil

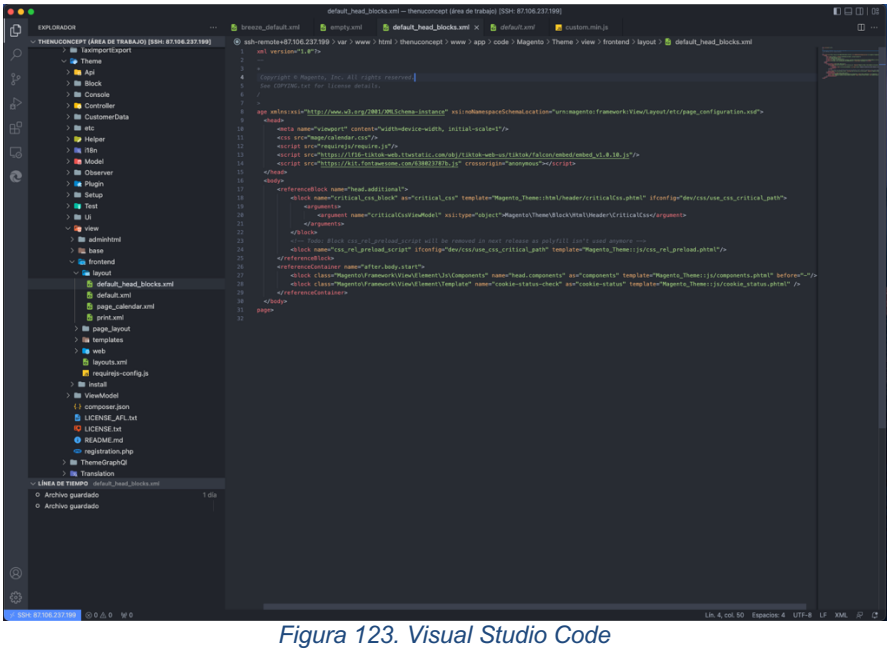
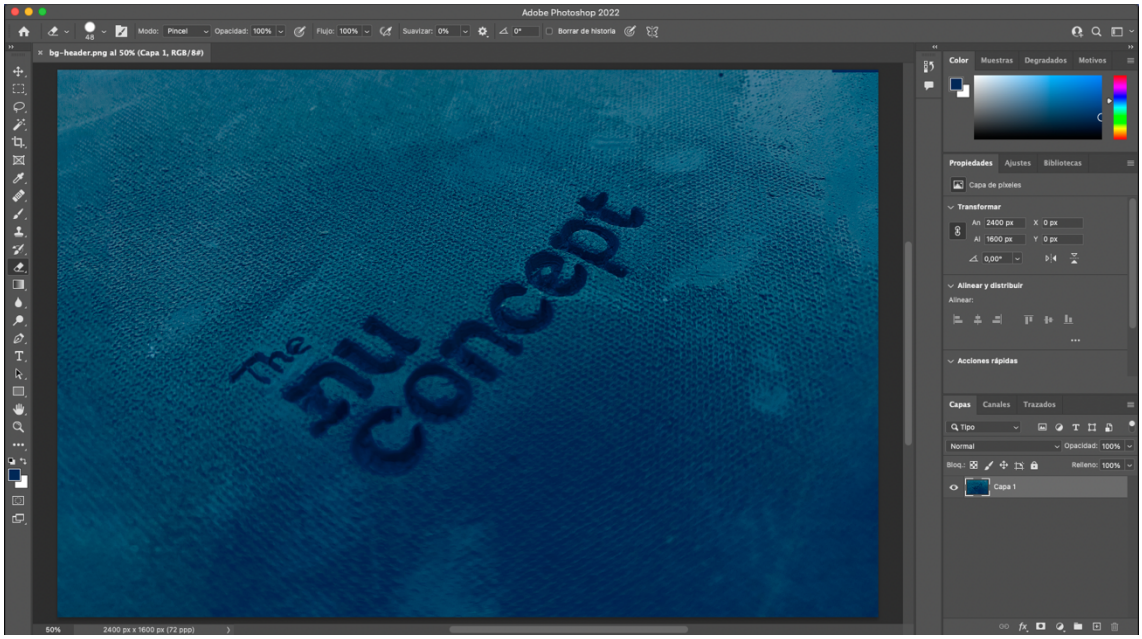


Figura 123. Visual Studio Code

Adobe Photoshop 2022

Hemos utilizado el Adobe Photoshop 2022 para realizar diseños personalizados de imágenes que harán la pagina mas atractiva y con un diseño a medida para la marca “The nu concept”.



BIBLIOGRAFÍA

1. Bello, E. (2021) "Qué es un CMS, características y cuáles son los más populares", ***Thinking for Innovation*, p. (Accedido: 19 Julio 2022)**
   Disponible en: https://www.iebschool.com/blog/que-es-un-cms-e-commerce/

2. Sistema de gestión de contenidos - Wikipedia, la enciclopedia libre
   ***Sistema de gestión de contenidos - Wikipedia, la enciclopedia libre* (2022). (Accedido: 19 Julio 2022)**
   Disponible en: https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos

3. 10 Características de un CMS
   ***10 Características de un CMS* (2019). (Accedido: 19 Julio 2022)**
   Disponible en: https://www.arquitectoit.com/cms/10-caracteristicas-cms/

4. CMS
   ***CMS* (2022). (Accedido: 19 Julio 2022)**
   Disponible en: https://desarrolloweb.com/home/cms

5. Fraguela, N.
   Fraguela, N. (2022) *Top CMS para eCommerce: las mejores plataformas para tu tienda (2022)*, *Marketing 4 Ecommerce - Tu revista de marketing online para e-commerce*. (Accedido: 19 Julio 2022)
   Disponible en: https://marketing4ecommerce.net/mejores-cms-para-ecommerce/

6. Ecommerce market share, websites and contacts - Wappalyzer
   ***Ecommerce market share, websites and contacts - Wappalyzer* (2022 2022). (Accedido: 20 Julio)**
   Disponible en: https://www.wappalyzer.com/technologies/ecommerce/

7. [CMS para e-commerce] Qué es y cuál es el mejor [2022]
   ***[CMS para e-commerce] Qué es y cuál es el mejor [2022]* (2022). (Accedido: 20 Julio 2022)**
   Disponible en: https://www.doofinder.com/es/blog/cms-tienda-online

8. ¿Qué es Squarespace? Todo lo que necesitas saber
   ***¿Qué es Squarespace? Todo lo que necesitas saber* (2020). (Accedido: 20 Julio 2022)**
   Disponible en: https://ecommerce-platforms.com/es/articles/what-is-squarespace

9. System requirements | Adobe Commerce Developer Guide
***System requirements | Adobe Commerce Developer Guide* (2022). (Accedido: 9 Agosto 2022)**
Disponible en: https://devdocs.magento.com/guides/v2.4/install-gde/system-requirements.html

10. A brief history of the Content Management System
***A brief history of the Content Management System* (2022). (Accedido: 9 Agosto 2022)**
Disponible en: https://opensource.com/article/20/7/history-content-management-system

11. López, B.
López, B. (2022) *Mejor VPS barato en España en 2022 Comparativa VPS*, *Ciudadano 2.0*. (Accedido: 9 Agosto 2022)
Disponible en: https://www.ciudadano2cero.com/mejor-hosting-vps/

12. Requerimientos Funcionales y No Funcionales, ejemplos y tips
***Requerimientos Funcionales y No Funcionales, ejemplos y tips* (2018). (Accedido: 6 Septiembre 2022)**
Disponible en: https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a

13. Requisitos funcionales y requisitos no funcionales - Otro
***Requisitos funcionales y requisitos no funcionales - Otro* (2022). (Accedido: 6 Septiembre 2022)**
Disponible en: https://spa.myservername.com/features-functional-requirements

14. ¿Qué es un certificado SSL?
***¿Qué es un certificado SSL?* (2020). (Accedido: 2 Septiembre 2022)**
Disponible en: https://www.globalsign.com/es/ssl-information-center/what-is-an-ssl-certificate

15. Qué es un certificado SSL | Redalia
***Qué es un certificado SSL | Redalia* (2022). (Accedido: 2 Septiembre 2022)**
Disponible en: https://www.redalia.es/ssl/certificado-ssl/

16. Step by Step how to install Magento 2.4.x on Ubuntu (with Elasticsearch)
***Step by Step how to install Magento 2.4.x on Ubuntu (with Elasticsearch)* (2022). (Accedido: 17 Agosto 2022)**
Disponible en: https://magentip.com/install-magento-2-4-x-on-ubuntu-with-elasticsearch/

17. Conectar un dominio con una dirección IP estática - IONOS Ayuda
***Conectar un dominio con una dirección IP estática - IONOS Ayuda* (2022). (Accedido: 16 Agosto 2022)**

Disponible en: https://www.ionos.es/ayuda/dominios/configurar-la-direccion-ip/conectar-el-dominio-con-una-direccion-ip-estatica/

18. Certificate Installation: Apache & mod_ssl - Powered by Kayako Help Desk Software
***Certificate Installation: Apache & mod_ssl - Powered by Kayako Help Desk Software* (2022). (Accedido: 17 Agosto 2022)**
Disponible en: https://support.comodo.com/index.php?/Knowledgebase/Article/View/637/37/certificate-installation-apache--mod_ssl

19. Layout overview | Adobe Commerce Developer Guide
***Layout overview | Adobe Commerce Developer Guide* (2020). (Accedido: 3 Septiembre 2022)**
Disponible en: https://devdocs.magento.com/guides/v2.3/frontend-dev-guide/layouts/layout-overview.htm

20. GitHub - breezefront/breeze-evolution: Metapackage for Breeze Evolution theme
***GitHub - breezefront/breeze-evolution: Metapackage for Breeze Evolution theme* (2022). (Accedido: 23 Agosto 2022)**
Disponible en: https://github.com/breezefront/breeze-evolution

21. Breeze | Installation
***Breeze | Installation* (2022). (Accedido: 23 Agosto 2022)**
Disponible en: https://breezefront.com/installation

22. Salario para Desarrollador Web en España - Salario Medio
***Salario para Desarrollador Web en España - Salario Medio* (2022). (Accedido: 5 Septiembre 2022)**
Disponible en: https://es.talent.com/salary?job=desarrollador+web

23. Diagramas de secuencia: mostrar interacciones con UML
***Diagramas de secuencia: mostrar interacciones con UML* (2019). (Accedido: 2 Septiembre 2022)**
Disponible en: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagramas-de-secuencia/

24. Metodologías de desarrollo de software: ¿qué son?
***Metodologías de desarrollo de software: ¿qué son?* (2022). (Accedido 8 Septiembre 2022)**
Disponible en: https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html

25. Grudemi, E.
Grudemi, E. (2018) *Metodologías ágiles - ¿Qué son?, ventajas, desventajas y las más usadas*, *Enciclopedia Económica*. (Accedido 8 Septiembre 2022)
Disponible en: https://enciclopediaeconomica.com/metodologias-agiles/

26. El modelo en cascada: desarrollo secuencial de software
***El modelo en cascada: desarrollo secuencial de software* (2019). (Accedido 8 Septiembre 2022)**
Available at: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/