Escuela Politécnica Superior de Jaén

Grado en Ingeniería Informática

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Grado

DESARROLLO DE UNA TIENDA ONLINE ORIENTADA A LA VENTA DE VIDEOJUEGOS MEDIANTE EL USO DE UN CMS

Alumno: Pablo Lozano Ropa

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

Junio, 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 orientada a la venta de videojuegos mediante el uso de un CMS, que presenta Pablo Lozano Ropa, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Junio de 2022

El alumno:

El tutor:

GOMEZ ESPINOLA JOSE IGNACIO - 24277832K

Firmado digitalmente por GOMEZ ESPINOLA JOSE IGNACIO - 24277832K
Fecha: 2022.06.27 17:14:11 +02'00'

Pablo Lozano Ropa

José Ignacio Gómez Espínola

Índice

Índice de figuras *5*
Agradecimientos *10*
CAPÍTULO 1. INTRODUCCIÓN *11*
1.1. Introducción al proyecto. **11**
1.2. Motivación. **11**
1.3. Objetivos. **12**
CAPÍTULO 2. ESTADO DEL ARTE *14*
2.1. CMS. **14**
2.2. Características de un CMS. **14**
2.3. Ventajas e inconvenientes de un CMS. **14**
2.3.1. Ventajas de un CMS. **14**
2.3.2. Inconvenientes de un CMS. **15**
2.4. Tipos de CMS. **15**
2.5. Elección del CMS **21**
2.6. Servicio de alojamiento web. **22**
2.6.1. Definición. **22**
2.6.3. Elección de servicio de alojamiento web. **23**
CAPÍTULO 3. TECNOLOGÍAS UTILIZADAS. *24*
3.1. Visual Paradigm. **24**
3.2. Magento 2 ..... **24**
3.3. XAMPP **24**
3.4. GanttProject. **25**
3.5. Selenium. **25**
CAPÍTULO 4. INGENIERÍA DEL SOFTWARE. *26*
4.1. Introducción. **26**
4.1.1. Modelos de Ingeniería del Software. **26**
4.1.2. Elección del modelo de Ingeniería del Software. **28**
4.2. Definición de requisitos. **28**
4.2.1. Requisitos funcionales. **29**
4.2.2. Requisitos no funcionales. **30**
4.3. Planificación. **31**
4.3.1. Estimación de tiempos. **31**
4.3.2. Estimación de costes. **32**
4.4. Análisis **32**

4.4.1. Diagrama de clases. ... 33
4.4.2. Diagramas de casos de uso. ... 33
4.5. Diseño. ... 35
4.5.1. Diagramas de secuencia. ... 35
4.5.2. Diseño de la interfaz. ... 40
4.6. Implementación. ... 55
4.6.1. Instalación del servidor web y de Magento 2. ... 55
4.6.2. Desarrollo web de MegaGameShop. ... 63
4.6.2.1. Instalación de la plantilla. ... 63
4.6.2.2. Accesibilidad en varios idiomas. ... 65
4.6.2.3. Personalización de la cabecera y pie de página. ... 66
4.6.2.4. Administración de categorías y subcategorías. ... 67
4.6.2.5. Configuración de impuestos. ... 70
4.6.2.5. Administración de productos. ... 71
4.6.2.6. Administración de bloques. ... 75
4.6.2.7. Modificaciones en el código. ... 77
4.6.2.8. Administración de cupones de descuento. ... 87
4.6.2.9. Administración de métodos de envío. ... 87
4.6.2.10. Administración de métodos de pago. ... 88
4.7. Pruebas. ... 90
4.7.1. Prueba de añadir un producto al carrito. ... 90
4.7.2. Prueba de añadir un producto a la lista de deseos. ... 92
4.7.3. Prueba de aplicar cupón de descuento a un producto con éxito. ... 94
4.7.4. Prueba de aplicar cupón de descuento a un producto sin éxito. ... 95
4.7.5. Prueba de inicio de sesión con éxito. ... 97
4.7.6. Prueba de inicio de sesión con formato de email erróneo. ... 98
4.7.7. Prueba de inicio de sesión con email o contraseña errónea. ... 100
4.7.8. Prueba de creación de cuenta de cliente. ... 101
4.7.9. Prueba de compra de cliente registrado. ... 103
4.7.10. Prueba de compra de cliente invitado. ... 106
CAPÍTULO 5. CONCLUSIÓN. ... 109
5.1. Conclusiones. ... 109
5.2. Posibles desarrollos futuros. ... 109
BIBLIOGRAFÍA ... 110

Índice de figuras

Figura 1. Evolución trimestral del volumen de negocio del comercio electrónico y variación interanual. ... 12
Figura 2. CMS más utilizados. ... 15
Figura 3. Logotipo de WooCommerce. ... 16
Figura 4. Logotipo de Shopify. ... 17
Figura 5. Logotipo de Squarespace Commerce. ... 18
Figura 6. Logotipo de Wix eCommerce ... 19
Figura 7. Logotipo de PrestaShop. ... 19
Figura 8. Logotipo de Magento. ... 20
Figura 9. Logotipo de Visual Paradigm. ... 24
Figura 10. Logotipo de Magento 2. ... 24
Figura 11. Logotipo de XAMPP. ... 25
Figura 12. Logotipo de GanttProject. ... 25
Figura 13. Logotipo de Selenium. ... 25
Figura 14. Modelo en cascada. ... 26
Figura 15. Modelo incremental. ... 27
Figura 16. Modelo en espiral. ... 27
Figura 17. Etapas de las metodologías ágiles. ... 28
Figura 18. Estimación de tiempos. ... 31
Figura 19. Diagrama de Gantt. ... 32
Figura 20. Estimación de costes. ... 32
Figura 21. Diagrama de clases. ... 33
Figura 22. Diagrama general de casos de uso del usuario. ... 34
Figura 23. Diagrama general de casos de uso del administrador. ... 35
Figura 24. Diagrama de secuencia de registro del cliente. ... 36
Figura 25. Diagrama de secuencia de búsqueda de productos. ... 37
Figura 26. Diagrama de secuencia de inicio de sesión. ... 37
Figura 27. Diagrama de secuencia de añadir producto al carrito. ... 38
Figura 28. Diagrama de secuencia de realizar un pedido. ... 39
Figura 29. Prototipo de la vista principal de MegaGameShop. ... 40
Figura 30. Prototipo de la vista principal para SmartPhone de MegaGameShop. ... 41
Figura 31. Prototipo de la vista de una categoría de MegaGameShop. ... 42

Figura 32. Prototipo de la vista de categorías para SmartPhone de MegaGameShop. 43

Figura 33. Prototipo de la vista de un producto de MegaGameShop. 44

Figura 34. Prototipo de la vista un producto para SmartPhone de MegaGameShop. 45

Figura 35. Prototipo de la vista del carrito de compra de MegaGameShop. 46

Figura 36. Prototipo de la vista del carrito del carrito de compra para SmartPhone de MegaGameShop. 47

Figura 37. Prototipo de la vista de pedido de MegaGameShop. 48

Figura 38. Prototipo de la vista del pedido para SmartPhone de MegaGameShop. 49

Figura 39. Prototipo de la vista de iniciar sesión o crear cuenta de MegaGameShop. 50

Figura 40. Prototipo de la vista de iniciar sesión o crear cuenta para SmartPhone de MegaGameShop. 51

Figura 41. Prototipo de la vista de crear cuenta de MegaGameShop. 52

Figura 42. Prototipo de la vista de crear cuenta para SmartPhone de MegaGameShop. 53

Figura 43. Prototipo de la vista del panel de control de cliente de MegaGameShop. 54

Figura 44. Prototipo de la vista del panel de control de cliente para SmartPhone de MegaGameShop. 55

Figura 45. Descarga de XAMPP. 56

Figura 46. Componentes de XAMPP. 56

Figura 47. Módulos a iniciar en XAMPP. 57

Figura 48. Configuración del puerto principal en XAMPP. 57

Figura 49. Descarga de Magento. 58

Figura 50. Comienzo de la instalación de Magento 2. 58

Figura 51. Comprobación de las extensiones PHP. 59

Figura 52. Acceder al fichero php.ini en XAMPP. 59

Figura 53. Creación de base de datos en phpMyAdmin. 60

Figura 54. Configuración de la base de datos en Magento. 60

Figura 55. Configuración web de Magento. 61

Figura 56. Configuración moneda, región e idioma de MegaGameShop. 61

Figura 57. Configuración de datos del panel de administración de Magento. 62

Figura 58. Finalización de instalación de Magento. 62

Figura 59. Descarga del tema Next - Free Magento 2 Theme. 63

Figura 60. Consola de XAMPP. 64

Figura 61. Selección del tema Next en Magento. 64

Figura 62. Visualización del tema Next en el frontend. 65

Figura 63. Creación de vistas en distintos idiomas. 65

Figura 64. Desplegable con los idiomas de MegaGameShop. 66
Figura 65. Configuración del icono y título de la pestaña de MegaGameShop. 66
Figura 66. Icono y título de la pestaña de MegaGameShop. 66
Figura 67. Configuración de la cabecera de MegaGameShop. 67
Figura 68. Cabecera de MegaGameShop. 67
Figura 69. Datos de contacto del footer de MegaGameShop. 67
Figura 70. Configuración de categorías y subcategorías de MegaGameShop. 68
Figura 71. Categorías y subcategorías en el frontend versión web de MegaGameShop. 69
Figura 72. Categorías y subcategorías en el frontend versión Smartphone de MegaGameShop. 69
Figura 73. Configuración de Tasas en el backend. 70
Figura 74. Configuración de regla de impuestos en el backend. 70
Figura 75. Creación de un producto en el backend (parte I). 71
Figura 76. Creación de un producto en el backend (parte II). 72
Figura 77. Creación de un producto en el backend (parte III). 72
Figura 78. Producto en el frontend versión web de MegaGameShop. 74
Figura 79. Producto en el frontend versión Smartphone de MegaGameShop. 75
Figura 80. Administración de bloques en el backend. 76
Figura 81. Administración del bloque Home Slider en el backend. 76
Figura 82. Código fuente del bloque Home Slider (parte I). 77
Figura 83. Código fuente del bloque Home Slider (parte II). 78
Figura 84. Código fuente del bloque Home Slider (parte III). 78
Figura 85. Carrusel de imágenes en el frontend para versión web de MegaGameShop. 79
Figura 86. Carrusel de imágenes en el frontend para versión Smartphone de MegaGameShop. 79
Figura 87. Código fuente del bloque Home Images Block. 80
Figura 88. Banners publicitarios en el frontend para versión web de MegaGameShop. 80
Figura 89. Banners publicitarios en el frontend para versión Smartphone de MegaGameShop. 81
Figura 90. Código fuente del bloque Home Best Sellers Block (parte I). 82
Figura 91. Código fuente del bloque Home Best Sellers Block (parte II). 82
Figura 92. Slider de los productos más vendidos en el frontend para versión web de MegaGameShop. 83
Figura 93. Slider de los productos más vendidos en el frontend para versión Smartphone de MegaGameShop. 83
Figura 94. Código fuente del bloque Home Brands. 84
Figura 95. Slider de Nuestras Marcas en el frontend para versión web de MegaGameShop. 84
Figura 96. Slider de Nuestras Marcas en el frontend para versión Smartphone de MegaGameShop.

..... 85

Figura 97. Código fuente de política de privacidad y cookies. 85

Figura 98. Política de privacidad y cookies en el frontend para versión web de MegaGameShop. 86

Figura 99. Política de privacidad y cookies en el frontend para versión Smartphone de MegaGameShop. 86

Figura 100. Administración de cupones de descuento en el backend. 87

Figura 101. Administración de métodos de envío en el backend. 88

Figura 102. Administración de métodos de pago en el backend. 88

Figura 103. Vista de métodos de pago en el frontend para versión web de MegaGameShop. ... 89

Figura 104. Vista de métodos de pago en el frontend para versión Smartphone de MegaGameShop. 90

Figura 105. Prueba de agregar un producto al carrito en Selenium. 91

Figura 106. Prueba de agregar un producto al carrito en el frontend de MegaGameShop. 91

Figura 107. Código fuente de la prueba de agregar un producto al carrito en Selenium. 92

Figura 108. Prueba de agregar un producto a la lista de deseos en Selenium. 92

Figura 109. Prueba de agregar un producto a la lista de deseos en el frontend de MegaGameShop. 93

Figura 110. Código fuente de la prueba de agregar un producto a la lista de deseos en Selenium. 93

Figura 111. Prueba de aplicar cupón de descuento a un producto con éxito en Selenium. 94

Figura 112. Prueba de aplicar cupón de descuento a un producto con éxito en el frontend de MegaGameShop. 95

Figura 113. Código fuente de la prueba de aplicar cupón de descuento a un producto con éxito en Selenium. 95

Figura 114. Prueba de aplicar cupón de descuento a un producto sin éxito en Selenium. 96

Figura 116. Código fuente de la prueba de aplicar cupón de descuento a un producto sin éxito en Selenium. 97

Figura 117. Prueba de iniciar sesión con éxito en Selenium. 97

Figura 118. Prueba de iniciar sesión con éxito en el frontend de MegaGameShop. 98

Figura 119. Código fuente de la prueba de iniciar sesión con éxito en Selenium. 98

Figura 120. Prueba de inicio de sesión con formato de email erróneo en Selenium. 99

Figura 121. Prueba de inicio de sesión con formato de email erróneo en el frontend de MegaGameShop. 99

Figura 122. Código fuente de la prueba de inicio de sesión con formato de email erróneo en Selenium. 100

Figura 123. Prueba de inicio de sesión con email o contraseña errónea en Selenium. 100

Figura 124. Prueba de inicio de sesión con email o contraseña errónea en el frontend de MegaGameShop. 101

Figura 125. Código fuente de la prueba de inicio de sesión con email o contraseña errónea en Selenium. 101

Figura 126. Prueba de creación de cuenta de cliente en Selenium. 102

Figura 127. Prueba de creación de cuenta de cliente en el frontend de MegaGameShop. 102

Figura 128. Código fuente de la prueba de creación de cuenta de cliente en Selenium. 103

Figura 129. Prueba de compra de cliente registrado en Selenium. 104

Figura 130. Prueba de compra de cliente registrado en el frontend de MegaGameShop. 104

Figura 131. Factura de pedido de compra de cliente registrado de MegaGameShop. 105

Figura 132. Código fuente de la prueba de compra de cliente registrado en Selenium. 106

Figura 133. Prueba de compra de cliente invitado en Selenium. 107

Figura 134. Prueba de compra de cliente invitado en el frontend de MegaGameShop. 107

Figura 135. Código fuente de la prueba de compra de cliente invitado en Selenium. 108

Agradecimientos

Me gustaría agradecer a mis familiares el apoyo ofrecido y la confianza depositada durante este trayecto, para poder llegar al final de este Grado en Ingeniería Informática.

A mis profesores, por estar ahí siempre para cualquier duda que haya tenido durante el desarrollo de las asignaturas.

Dar las gracias a compañeros y amigos, que siempre me han prestado su ayuda desinteresada durante toda esta etapa.

Por último, agradecer a mi tutor del Trabajo de Fin de Grado, José Ignacio Gómez Espínola, por ayudarme durante el desarrollo del mismo.

CAPÍTULO 1. INTRODUCCIÓN.

1.1. Introducción al proyecto.

El siguiente Trabajo de Fin de Grado consiste en el desarrollo de una tienda online mediante el uso de un CMS (sistema de gestión de contenido).

Esta web, de creación propia, está orientada a la venta de videojuegos (consolas, videojuegos, accesorios, merchandising, etc). Su nombre es MegaGameShop.

El usuario podrá registrarse en la página web mediante un formulario o iniciar sesión si ya dispone de una cuenta.

El usuario podrá navegar por los distintos productos, ordenados por distintas categorías. Cada uno de los productos ofertados por la web dejará visualizar los detalles y descripción de los mismos, así como sus valoraciones y opiniones dadas previamente por otros clientes que compraron el mismo producto. También permitirá añadir o quitar productos del carrito y esta se permitirá la compra de los mismos pagando a través de Paypal, contrareembolso, transferencia bancaria y tarjeta de crédito o débito.

Por último, se calculará si el pedido tiene gastos de envío o no, dependiendo del importe a pagar por el cliente. Este límite estará establecido previamente por el gestor de la tienda online.

La web podrá ser utilizada en dispositivos como ordenadores, smartphones o tablets, ajustándose la información dada por la web al tamaño de la pantalla del cliente. También se le ofrecerá la posibilidad de poder visualizarla en 2 idiomas, inglés y español.

1.2. Motivación.

Con el transcurso de los años, las personas cada vez optan más por adquirir productos a través de Internet. Esto se produce debido a que es muy cómodo adquirir un producto sin importar el día o la hora, con tan solo una conexión de red y un dispositivo que nos permita navegar por internet. Este producto llegaría a nuestro domicilio en apenas unos días dependiendo de dónde lo adquiramos. Este producto dispondría de una garantía idéntica a la que tendría si lo adquiriéramos en un establecimiento físico: siempre podría ser devuelto si presenta algún fallo de fabricación o si se trata de alguna prenda que no es de nuestra talla entre otros casos. Esta devolución nos permitiría devolvernos el importe del artículo, descambiarlo por otro igual o similar, etc., siempre dependiendo de la política de la empresa.

El comercio electrónico es la vía principal que tienen los negocios para abrir nuevos mercados y aumentar sus ventas. Según datos recogidos en 2020 por la CNMC (Comisión Nacional de los Mercados y Competencia), el comercio online creció en España más del 20%, sobre todo en los sectores de moda y aparatos electrónicos. [1]

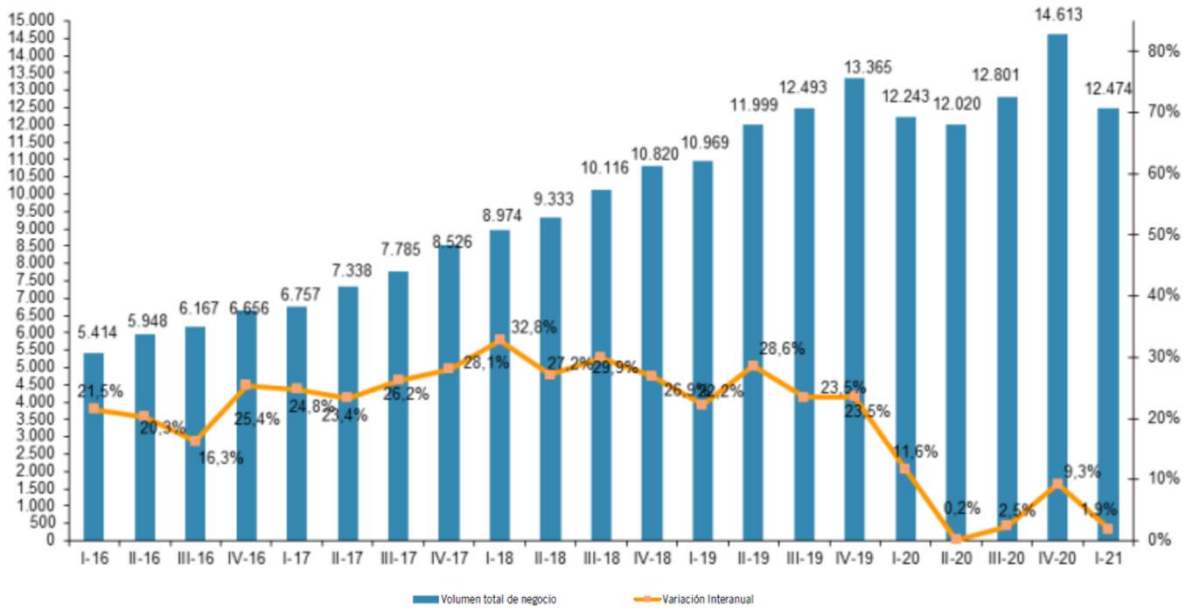

Figura 1. Evolución trimestral del volumen de negocio del comercio electrónico y variación interanual.

Como podemos ver en la Figura 1, la evolución del volumen de negocio del comercio electrónico tiende a crecer cada trimestre con el paso de los años, por lo que cada vez es una manera más recurrente para el consumidor de adquirir un producto.[2]

Respecto al sector de los videojuegos, según Newzoo[1], España se sitúa entre los diez países con mayor consumo de videojuegos en 2019. Durante este año, los datos de facturación fueron de 1479 millones de euros, donde se obtuvo una cifra récord en ventas online.[3]

Por estos motivos, vi una oportunidad de negocio la creación de una tienda online basada en videojuegos, ya que este sector en España es bastante puntero y las ventas online cada vez tienen mayor peso en nuestra sociedad.

Además, la creación de una empresa online puede ayudarme a obtener conocimiento de cara a mi futura carrera profesional al terminar este Grado de Ingeniería Informática.

1.3. Objetivos.

El objetivo más importante de este Trabajo de Fin de Grado es la creación de una tienda online orientada a la venta de videojuegos mediante el uso de un CMS.

Para realizar este proyecto debemos ir completando los siguientes objetivos:

- Selección de una empresa, en este caso estará basada en videojuegos.
- Búsqueda de información acerca del mercado de los videojuegos y sobre el

[1] Newzoo: fuente más confiable y citada del mundo de información y análisis del mercado de juegos.

comercio electrónico en España.

- Comparar los distintos servicios existentes de alojamiento web y elección de uno de ellos.
- Comparar los distintos CMS (sistema de gestión de contenido) y elección de uno de ellos.
- Instalar el CMS seleccionado en el servicio de alojamiento web escogido, para el posterior desarrollo de la tienda online.
- Realización de la tienda online de manera que pueda ser visualizada correctamente en cualquier tipo de dispositivo electrónico.
- Desarrollo de la tienda online en 2 idiomas: inglés y español.
- Elección del contenido del que va a disponer la tienda.
- Permitir el inicio de sesión de los distintos usuarios o registrarse en la página web.
- Calcular los gastos de envío del pedido de manera correcta.
- Poder pagar a través de la tienda online mediante distintos métodos de pago: contrareembolso, transferencia bancaria, tarjeta de crédito/débito o Paypal.
- Generar una factura por cada compra realizada por un cliente.
- Realización de pruebas de usuario (compras de productos, pagos, inicio de sesión, creación de usuario).
- Realización de un documento a modo de memoria, que incluya la información del estudio previo y el desarrollo de la tienda online, así como las pruebas realizadas sobre su funcionamiento.

CAPÍTULO 2. ESTADO DEL ARTE.

Este capítulo lo vamos a dedicar a realizar una comparación entre algunos de los distintos CMS más fundamentales que son utilizados en las empresas, así como la elección del servicio de alojamiento web utilizado para el Trabajo de Fin de Grado.

Después de esta comparación, se expondrán los motivos para la elección de dicho CMS.

2.1. CMS.

Un CMS (Content Management System) es una aplicación web que nos permite, de manera más sencilla, la creación de un sitio web. Además, permite administrar, gestionar y realizar el mantenimiento del sitio web sin la necesidad de utilizar herramientas externas, dentro de un entorno integrado.

Dispone de una interfaz donde podemos controlar tanto una como varias bases de datos que son utilizadas para alojar todo el contenido del que va a disponer el sitio web. [4]

2.2. Características de un CMS.

Cualquiera de los CMS existentes deben cumplir con las siguientes características:

- Crear, modificar o eliminar contenido del sitio web (textos, imágenes, vídeos...).
- Moderación de los comentarios realizados por un cliente.
- Control del inventario y sistemas de ventas de los productos ofertados en el sitio web.
- Instalación de plugins para aumentar las funcionalidades del sitio web.

2.3. Ventajas e inconvenientes de un CMS.

Una vez vistas las características de un CMS, vamos a enumerar las ventajas e inconvenientes que nos encontramos al utilizar estos gestores de contenido.

2.3.1. Ventajas de un CMS.

Las ventajas más destacadas son las siguientes:

- Facilidad de uso para que el usuario pueda crear, modificar o actualizar contenido para el sitio web de manera intuitiva.
- Gran personalización y flexibilidad de la configuración y opciones del programa.
- Fácil mantenimiento para la mayoría de los CMS.
- Menos costoso que realizar un sitio web desde cero.
- Desarrollo del sitio web de manera más rápida, ya que muchas

funcionalidades están ya.

- Soporte y documentación, como medio de ayuda al usuario.

2.3.2. Inconvenientes de un CMS.

Los inconvenientes más destacados son los siguientes:

- Gran gasto de recursos del servidor en algunos CMS, que pueden llegar a provocar la caída del servicio.
- Grandes vulnerabilidades en seguridad si no se realiza un mantenimiento adecuado del CMS.
- Algunos CMS son mantenidos por la comunidad, por lo que el soporte puede ser difícil de encontrar.
- Algunos temas o plugins pueden ser de pago. [5]

2.4. Tipos de CMS.

Existen una gran cantidad de CMS, pero nos vamos a centrar en los más importantes que sean más específicos para el comercio electrónico. Estos son los más utilizados a día 10 de marzo de 2022 según la web Wappalyzer:[2] [6]

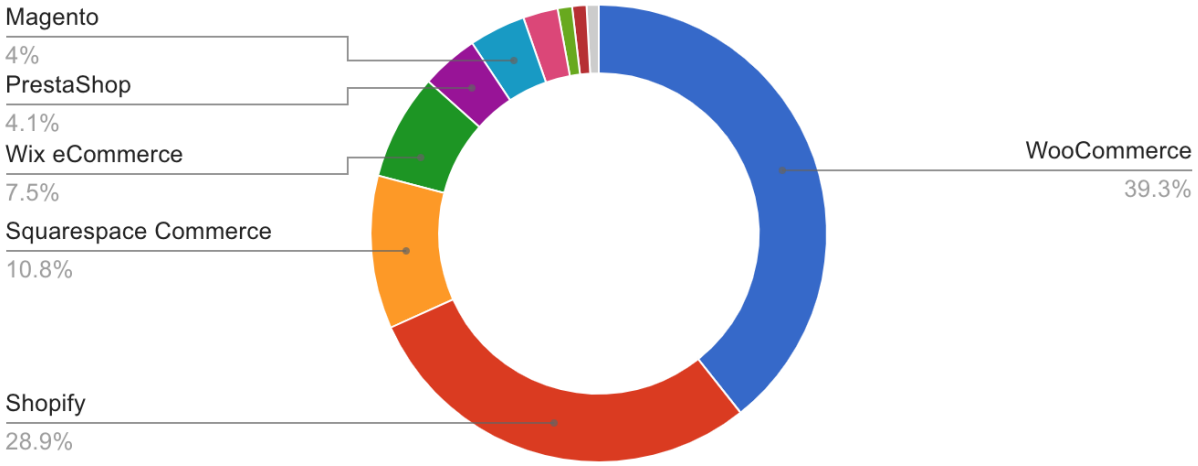

Figura 2. CMS más utilizados.

A continuación, vamos a detallar cada uno de estos CMS, de mayor a menor importancia:

<u>WooCommerce</u>

[2] Wappalyzer: Sitio web dedicado a indicar qué tecnología utiliza una determinada web.

Es un plugin de código abierto CSS, HTML, JS y PHP para WordPress. Es el más utilizado en la actualidad, con un 39.3% del total de sitios web dedicados al comercio electrónico.

Figura 3. Logotipo de WooCommerce.

WooCommerce proporciona de funcionalidades específicas ecommerce y destaca por su facilidad de instalar y la capacidad de personalizar la base de datos de los productos.

Ventajas

- Completamente gratuito y al ser Open Source, se pueden modificar sus archivos base según las necesidades para la tienda online.
- Fácilmente adaptable a la imagen de nuestro sitio web.
- Proporciona seguridad y privacidad en las ventas realizadas.
- Gran cantidad de tutoriales, guías y documentación sobre cómo realizar nuestro sitio web.
- Extensiones y plugins de terceros, que nos ofrecen mayores funcionalidades aún.
- Sencillo e intuitivo, tanto para el cliente como para el administrador del sitio web. Podrán administrarla personas sin conocimiento en programación.
- Automatizaciones de todo tipo como, por ejemplo, enviar email de agradecimiento al cliente tras adquirir un producto.
- Integración con Google Analytics para análisis de productos y ventas.

Desventajas

- Velocidad de carga lenta del sitio web si se utilizan demasiadas extensiones o plugins.
- Los plugins de calidad son de pago anual, por lo que utilizar este CMS no es completamente gratuito. [7]

Shopify

Es el segundo CMS más utilizado, con un 28.9% de cuota. Fue creado en el año 2004 y permite diseñar una tienda online sin necesidad de poseer conocimientos de programación.

Figura 4. Logotipo de Shopify.

Presenta una gran flexibilidad en su interfaz y como característica principal, permite la creación de la tienda de manera bastante intuitiva y eficaz.

Para comenzar a crear la web, basta con rellenar los datos que nos solicite la web, seleccionar un plan de pago y la plantilla (de pago o gratuita) sobre la que vamos a crear nuestro sitio web.

Existen 3 planes de pago, que son los siguientes:

- Basic: 27 euros/mes. Para emprendedores que quieran lanzar su tienda online.
- Shopify: 79 euros/mes. Para negocios en crecimiento que venden online o en tienda física.
- Advanced: 289 euros/mes. Para negocios en expansión internacional y con altas ventas.

Además de estos precios, cada vez que un cliente realice un pago a través de tarjeta de crédito, se nos cobrará una tarifa entre un 1.8% y un 2.4%, dependiendo del plan de pago elegido.

Ventajas

- Sencillez en la creación, personalización y gestión de la tienda online.
- Servicio de hosting incluido.
- Gran servicio de atención al cliente mediante chat, foros o email.
- Sistema de impuestos automáticos, dependiendo del país a partir de la ubicación.
- Facilidades de pago.
- Estadísticas de clientes.

Desventajas

- No es de código abierto, por lo que habrá que pagar para utilizar este CMS.
- El coste es mensual.

- Algunas características son de pago, por lo que el precio al mes podría ser demasiado elevado. [8]

Squarespace Commerce

Es el tercer CMS más utilizado en la actualidad, con un 10.8% de cuota. Este sistema de gestión de contenidos ofrece la posibilidad de crear un sitio web a partir de plantillas de una forma bastante sencilla sin la necesidad de contratar un servicio de hosting.

Figura 5. Logotipo de Squarespace Commerce.

Como en el CMS Shopify, se debe realizar un pago (anual o mensual) para acceder a sus servicios. Sus tarifas anuales, que son las que rentan más económicamente, rondan entre los 132 y los 432 euros.

Ventajas:

- Servicio de hosting incluido.
- No existen plugins, por lo que la plataforma es muy estable.
- Las plantillas son bastante atractivas y profesionales.
- No requiere de conocimientos de programación.

Desventajas:

- El precio es demasiado elevado.
- Ofrece pocas alternativas de pago.
- El uso de las plantillas que ofrece puede limitar al administrador del sitio web.
- El editor no se adapta a la interfaz móvil. [9]

Wix eCommerce

Es el cuarto CMS más utilizado en la actualidad, con un 7.5% de cuota. Ofrece la posibilidad de crear un sitio web fácilmente mediante “drag & drop” (arrastrar y soltar), sin la necesidad de contratar un servicio de hosting.

Figura 6. Logotipo de Wix eCommerce

Ventajas

- Es uno de los CMS en los que más rápido se puede crear un sitio web, siendo bastante vistoso estéticamente.
- Interfaz muy intuitiva para crear el sitio web.
- Gran cantidad de plantillas.
- Cuenta con una versión gratuita, para la creación del sitio web.

Desventajas

- Es necesario pagar el dominio del sitio web, por lo que es necesario pagar una tarifa mensual.
- La versión gratuita no ofrece correo de empresa.
- Poco nivel de personalización del sitio web.
- Migración a otro CMS te hará perder todo el contenido.
- Los contenidos en este CMS son más difíciles de indexar en Google, por lo que se deberán dedicar más recursos a esta tarea.[10]

PrestaShop

Es el quinto CMS más utilizado en la actualidad, con un 4.1% de cuota. Este sistema de gestión de contenidos, de código abierto, ofrece la posibilidad de crear un sitio web de manera muy sencilla en unos minutos, de manera similar a la que se realiza con el CMS WordPress.

Figura 7. Logotipo de PrestaShop.

Ventajas

- Es OpenSource, se puede modificar su código.
- Gran facilidad de uso e instalación rápida mediante interfaz web.

- Compatibilidad con la gran mayoría de métodos de pago.
- Recibe menos ataques de ciberseguridad en comparación a otros CMS.
- Gran comunidad, soporte y cantidad de documentación.
- Posibilidad de integrar varias tiendas desde un solo administrador.
- La tienda se puede visualizar en distintos idiomas mediante traducciones automáticas.
- Gran capacidad de personalización mediante módulos.

Desventajas

- Ofrece menos variedad de plantillas que otros CMS de la competencia.
- A partir de los 10000 productos en la tienda creada, el sitio web puede volverse lento.
- Las mejores plantillas y módulos son de pago.
- Las URLs generadas son menos amigables que en otros CMS.
- Manejar todos los ámbitos de este CMS es más complejo que en otros gestores de contenido como WordPress. [11]

Magento

Es el sexto CMS más utilizado en la actualidad, con un 4% de cuota. Este sistema de gestión de contenidos, de código abierto, es considerado el CMS que más potencia nos ofrece de todo el mercado, para la creación de sitios web enfocados a la venta de productos. Ofrece la posibilidad de crear un sitio web de manera que permita gestionar de manera avanzada sus distintos sectores, como el sector del marketing y el del posicionamiento web, elementos muy a tener en cuenta a la hora de crear una tienda online.

Figura 8. Logotipo de Magento.

Ventajas

- Facilidad de instalación y uso, mediante un panel de administración.
- Gestión de la ficha de cada producto de nuestro sitio web.

- Capacidad para soportar una gran cantidad de productos (hasta 1 millón) y un gran tráfico web (hasta 80.000 pedidos por hora).
- Integración con herramientas externas como Google Analytics, para análisis de productos y ventas.
- Gran posicionamiento web a través del motor de búsqueda Google.
- Utiliza estrategias de up selling (vender los productos más caros y con mayor margen de beneficios para nuestra empresa).
- Alto nivel de seguridad para el cliente, a través de las principales pasarelas de pago de forma segura.
- Posibilidad de integrar varias tiendas desde un solo administrador.

Desventajas

- Necesita un mayor nivel de desarrollo que otros CMS, como WooCommerce o PrestaShop.
- Necesidad de contratar un servicio de alojamiento web.
- Para proyectos de tiendas de una mayor envergadura, podría ser necesario contratar versiones más avanzadas de Magento. Estas versiones permiten utilizar todas las funcionalidades y capacidades de Magento, pero son bastante caras. [12]

2.5. Elección del CMS.

Una vez vistos los CMS más utilizados en el mundo, junto a sus ventajas y desventajas, llega la hora de elegir uno de ellos para realizar el desarrollo de nuestra tienda online de videojuegos.

El CMS utilizado va a ser Magento, ya que es el gestor de contenidos más potente del mercado, y esto nos puede llegar a servir por si en un futuro decidimos llevar este proyecto al mercado laboral. Otra de las razones por las que se ha escogido este CMS es la gran capacidad de personalización modificando su código, ya que es de código abierto, lo cual es especialmente interesante para un Graduado en Ingeniería Informática.

Por último, debemos decantarnos por uno de los planes que nos ofrece Magento, dependiendo de la envergadura de nuestro proyecto o características que deseemos incluir en nuestra tienda. Estos planes son:

- Magento Community Edition: es completamente gratuita. Tiene la opción de aplicar personalizaciones limitadas para adaptar la tienda online a nuestras necesidades.
- Magento Enterprise Edition: tiene un precio desde 22.000 dólares al año. Viene con una gran cantidad de beneficios y libres personalizaciones para nuestra tienda online.
- Magento Commerce Cloud: tiene un precio desde 40.000 dólares al año. Este plan es idéntico al anterior, con la diferencia de que este nos incluye además

almacenamiento en la nube para la optimización y rendimiento de Magento tanto en el servidor web como en la base de datos. [13]

Tras ver los planes de pago, me decantaré por el plan Magento Community Edition. Aunque este plan tiene ciertas limitaciones en la personalización, es gratuito y puede servirnos para comenzar con el desarrollo de nuestra tienda online y familiarizarnos con el entorno de desarrollo.

2.6. Servicio de alojamiento web.

2.6.1. Definición.

El servicio de alojamiento web es el que provee a los usuarios un sistema para poder almacenar cualquier tipo de contenido accesible vía web (texto, imágenes, vídeo...).

2.6.2. Servicios más destacados.

Una vez elegido Magento como CMS a utilizar para el desarrollo de nuestra tienda online, es necesario contratar un servicio de alojamiento web para almacenar el contenido de la misma.

A continuación, vamos a ver las características de algunos de los servicios de hosting más destacados que existen, ordenados de manera alfabética:

Arsys

Este servicio de hosting destaca porque al utilizar direcciones IP españolas, mejora el posicionamiento web frente a sitios web del extranjero. Además, ofrece el poder filtrar posibles ataques, por lo que es bastante puntera en ciberseguridad. Su precio es económico, ya que oscila entre los 3-5 € al mes durante el primer año.

Axarnet

Entre sus características podemos destacar servidores con IP españolas para mejorar el posicionamiento web, seguridad anti-hackeo, servicio técnico 24 horas, migración de correo y web al servidor gratuita y webs de gran velocidad, debido a que utilizan discos duros SSD en sus servidores. Tras 3 meses de prueba gratuita, su precio oscila entre los 3-10 € al mes, dependiendo del plan escogido.

Cdmon

Su característica más importante es que se basa en discos duros Intel SSD NVMe (Non-Volatile Memory express), que proporcionan mayor capacidad de disco y CPU/RAM. Además, incluye certificados de seguridad SSL (Secure Sockets Layer), servicio técnico 24 horas, no tiene permanencia y utiliza IP españolas para mejorar el posicionamiento web. Su precio oscila entre los 5-50 € al mes.

Dinahosting

Su característica más importante es el Varnish Caché (reduce el tiempo de carga del sitio web, lo que conlleva a una mejora sustancial de la experiencia de usuario). También

incluye certificados de seguridad SSL, servicio técnico 24 horas y la posibilidad de cambiar de suscripción de una manera sencilla. Ofrece durante los primeros 3 meses hosting gratuito a los nuevos clientes. Su precio oscila entre los 2-12 € al mes.

Factoría Digital

Se diferencia principalmente por la capacidad de espacio que ofrece en el disco duro. Además, ofrece certificados de seguridad SSL, servicio técnico a través de chat, teléfono o email 24 horas, copias de seguridad diarias automáticas o en remoto. Después del primer mes de prueba, su precio oscila entre los 7-19 € al mes.

Hostalia

Ofrece certificado de seguridad AENOR, la posibilidad de ofrecer estadísticas gráficas web, tanto diarias, mensuales y por horas. Dependiendo del plan de pago escogido, este servicio de hosting nos ofrece entre 3 y 6 meses gratuitos. Posteriormente, su precio oscilará entre los 4-13 € al mes.

Ionos

Este servicio de hosting es uno de los más conocidos. Ofrece seguridad, escalabilidad, flexibilidad y rapidez. Durante el primer año, el dominio es gratuito. Sus planes son: 1 € al mes durante 6 meses o 1 año al principio y, posteriormente, entre 3-15 € al mes, dependiendo de los GB contratados. El más barato es de 50 GB.

ProfessionalHosting

Este servicio de hosting está especializado en el alojamiento de diversos CMS. Ofrece gran seguridad, migración de manera gratuita, servicio técnico 24 horas y alta velocidad de los sitios web. Sus planes de pago se sitúan a partir de los 5 €. [13]

2.6.3. Elección de servicio de alojamiento web.

Una vez vistos algunos de los servicios de alojamiento web más destacados, debemos elegir uno de ellos para instalar Magento y así poder llevar a cabo el desarrollo de nuestra tienda online.

Tras varias comparativas, decidí configurar mi ordenador como servidor, ya que esto me permitiría gestionar y administrar tanto la tienda como la base de datos localmente sin tener gastos económicos. De esta manera, si en un futuro decido poner esta web en marcha, solamente necesitaría contratar alguno de los servicios de alojamiento web citados anteriormente.

CAPÍTULO 3. TECNOLOGÍAS UTILIZADAS.

En este capítulo vamos a ver las distintas tecnologías que hemos utilizado para llevar a cabo la elaboración de este Trabajo de Fin de Grado.

3.1. Visual Paradigm.

Es una herramienta software que he utilizado para la creación de diagramas de casos de uso, diagramas de secuencia y diagrama de clases del proyecto. La función de estos diagramas es la de ayudar a los desarrolladores a crear el software adecuado a partir de unos determinados requisitos.

Otra de las funciones para lo que lo he utilizado para la fase de diseño de la interfaz, donde haré un prototipo de las vistas más relevantes.

Figura 9. Logotipo de Visual Paradigm.

3.2. Magento 2.

La versión utilizada en este proyecto Magento 2, en concreto la versión Magento 2.2. He utilizado esta versión debido a que las últimas versiones, como Magento 2.3 o 2.4, necesitan programas adicionales complejos para llevar a cabo su instalación y esto podría acarrear problemas futuros en el funcionamiento de nuestra tienda.

Figura 10. Logotipo de Magento 2.

3.3. XAMPP.

Es una distribución de Apache compuesta por varios softwares libres. Estos programas son: el servidor web Apache, los sistemas relacionales de administración de bases de datos MySQL y MariaDB, así como los lenguajes de programación Perl y PHP.

De esta manera, nos permitiría gestionar y administrar tanto la tienda como la base de datos de forma local.

Figura 11. Logotipo de XAMPP.

3.4. GanttProject.

Es una aplicación de escritorio multiplataforma gratuita para la programación y gestión de proyectos. La utilizaremos en nuestro proyecto para la creación de diagramas de Gantt.

Figura 12. Logotipo de GanttProject.

3.5. Selenium.

Es un framework que permite la realización de una depuración de pruebas automáticas en cualquier aplicación web. Lo utilizaremos en la fase final del proyecto, durante la fase de pruebas.

Figura 13. Logotipo de Selenium.

CAPÍTULO 4. INGENIERÍA DEL SOFTWARE.

4.1. Introducción.

En este apartado vamos a detallar paso a paso la metodología a seguir para el desarrollo de nuestra tienda online.

4.1.1. Modelos de Ingeniería del Software.

Primero, vamos a ver algunos de los modelos más destacados en Ingeniería del Software y nos decantaremos por uno de ellos.

Modelo en cascada

Es un procedimiento lineal que divide los procesos de desarrollo en sucesivas fases de proyecto. Por lo tanto, cada una de estas fases se realiza solamente una vez. Este modelo se utiliza, principalmente, en el desarrollo de software.

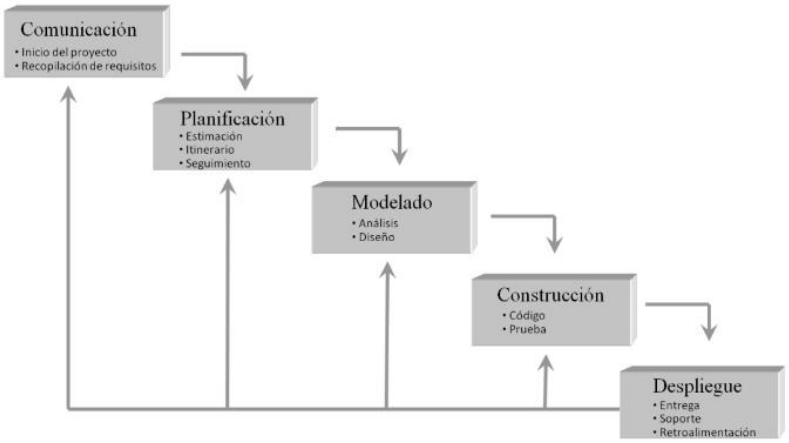

Figura 14. Modelo en cascada.

Modelo incremental

Tiene como objetivo evolucionar el producto en cada una de las entregas previstas hasta que se ajuste a los requisitos del cliente. En cada entrega, el producto debe mostrar una evolución con respecto a la fecha anterior.

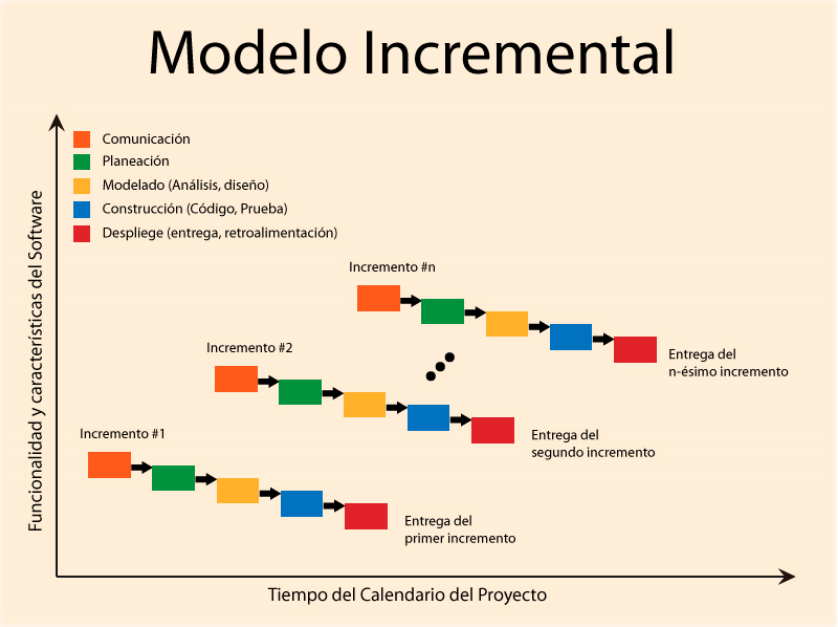

Figura 15. Modelo incremental.

Modelo en espiral

Es una combinación del modelo en cascada y el incremental. En este modelo, el equipo de desarrollo comienza con un pequeño número de requisitos y pasa por cada fase de desarrollo para ese conjunto de requisitos.

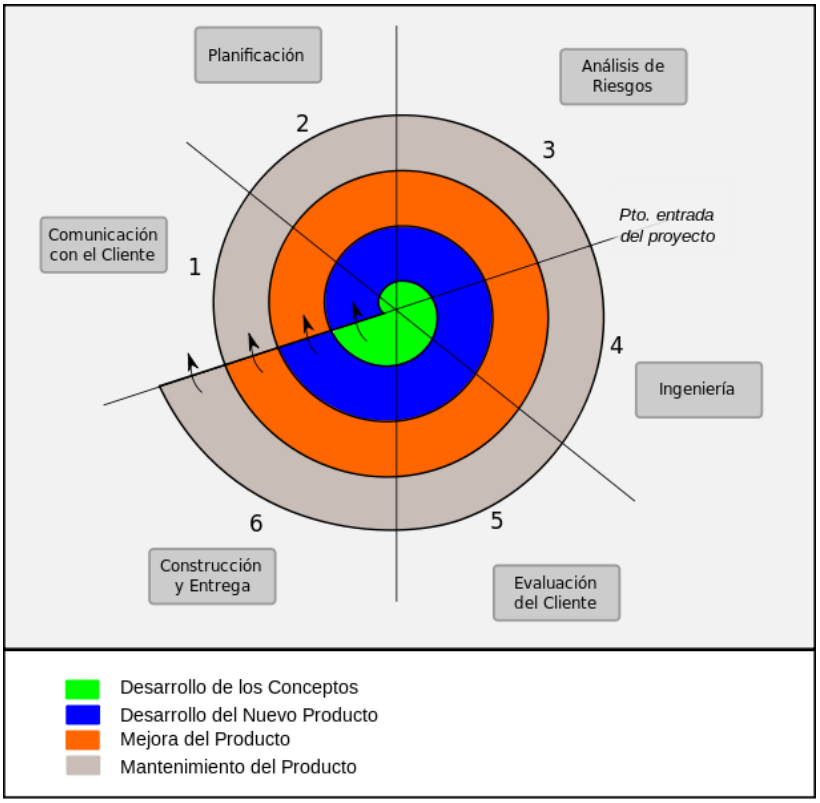

Figura 16. Modelo en espiral.

Metodologías ágiles

Es un modelo de mejora continua en el que se planifica, se crea, se comprueba el resultado y se mejora. Al contar con unos plazos de entregas tan reducidos, se centra toda la atención en una tarea determinada. Los métodos más utilizados son Scrum, XP (programación extrema) Lean y Kamban. Sus etapas se muestran de manera clara en la siguiente ilustración:

Figura 17. Etapas de las metodologías ágiles.

4.1.2. Elección del modelo de Ingeniería del Software.

Una vez explicadas las distintas metodologías utilizadas en Ingeniería del Software, he decidido utilizar el modelo en cascada para la realización de nuestro proyecto. Esto es debido a que, como la empresa es de invención propia, no necesitaba adaptarme a las necesidades de un cliente en concreto.

Tanto los requisitos como la temática de la tienda desarrollada se encontraban ya establecidos antes de comenzar, por lo que hemos ido efectuando cada una de las fases vistas anteriormente de un modelo en cascada (comunicación, planificación, modelo, construcción y despliegue).

Al finalizar cada etapa, contactaba con el tutor del proyecto para realizar una reunión y obtener un feedback que me ayudara a mejorar su desarrollo.

4.2. Definición de requisitos.

En este apartado se definen cada uno de los requisitos que se deben cumplir en el desarrollo del proyecto. Como la tienda desarrollada no dispone de un cliente real, puesto que es de invención propia, los requisitos se establecen a partir de mi experiencia en compras online.

Existen dos tipos de requisitos: funcionales y no funcionales.

4.2.1. Requisitos funcionales.

Los requisitos funcionales definen cualquier actividad que el sistema de software o sus componentes deban realizar, cumpliendo ciertas condiciones. Estos requisitos son los siguientes:

-Iniciar sesión: un usuario registrado podrá iniciar sesión en la tienda online.

-Cerrar sesión: un usuario registrado podrá cerrar sesión en la tienda online.

-Visualizar productos: el usuario podrá ver los productos de la tienda, así como su descripción, valoraciones o precio con IVA incluido.

-Buscar productos: el usuario podrá buscar productos. Además, dispondrá de filtros que le ayudarán en el proceso.

-Comparar productos: el usuario podrá realizar comparativas de productos.

-Administrar lista de deseos: un usuario registrado podrá añadir o eliminar productos de su lista de deseos

-Administrar carrito de compra: el usuario podrá añadir, modificar o eliminar productos del carrito de compra.

-Utilizar cupones de compra: el usuario podrá utilizar cupones de compra si hay alguno disponible.

-Cambiar idioma: el usuario podrá cambiar si desea visualizar la interfaz en inglés o español.

-Descargar factura de pedido: el usuario podrá descargar la factura tras realizar un pedido.

-Realizar pago: el usuario podrá realizar el pago del pedido a través de Paypal, contrareembolso, transferencia bancaria y tarjeta de crédito o débito.

-Valorar producto: el usuario podrá valorar un producto.

-Administrar cuenta: el usuario tendrá acceso a un panel de control donde podrá modificar sus datos personales, ver su historial de pedidos, descargar sus facturas y ver su lista de deseos.

-Administrar clientes: el administrador podrá añadir clientes, editar sus datos personales o eliminarlos.

-Administrar categorías: el administrador podrá añadir, editar o eliminar categorías y subcategorías de la tienda online.

-Administrar productos: el administrador podrá añadir, editar o eliminar productos de la tienda online.

-Control estadístico de los productos: el administrador podrá obtener una visualización global sobre las ventas, en las que podrá ver qué productos son los más vendidos y cuáles son los más buscados.

-Control estadístico de los clientes: el administrador podrá obtener estadísticas de sus clientes, en las que podrá ver sus pedidos y los clientes que más compras o consultas realizan.

-Administrar comentarios: el administrador podrá añadir o modificar sus comentarios, además de eliminar los existentes.

-Administrar pedidos: el administrador administrará el estado de los pedidos modificándolo hasta que lleguen al domicilio del cliente, así como poder rechazar un pedido si lo considerara necesario.

-Generar facturas de pedidos: el administrador generará una factura por cada pedido realizado por un cliente.

-Administrar cupones de compra y descuentos: el administrador podrá crear, modificar o eliminar cupones de compra que permitirán a los usuarios obtener descuentos.

-Ofrecer envíos gratuitos: el sistema ofrecerá envíos gratuitos para todos los pedidos.

-Aplicar IVA: el sistema aplicará el 21% de IVA al finalizar el pedido.

4.2.2. Requisitos no funcionales.

Los requisitos no funcionales son atributos que determinan cómo funciona un sistema. Hacen que el software funcione de manera más eficiente y representa la calidad de un sistema.

Algunos de estos requisitos son: disponibilidad, portabilidad, mantenibilidad, rendimiento, seguridad y diseño. Se detallan a continuación:

-Disponibilidad: estará disponible durante las 24 horas de cada día al ser una tienda online.

-Portabilidad: la tienda online estará disponible para ordenador, tablet y smartphone, adaptando su diseño al dispositivo que esté utilizando el cliente en ese momento (responsive design).

-Mantenibilidad: el administrador podrá modificar de manera sencilla tanto el contenido como el catálogo de la tienda online a través del panel de administración de la misma.

-Rendimiento: el usuario deberá obtener tiempos de respuesta relativamente bajos en

la navegación por las distintas vistas de la tienda online.

-Seguridad: los datos personales de cada cliente, así como los datos de pago, pedidos y facturas deberán almacenarse de manera segura, de tal forma que estos datos no puedan ser perdidos o vulnerados, ya que perderíamos la confianza de nuestros clientes. Estos datos solamente pueden ser accesibles a través del propio cliente y por el administrador de la tienda online.
Por otro lado, al backend solamente podrá acceder el administrador del sitio web, que es la parte donde se lleva a cabo la gestión y el desarrollo de la web. Y el frontend que es la web que ofrecemos a los usuarios, será visible para cualquier usuario, esté o no registrado.

-Diseño: la interfaz debe ser clara e intuitiva, para que cualquier usuario pueda navegar por las distintas vistas sin ningún tipo de complejidad. Además, el diseño de la web debe ser acorde al tipo de productos que estamos vendiendo (en este caso videojuegos).

4.3. Planificación.

En este apartado, realizaremos una estimación dinámica de los tiempos empleados en cada tarea durante este proyecto, y una estimación del coste que nos supondría realizar este proyecto.

4.3.1. Estimación de tiempos.

En la siguiente figura, se muestra una estimación del tiempo que nos llevado realizar cada una de las tareas de este proyecto:

Tarea	Duración en horas
Búsqueda de información	20
Estudio del CMS Magento	30
Estudio e instalación del servidor web y base de datos	15
Análisis del proyecto	35
Diseño del proyecto	45
Implementación y desarrollo del proyecto	180
Realización de pruebas	20
Realización de la documentación	60
Total	**405**

Figura 18. Estimación de tiempos.

Como podemos ver, hemos tardado un total de 405 horas en la realización de este proyecto. El proyecto comenzó el día 3 de marzo de 2022 y finalizó el día 27 de junio de 2022. Cada día le hemos dedicado unas 5 horas de media a este proyecto, sin contar fines de semana (sábados y domingos).

A continuación, se muestra un diagrama de Gantt, en el cual podemos ver el orden en el que se han ido ejecutando las tareas del proyecto, así como su fecha de inicio y fin y lo que tarda en realizarse cada una de ellas:

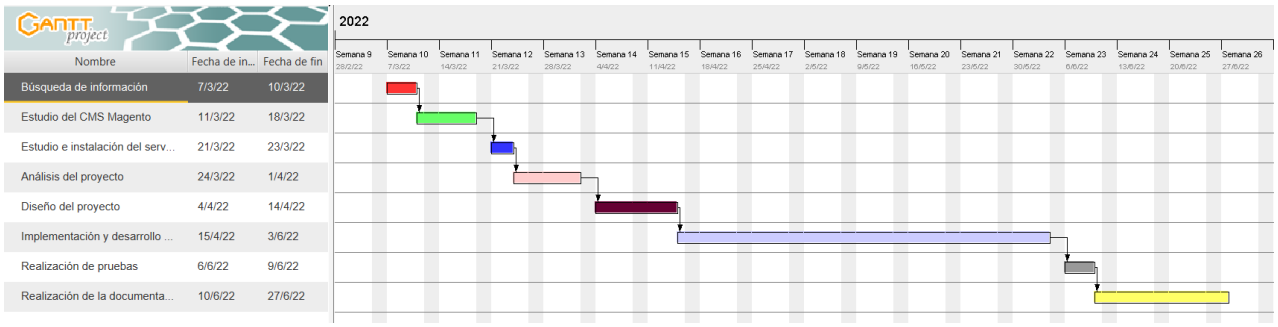

Figura 19. Diagrama de Gantt.

4.3.2. Estimación de costes.

En este apartado vamos a realizar una estimación del coste del desarrollo de este proyecto.

En este caso, estos costes son:

-Coste del personal: este proyecto se ha llevado a cabo por un único desarrollador. Realizando un estudio, vemos que un programador web cobra entre 15 y 90 € la hora, dependiendo de la complejidad del proyecto. Al ser nuestro primer proyecto y haber dedicado mucho tiempo a la búsqueda de información, cobraremos la hora 12 €. Como a este proyecto le hemos dedicado 405 horas, esto nos daría un total de 4860 €. [14]

-Coste recursos hardware y software: al no tener que contratar ningún servicio de Alojamiento Web y tener que utilizar nuestro ordenador como servidor local, esto nos permite ahorrar en este apartado, por lo que el gasto en recursos hardware será de 0 €. En cuanto al software utilizado, todo el software utilizado es libre, por lo que no requiere ningún coste para el desarrollo del proyecto.

A continuación, se muestra una estimación de costes del desarrollo del proyecto de una manera más clara:

	Coste
Coste del personal	12 € x 405 horas = 4860 €
Coste de recursos hardware	0 €
Coste de recursos software	0 €
Coste total	**4860 €**

Figura 20. Estimación de costes.

El coste estimado total del proyecto es de 4860 €.

4.4. Análisis.

En este apartado se debe definir claramente el problema que se quiere resolver o el sistema que se desea crear, así como identificar los principales componentes que formarán

el producto. Para ello, deberemos crear diagrama de clases y de casos.

4.4.1. Diagrama de clases.

En el siguiente diagrama se muestran las clases incluidas en este proyecto, así como las relaciones entre ellas y su multiplicidad. También se muestran algunos de los atributos de cada clase.

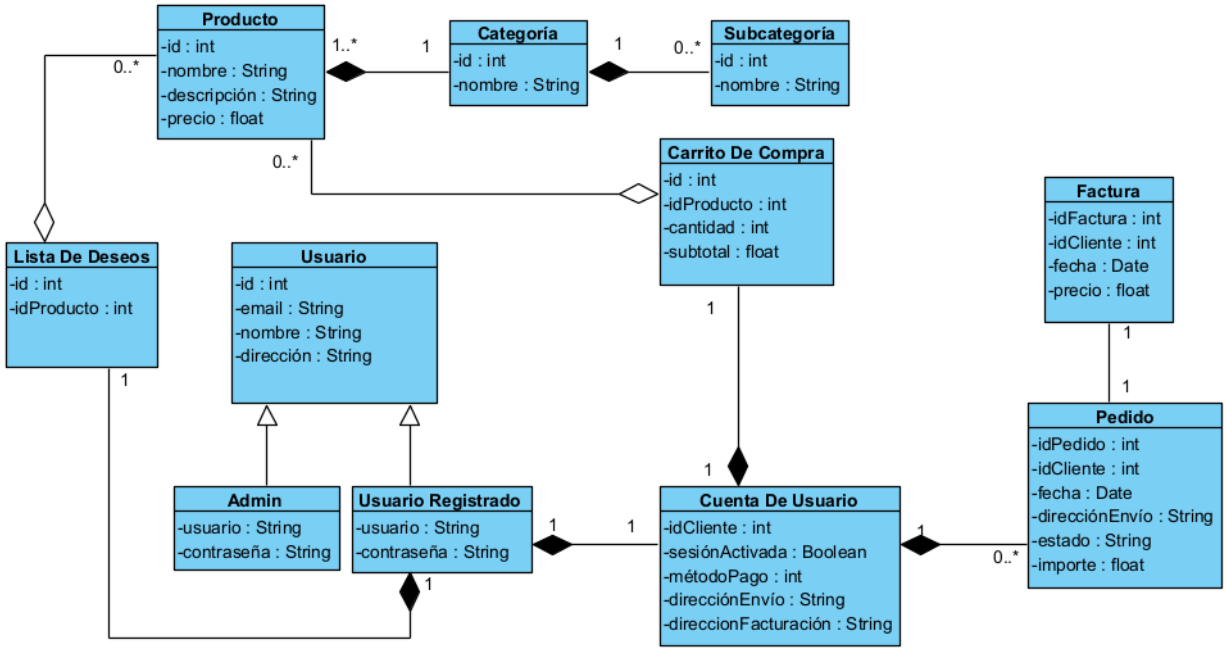

Figura 21. Diagrama de clases.

4.4.2. Diagramas de casos de uso.

En este apartado, veremos los distintos diagramas de casos de uso, según los requisitos funcionales expuestos anteriormente. Los casos de uso son los encargados de definir la comunicación, interacción y el comportamiento entre el sistema y sus usuarios. Primero, veremos uno general con las distintas funciones de la tienda, dependiendo del tipo de usuario que se encuentra interactuando con el sistema.

Diagrama general de casos de uso del cliente

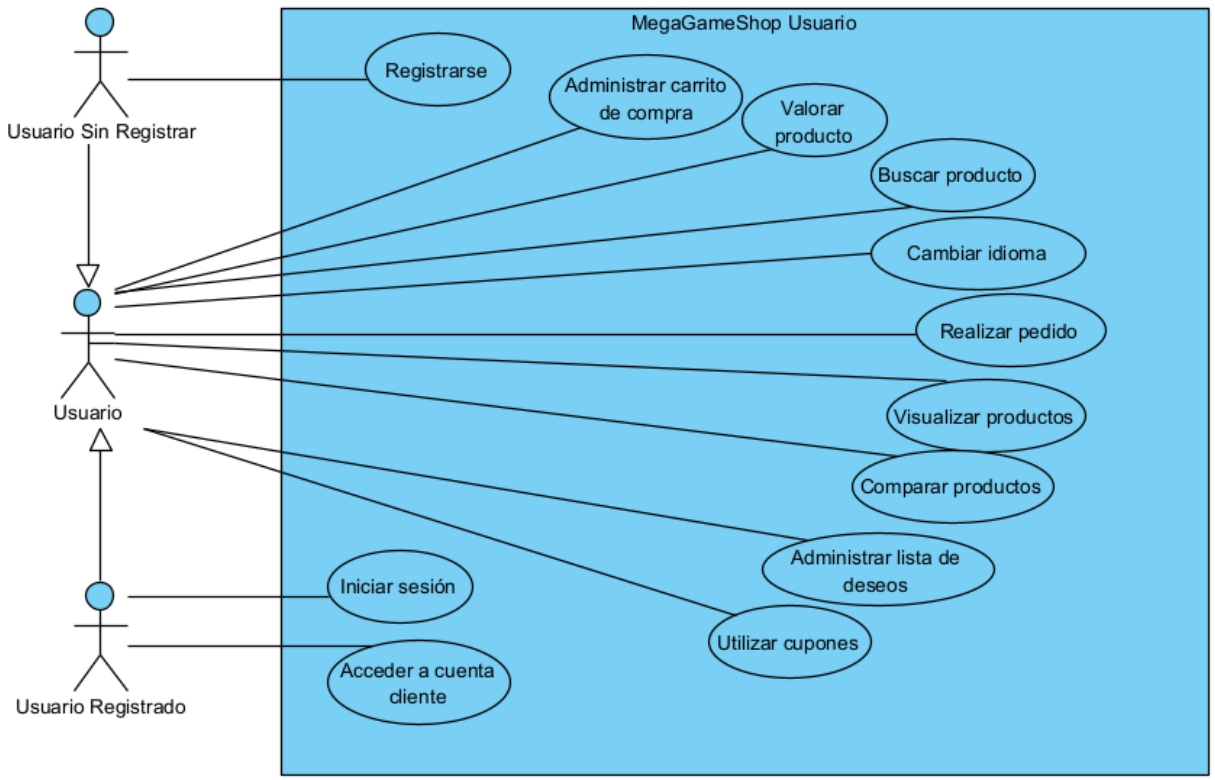

Figura 22. Diagrama general de casos de uso del usuario.

Diagrama general de casos de uso del administrador

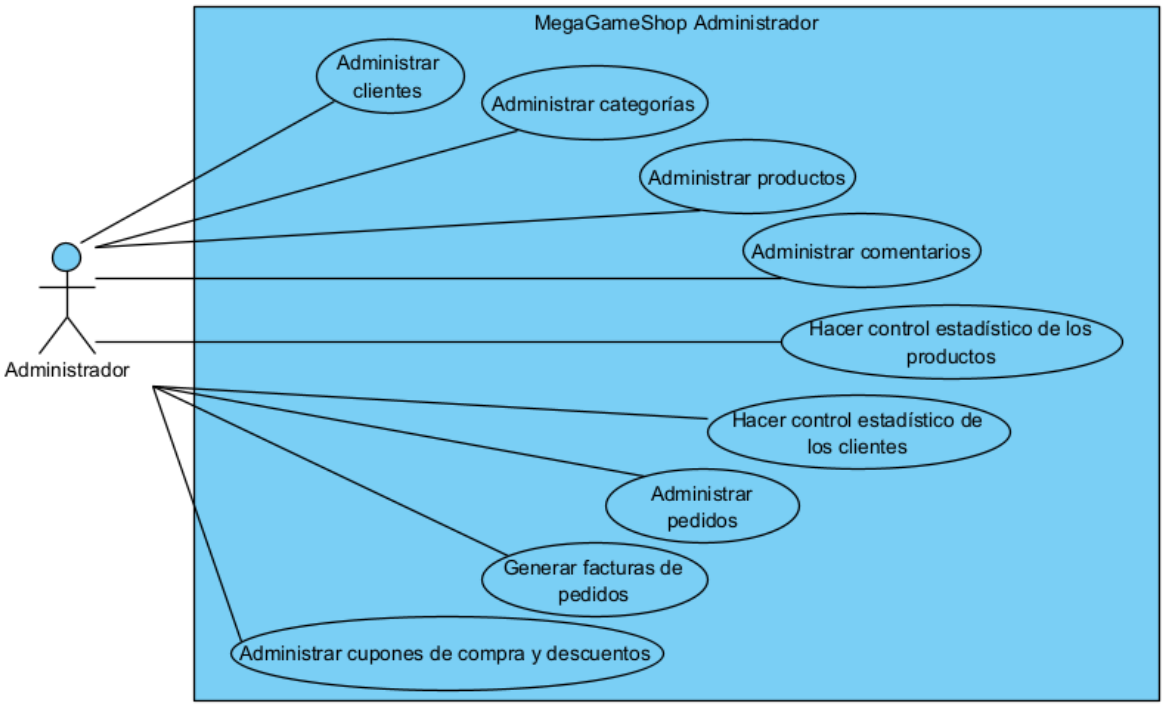

Figura 23. Diagrama general de casos de uso del administrador.

4.5. Diseño.

En esta sección, tras la información obtenida en la etapa de análisis, tenemos que desarrollar la etapa de diseño.

En esta etapa, debemos desarrollar un modelo o las especificaciones para el producto. Esto lo haremos mediante diagramas de secuencia y posteriormente, mostraremos el diseño de la interfaz de nuestra web.

4.5.1. Diagramas de secuencia.

Estos diagramas muestran los mensajes que se intercambian los objetos que existen en el sistema hasta que su línea de vida finaliza.

Diagrama de secuencia de registro de cliente

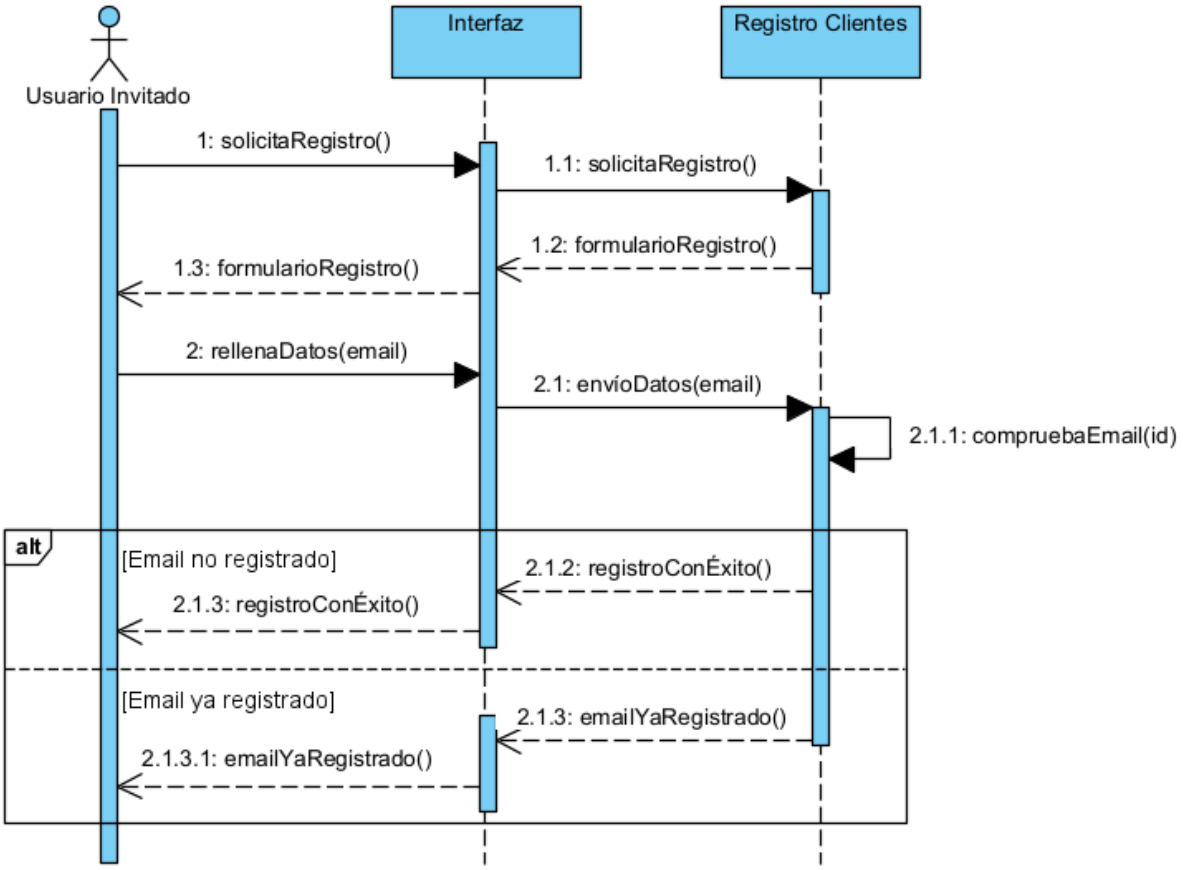

Figura 24. Diagrama de secuencia de registro del cliente.

Diagrama de secuencia de búsqueda de productos

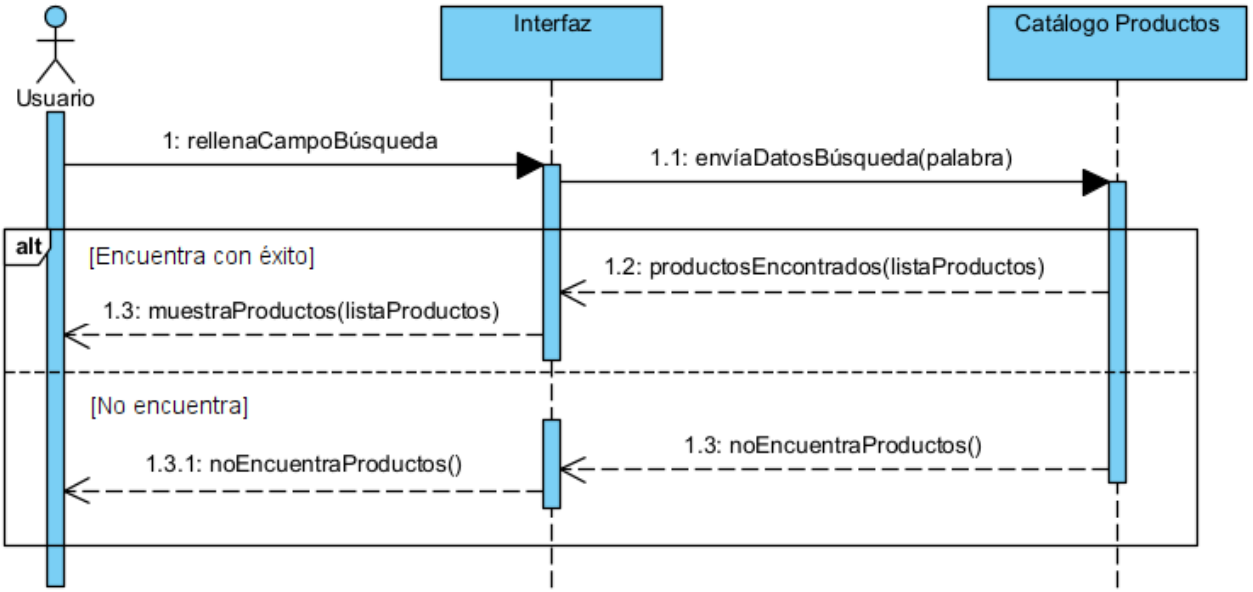

Figura 25. Diagrama de secuencia de búsqueda de productos.

Diagrama de secuencia de inicio de sesión

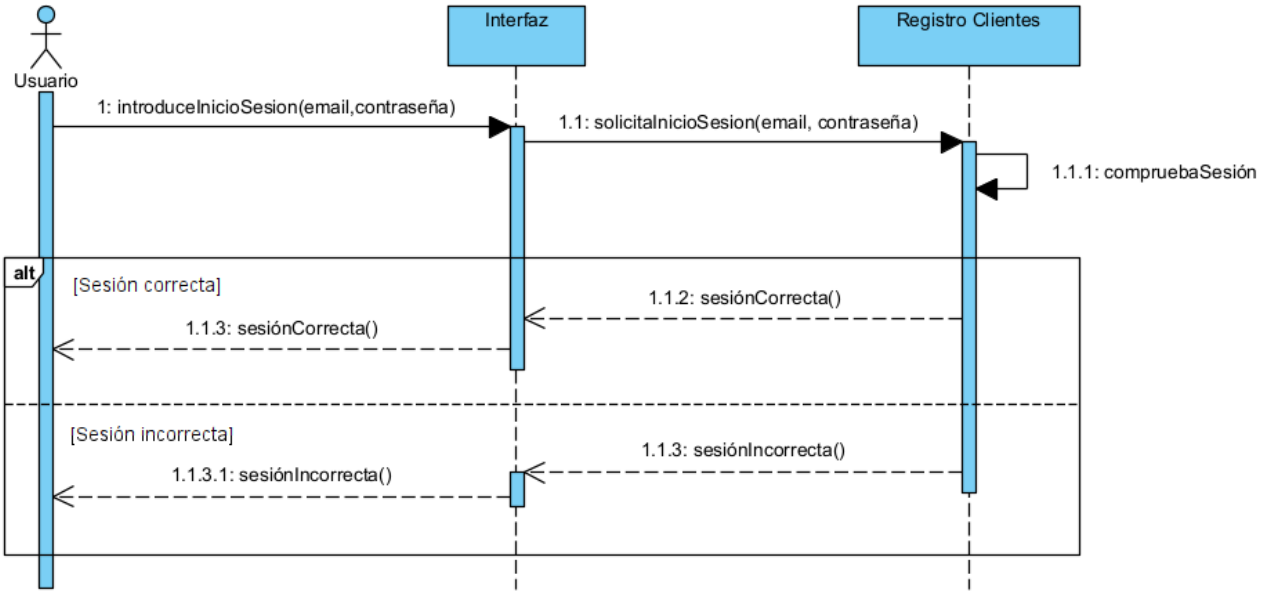

Figura 26. Diagrama de secuencia de inicio de sesión.

Diagrama de secuencia de añadir producto al carrito

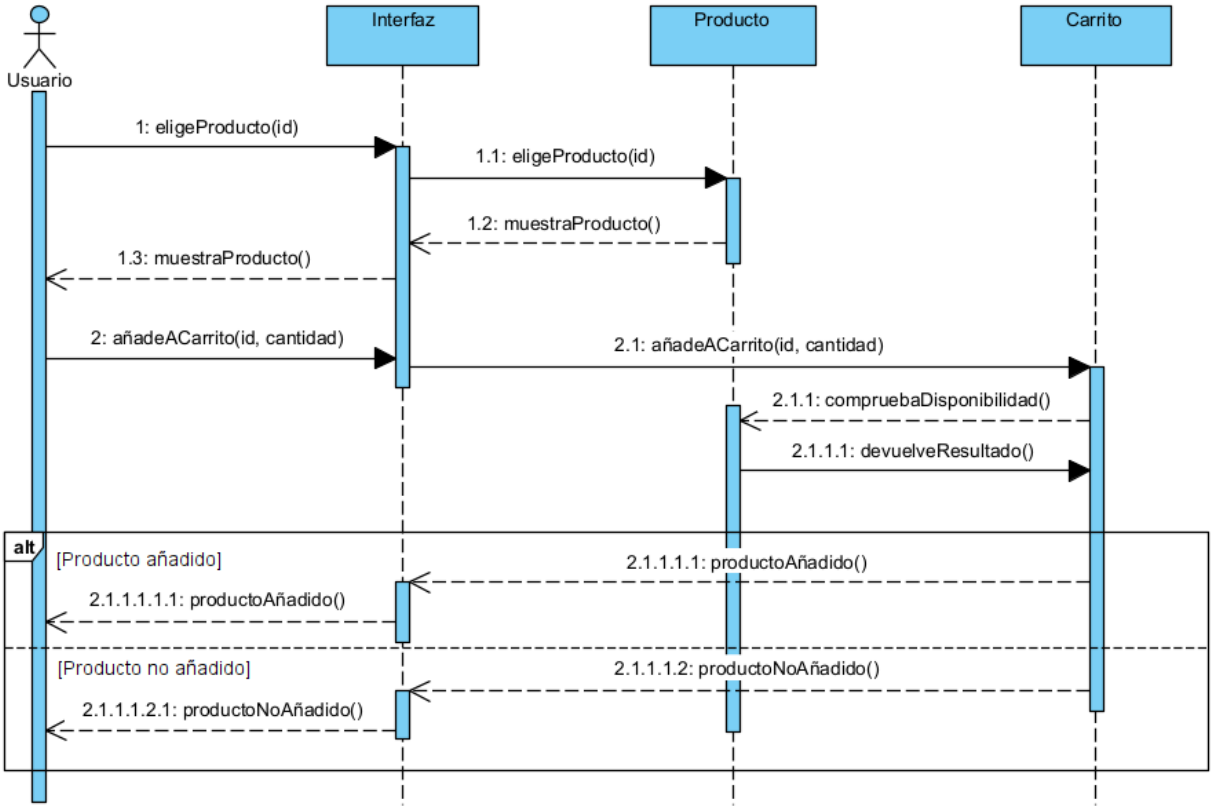

Figura 27. Diagrama de secuencia de añadir producto al carrito.

Diagrama de secuencia de realizar un pedido

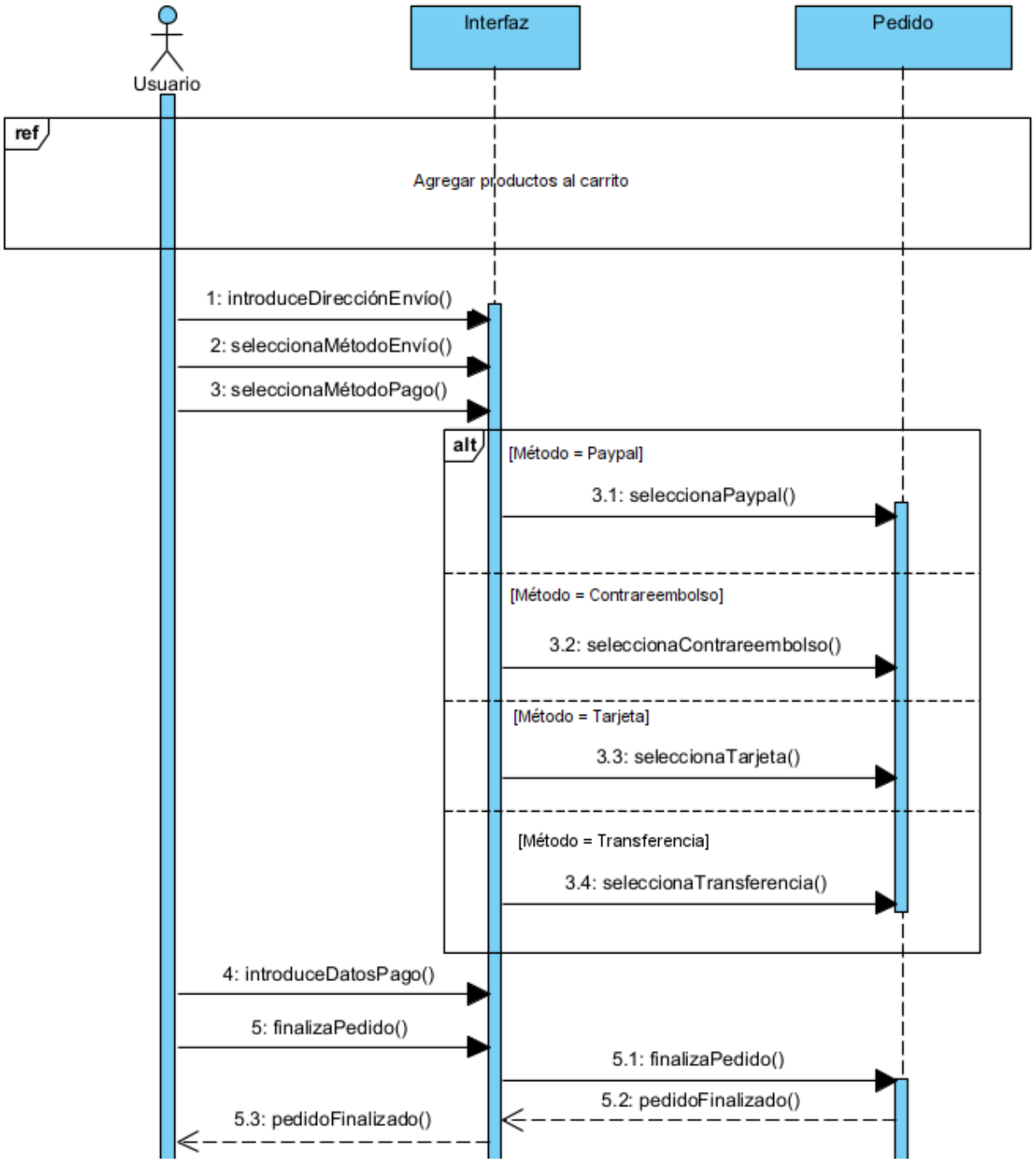

Figura 28. Diagrama de secuencia de realizar un pedido.

4.5.2. Diseño de la interfaz.

En este apartado, se han realizado prototipos con Visual Paradigm, donde vemos algunas de las vistas de la tienda de videojuegos que hemos creado para el cliente.

Vista principal del cliente

Primero, vamos a ver en la siguiente figura la vista principal que se muestra al usuario:

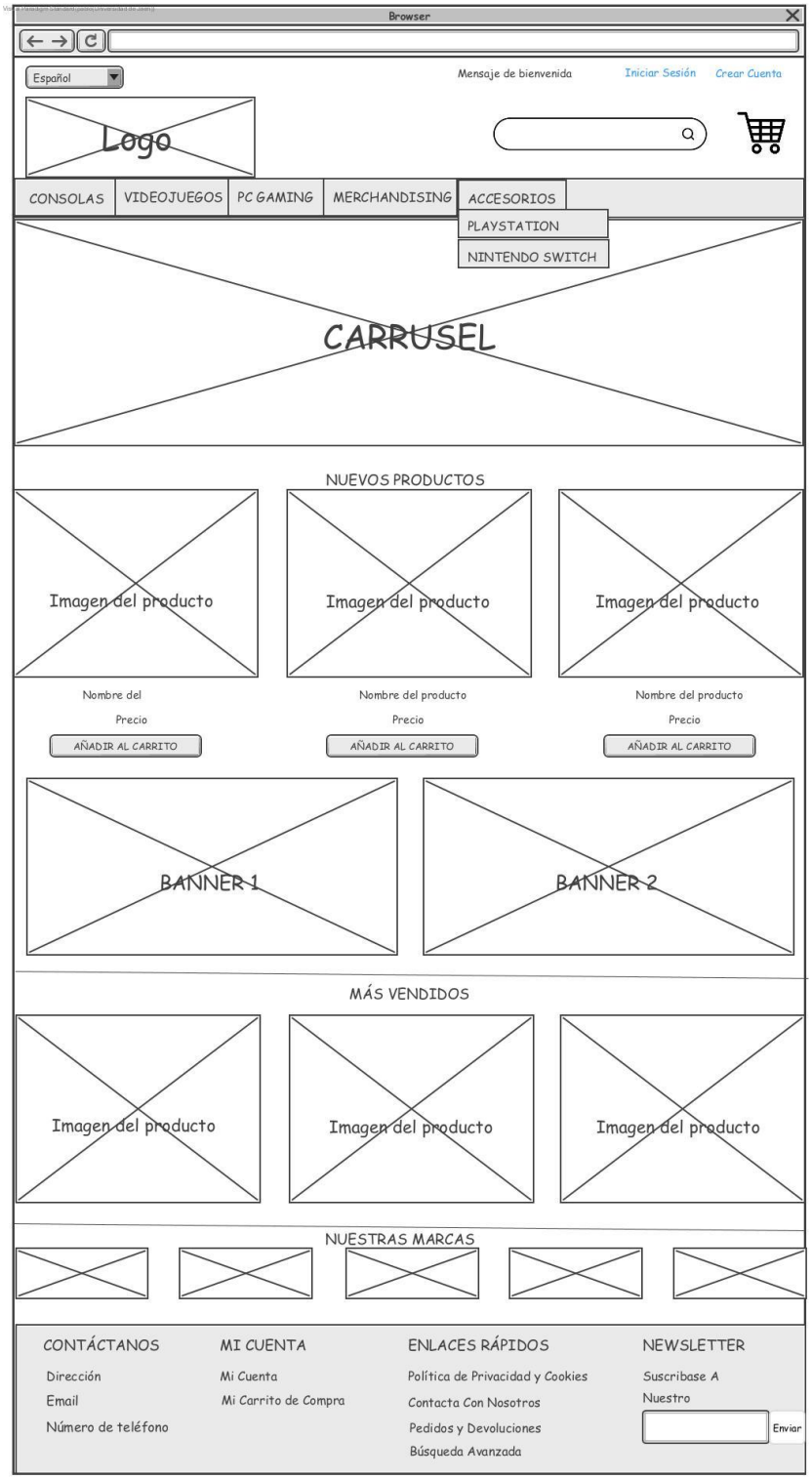

Figura 29. Prototipo de la vista principal de MegaGameShop.

En la parte superior vemos una cabecera, donde podremos: iniciar sesión, crear cuenta, cambiar de idioma, buscar productos, gestionar el carrito, ver el logo de la empresa y las distintas categorías que ofrecemos.

Debajo de las categorías, podemos ver un carrusel con 3 componentes y los últimos productos añadidos a la tienda.

En la parte media de la vista principal, vemos 2 banner publicitarios y debajo los productos más vendidos.

En la última parte de la vista principal, vemos un slider con las principales marcas que trabajan con nosotros, así como un pie de página con datos de contacto e información y política de la empresa. También se dispone de un apartado donde el cliente podrá suscribirse al boletín de noticias.

Al tratarse de una tienda responsive design, debemos realizar un diseño que se adapte al tamaño de pantalla de un Smartphone. En la siguiente figura, vemos el diseño de la vista principal y el menú lateral desplegado:

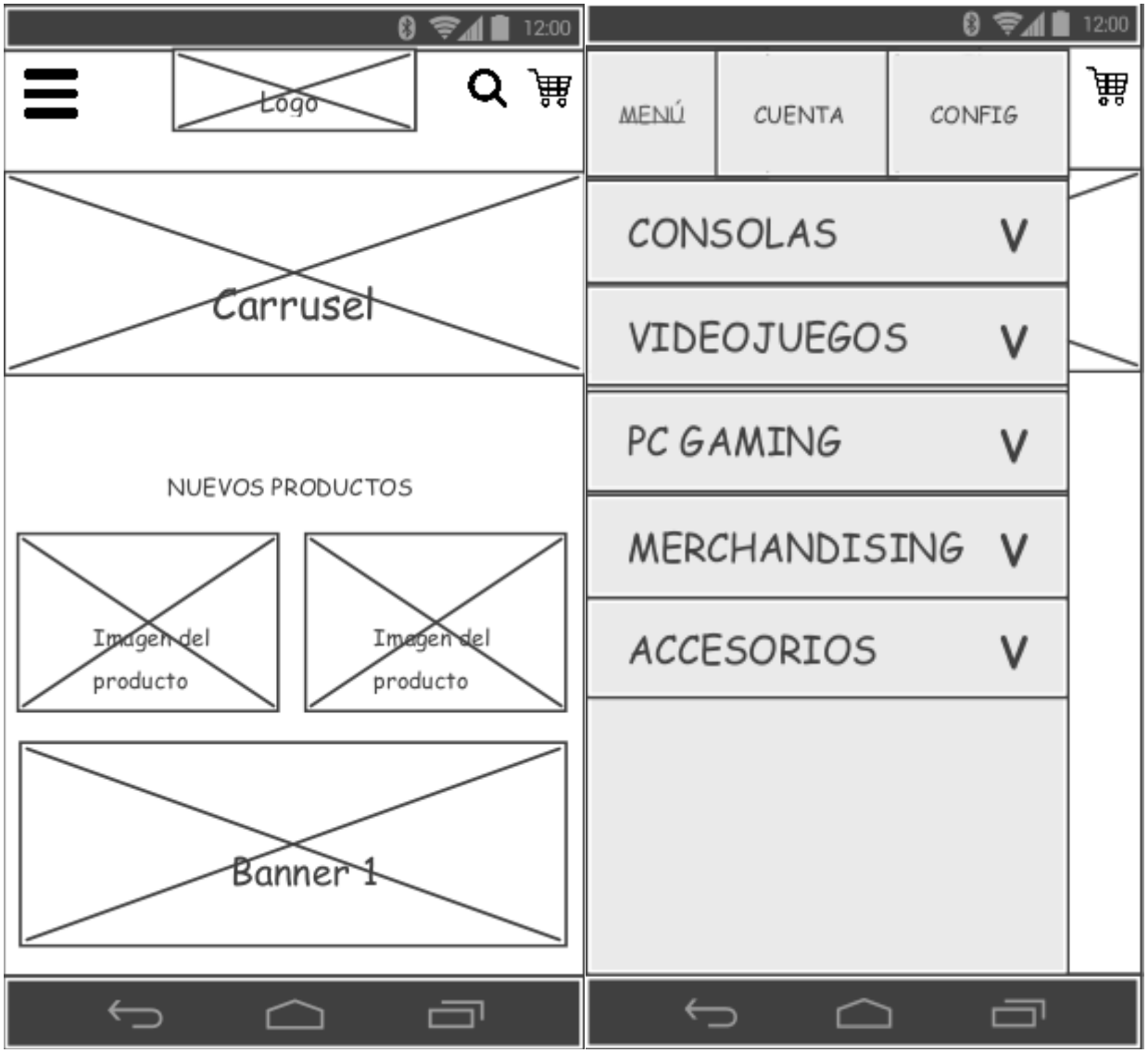

Figura 30. Prototipo de la vista principal para SmartPhone de MegaGameShop.

Vista de una categoría

Figura 31. Prototipo de la vista de una categoría de MegaGameShop.

En esta vista podemos ver una de las categorías de nuestra tienda, en la que vemos los productos que se encuentran en ella y algunos filtros para aplicar.

En la siguiente figura se muestra cómo quedaría la vista para Smartphone:

Figura 32. Prototipo de la vista de categorías para SmartPhone de MegaGameShop.

Vista de un producto

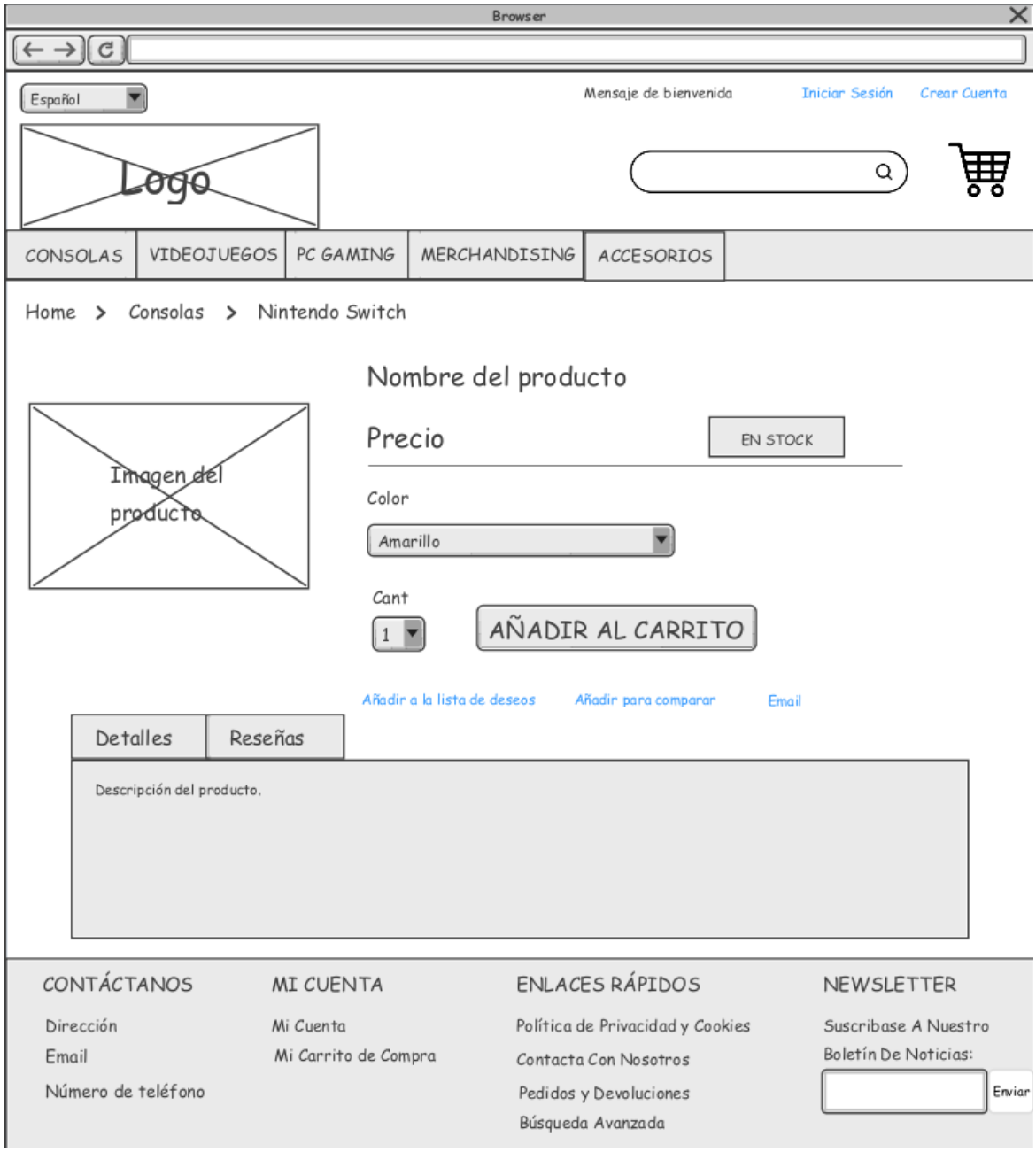

Figura 33. Prototipo de la vista de un producto de MegaGameShop.

En esta vista podemos ver un producto, el cual podremos elegir algunas configuraciones como el color. También podemos ver su disponibilidad, seleccionar una cantidad, leer una pequeña descripción y ver reseñas del mismo.

En la siguiente figura se muestra cómo quedaría la vista para Smartphone:

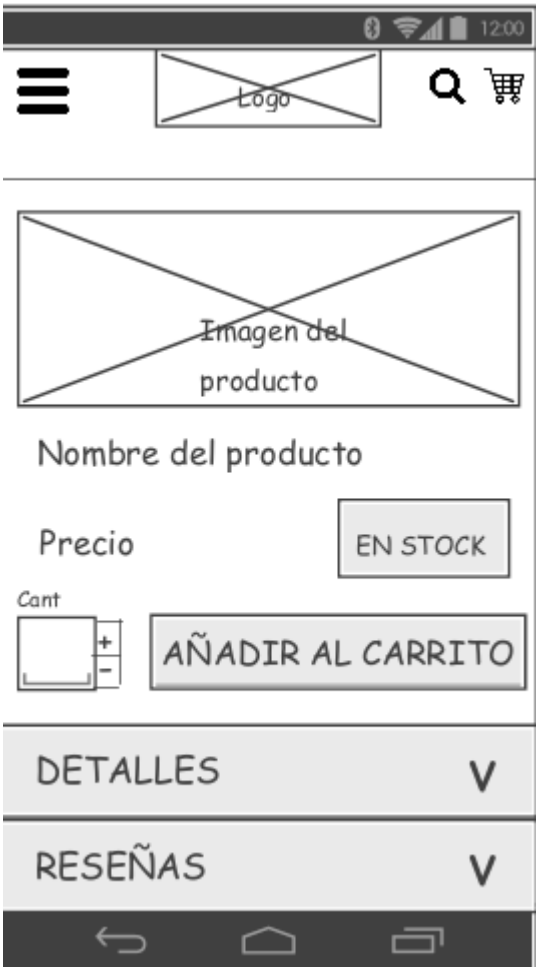

Figura 34. Prototipo de la vista un producto para SmartPhone de MegaGameShop.

Vista del carrito de compra

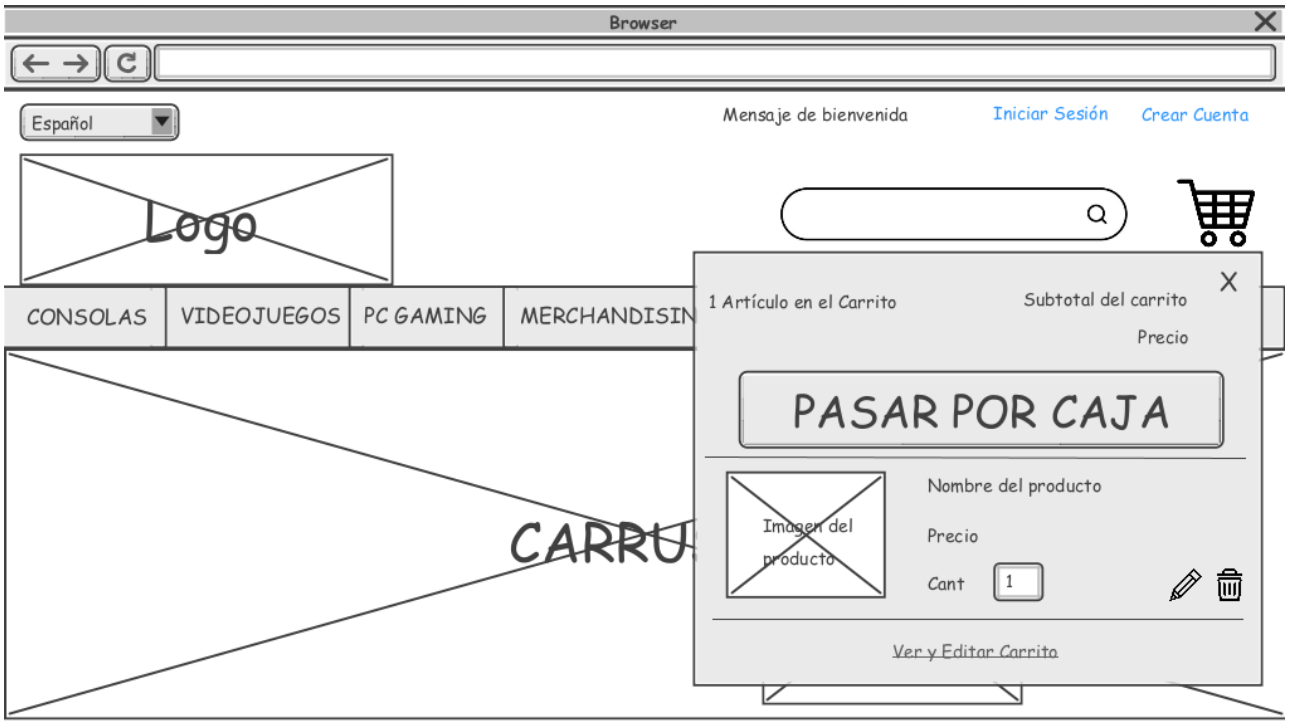

Figura 35. Prototipo de la vista del carrito de compra de MegaGameShop.

En esta vista vemos desplegado el carrito de compra, donde se nos muestra el precio del producto y nos permite eliminar o modificar la cantidad del mismo.

En la siguiente figura se muestra cómo quedaría la vista para Smartphone:

Figura 36. Prototipo de la vista del carrito del carrito de compra para SmartPhone de MegaGameShop.

Vista de pedido

Figura 37. Prototipo de la vista de pedido de MegaGameShop.

En esta vista, vemos los productos que deseamos comprar junto a los gastos de envío, que en este caso son gratuitos, y el IVA que se le aplica al pedido.

En la siguiente figura se muestra cómo quedaría la vista para Smartphone:

Figura 38. Prototipo de la vista del pedido para SmartPhone de MegaGameShop.

Vista de inicio de sesión o crear cuenta

Figura 39. Prototipo de la vista de iniciar sesión o crear cuenta de MegaGameShop.

En esta vista damos al usuario la elección de iniciar sesión con usuario y contraseña o crear una cuenta de MegaGameShop si no dispone de una para ingresar.

En la siguiente figura se muestra cómo quedaría la vista para Smartphone:

Figura 40. Prototipo de la vista de iniciar sesión o crear cuenta para SmartPhone de MegaGameShop.

Vista de crear cuenta

Figura 41. Prototipo de la vista de crear cuenta de MegaGameShop.

En esta vista vemos el formulario de creación de una cuenta de cliente. En él vemos los distintos datos que debe rellenar y también la opción de inscribirse al Boletín de Noticias.

En la siguiente figura se muestra cómo quedaría la vista para Smartphone:

Figura 42. Prototipo de la vista de crear cuenta para SmartPhone de MegaGameShop.

Vista de panel de control de cliente

Figura 43. Prototipo de la vista del panel de control de cliente de MegaGameShop.

En esta vista podemos ver la vista del panel de control de un cliente, en la cual puede configurar su dirección de envío y facturación, ver sus últimos pedidos y descargar sus facturas, su suscripción al Boletín de Noticias.

En la siguiente figura se muestra cómo quedaría la vista para Smartphone:

Figura 44. Prototipo de la vista del panel de control de cliente para SmartPhone de MegaGameShop.

4.6. Implementación.

En este apartado, veremos cómo hemos llevado a cabo la instalación del servidor web en nuestro ordenador, la instalación de Magento 2 y el desarrollo de la web que hemos creado.

4.6.1. Instalación del servidor web y de Magento 2.

Instalación del servidor web

Para instalar el servidor web en nuestro ordenador, hemos utilizado XAMPP, desde el que posteriormente llevaremos a cabo la instalación de Magento.

Primero debemos dirigirnos a la web oficial de XAMPP, y descargar el instalador a través del enlace https://www.apachefriends.org/download.html. Descargamos en concreto una versión que contenga PHP 7, como mostramos en la siguiente imagen.

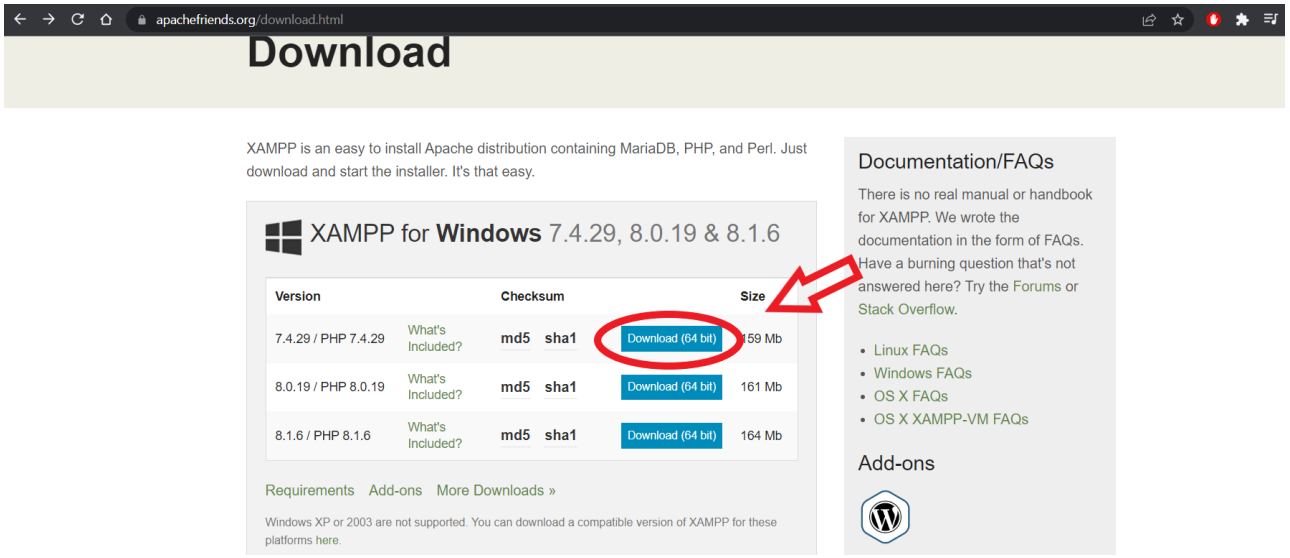

Figura 45. Descarga de XAMPP.

Ejecutamos el instalador del programa y seleccionamos los siguientes componentes que vamos a instalar:

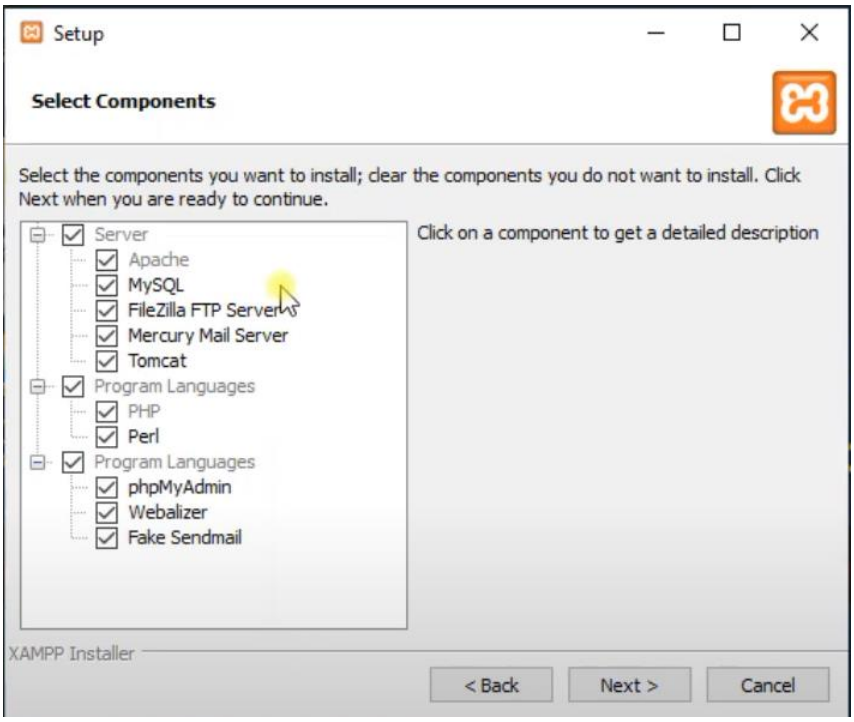

Figura 46. Componentes de XAMPP.

Tras seleccionar estos componentes, ya tendremos el programa instalado con éxito. Al ejecutar el programa, nos aparecerá el siguiente panel, donde debemos iniciar los módulos Apache y MySQL.

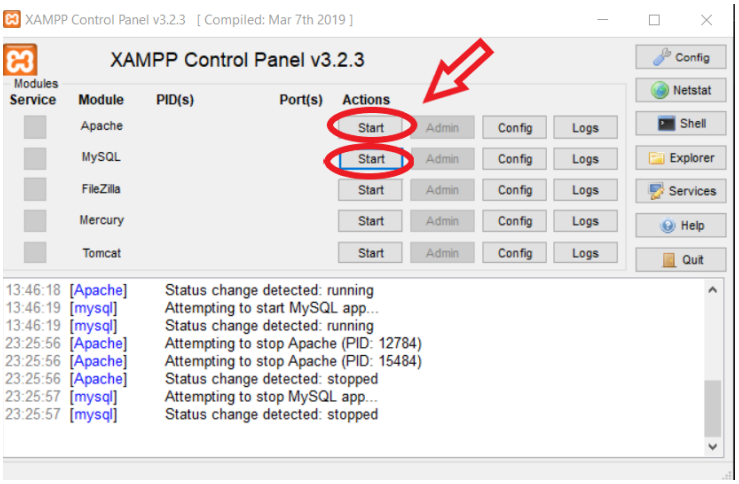

Figura 47. Módulos a iniciar en XAMPP.

Por último, entraremos en la configuración del panel XAMPP y estableceremos como principal el puerto 80.

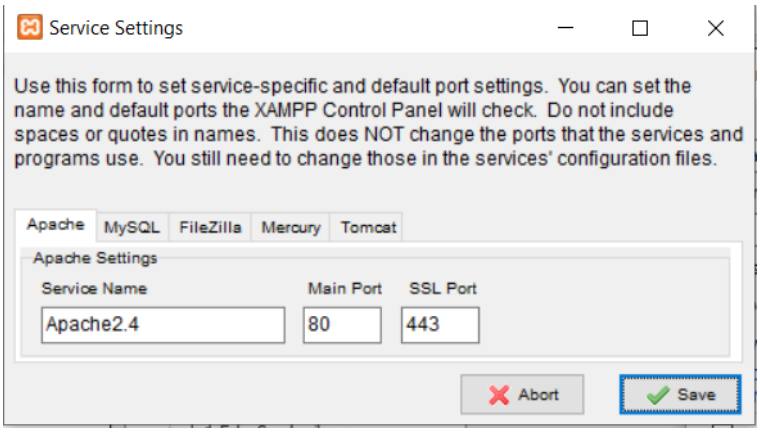

Figura 48. Configuración del puerto principal en XAMPP.

Tras estos pasos, ya tendremos configurado el programa. Ahora pasaremos a instalar Magento 2.

Instalación de Magento 2

Para llevar a cabo la instalación de Magento en nuestro servidor, primero debemos acceder al siguiente enlace https://magento.com/tech-resources/download y descargar, en nuestro caso, la versión 2.2.1.

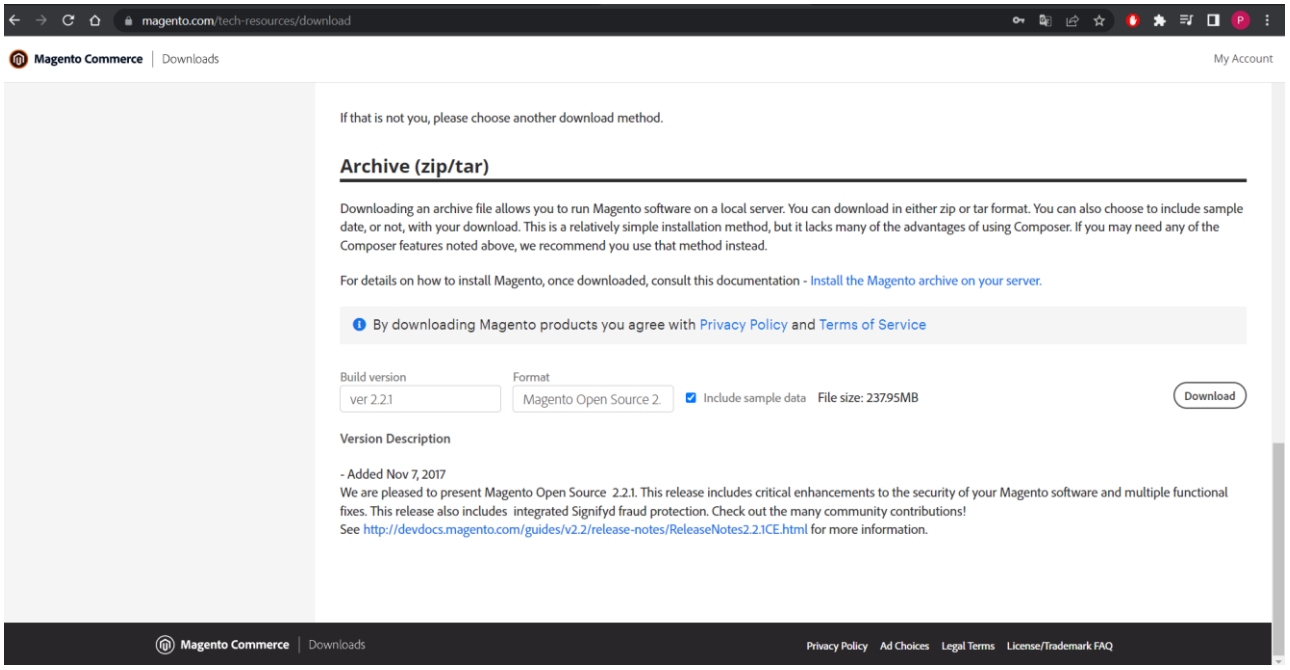

Figura 49. Descarga de Magento.

Al finalizar la descarga, descomprimimos la carpeta a la que llamaremos “Magento” y la copiaremos en la siguiente ruta de nuestro ordenador “C:\xampp\htdocs”.

Ahora, iremos a nuestro navegador de internet e introduciremos la siguiente URL: “localhost:8080/Magento” para comenzar con la instalación de Magento. En la URL vemos que 8080 se utiliza porque es el puerto principal que hemos establecido previamente en la configuración de XAMPP y es “/Magento” porque es el nombre que le hemos dado a la carpeta que hemos descomprimido. Si todo es correcto, comenzaremos con la instalación de Magento.

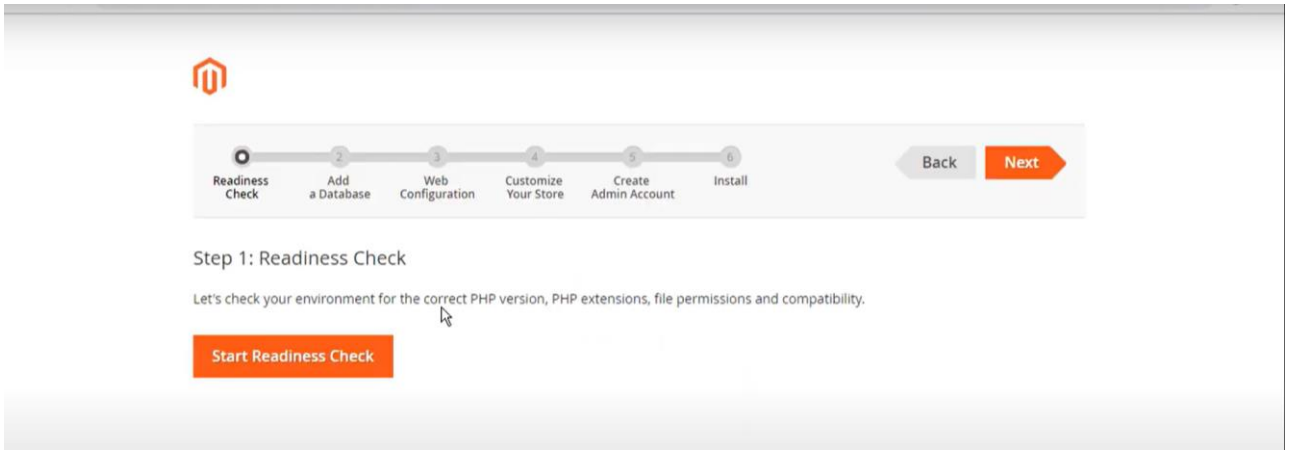

Figura 50. Comienzo de la instalación de Magento 2.

Ahora el instalador comprobará nuestras extensiones de PHP y nos alertará de que nos faltan 3 extensiones.

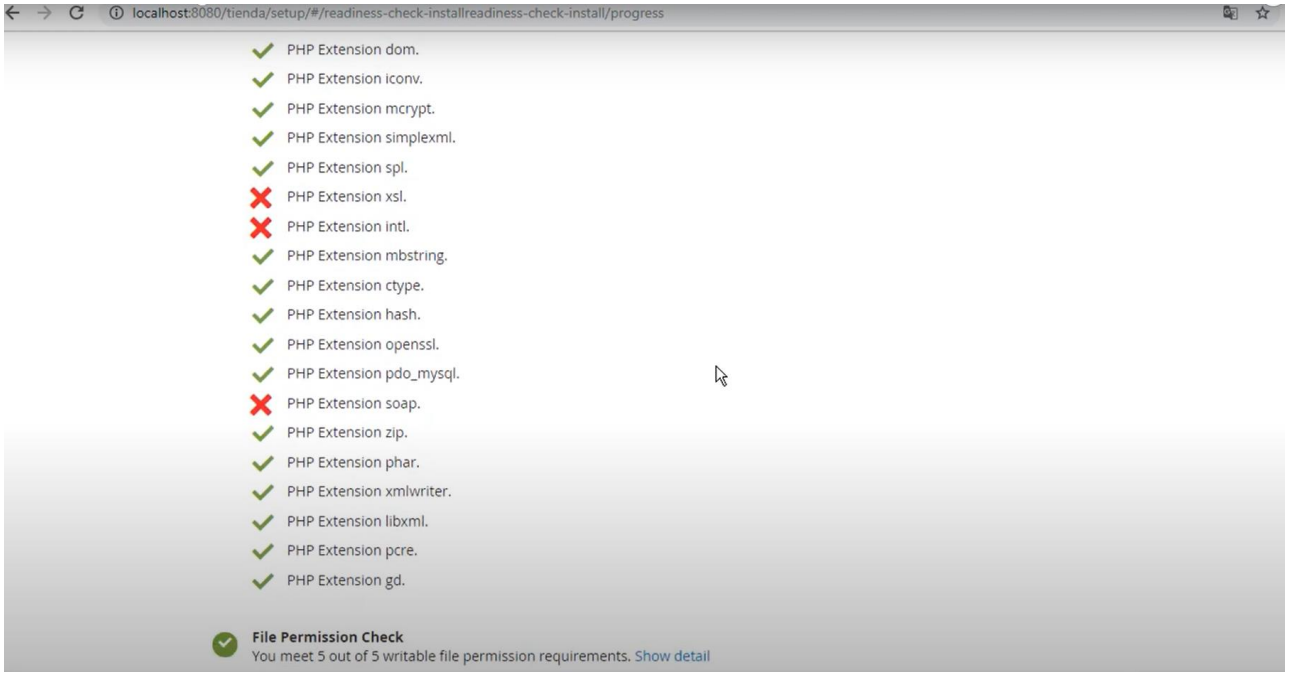

Figura 51. Comprobación de las extensiones PHP.

Para agregarlas y continuar con la instalación debemos volver a XAMPP y acceder al archivo php.ini de la siguiente manera.

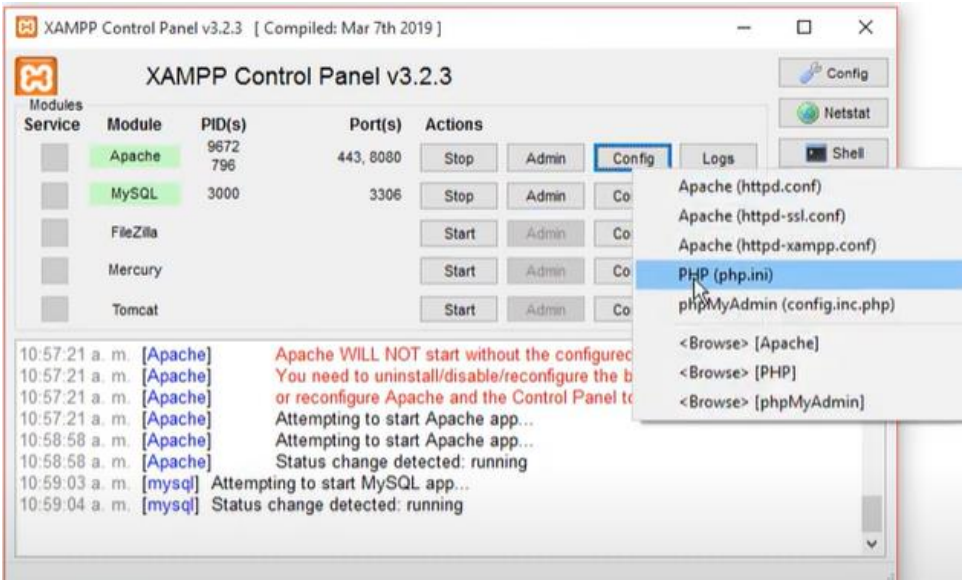

Figura 52. Acceder al fichero php.ini en XAMPP.

En este fichero, debemos buscar las 3 líneas “;extensión=php_xls.dll”, “;extensión=php_intl.dll” y “;extensión=php_soap.dll” y en las 3 líneas quitar el carácter ‘;’. Tras realizar esta modificación podemos avanzar al siguiente paso.

Para el siguiente paso, primero debemos acceder a la base de datos en phpMyAdmin a través del enlace http://localhost/phpmyadmin/ y pulsaremos en “Nueva” y añadiremos en nuestro caso la base de datos “tienda”. Pulsamos en “Crear” y tendremos nuestra base de datos creada.

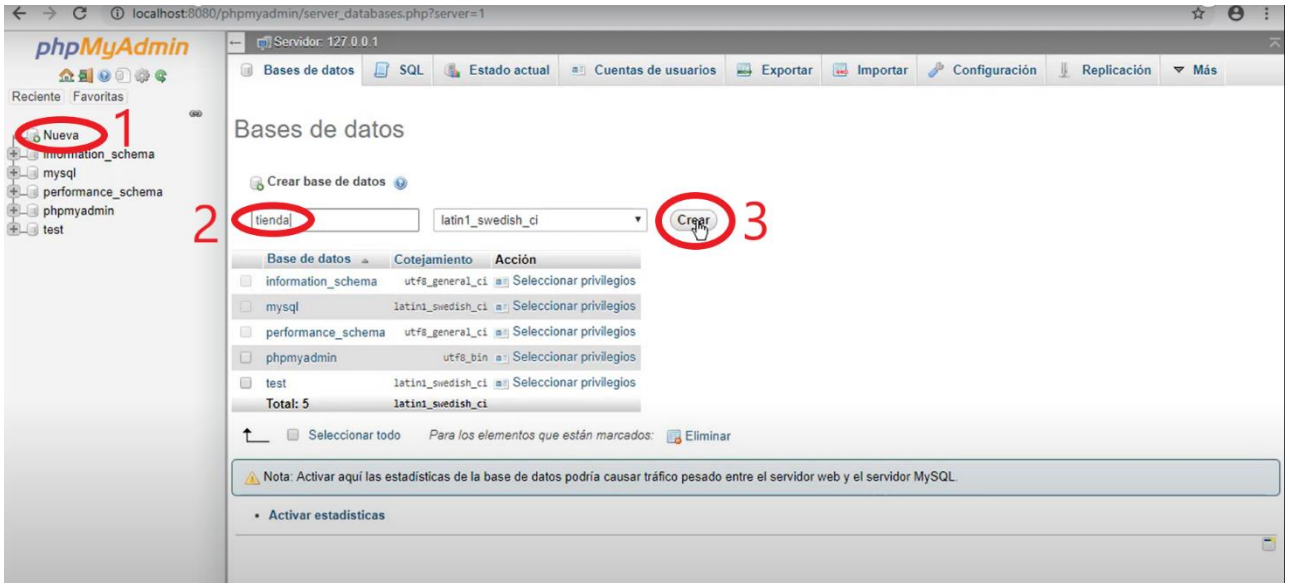

Figura 53. Creación de base de datos en phpMyAdmin.

Tras crear la base de datos, volvemos a la web de instalación de Magento y rellenamos el formulario con los siguientes datos de la base de datos.

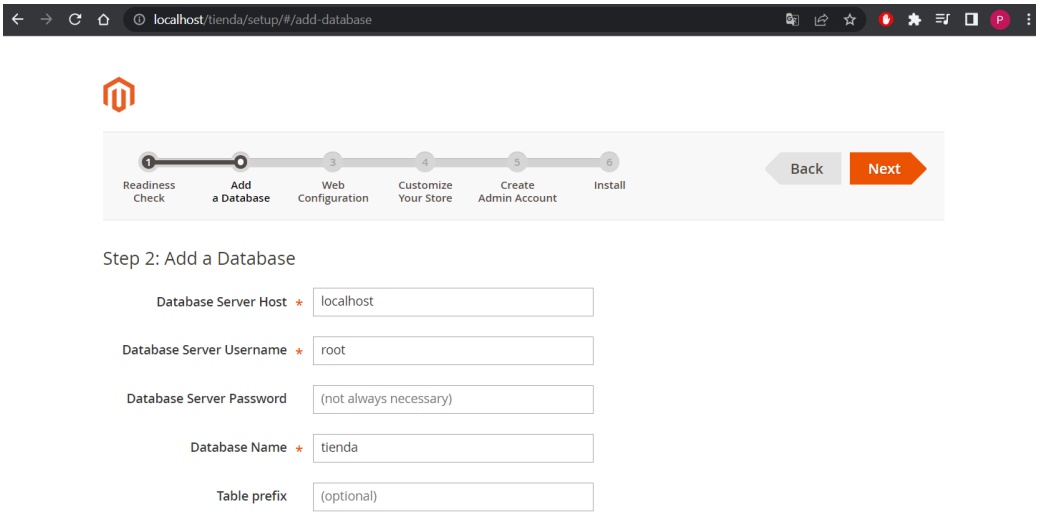

Figura 54. Configuración de la base de datos en Magento.

A continuación, procederemos a rellenar el formulario de la configuración de nuestra web, en la que tenemos que introducir los siguientes parámetros.

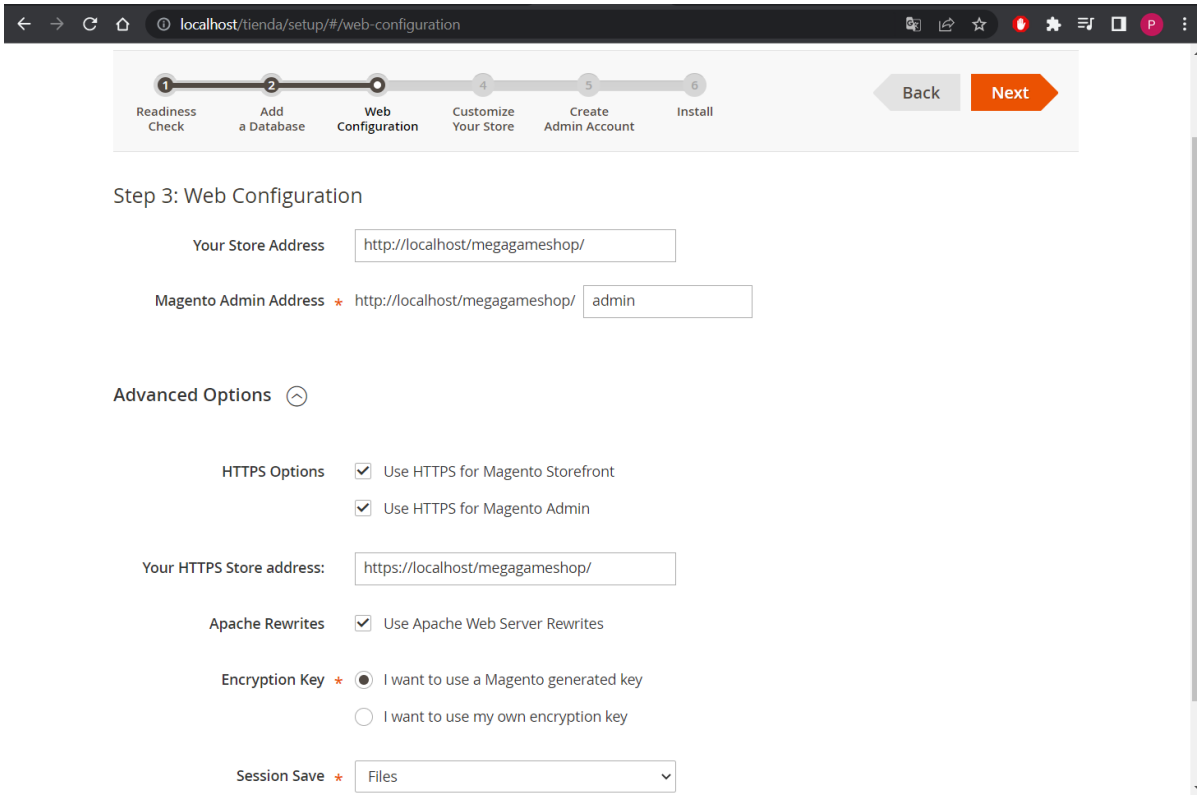

Figura 55. Configuración web de Magento.

Tras la configuración web de nuestra tienda, debemos configurar la moneda, región e idioma por defecto de la misma.

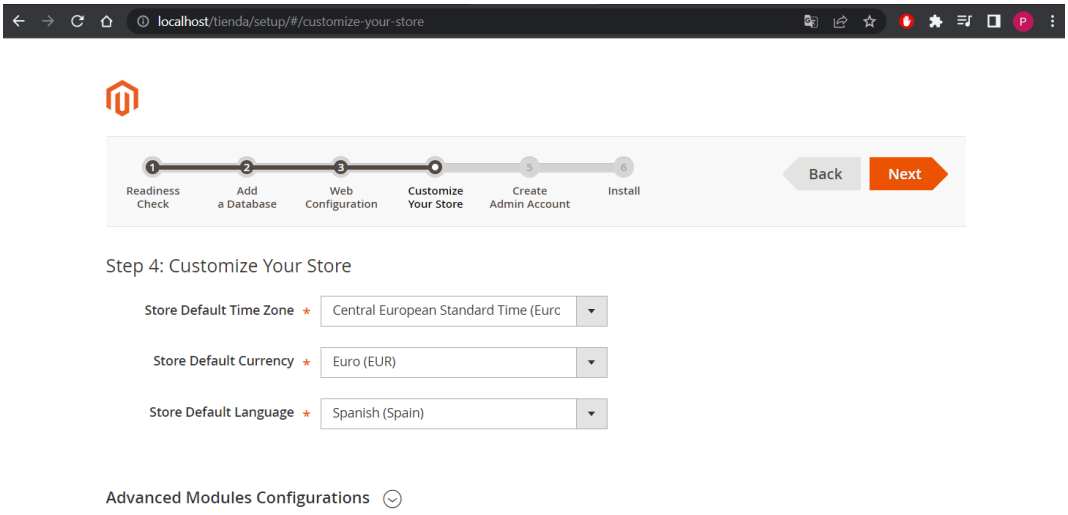

Figura 56. Configuración moneda, región e idioma de MegaGameShop.

Para finalizar, debemos introducir el usuario y contraseña que vamos a utilizar para entrar a nuestro panel de administración de Magento. Introduciremos los datos que se muestran en la siguiente figura.

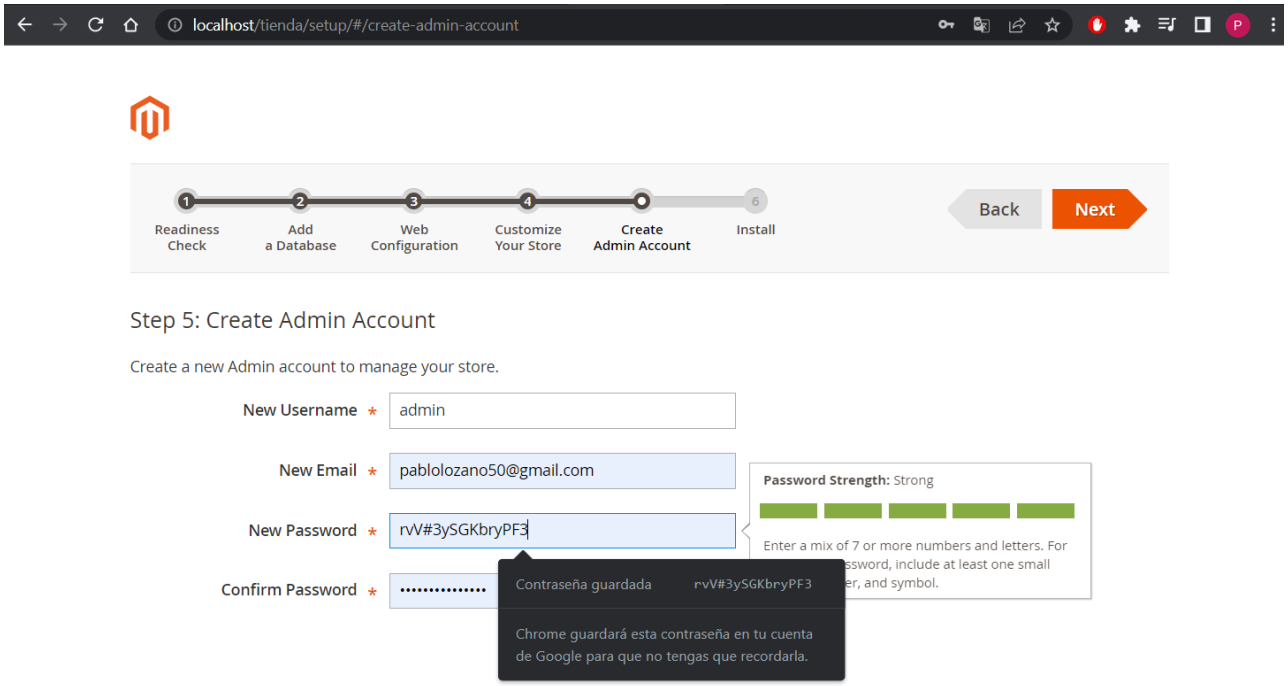

Figura 57. Configuración de datos del panel de administración de Magento.

Tras finalizar la barra de progreso, nos aparecerá un resumen con los datos para comenzar a utilizar el panel de administración de Magento.

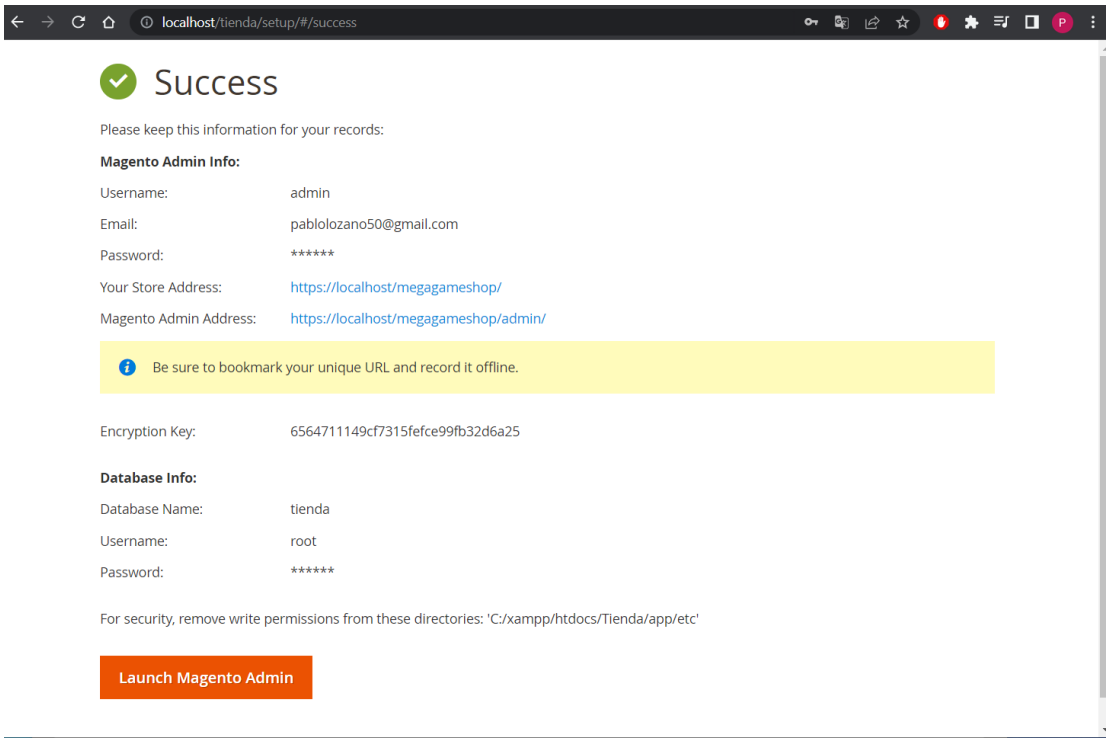

Figura 58. Finalización de instalación de Magento.

4.6.2. Desarrollo web de MegaGameShop.

Una vez instalado el servidor, podremos comenzar con el desarrollo de nuestra web. Para ello, buscaremos una plantilla compatible con nuestra versión de Magento.

4.6.2.1. Instalación de la plantilla.

Accederemos a la web https://www.hiddentechies.com, y descargamos el tema "Next – Free Magento 2 Theme", que como vemos en su descripción está diseñado especialmente para tiendas de electrónica. Lo seleccionamos y descargamos el archivo comprimido con el tema.

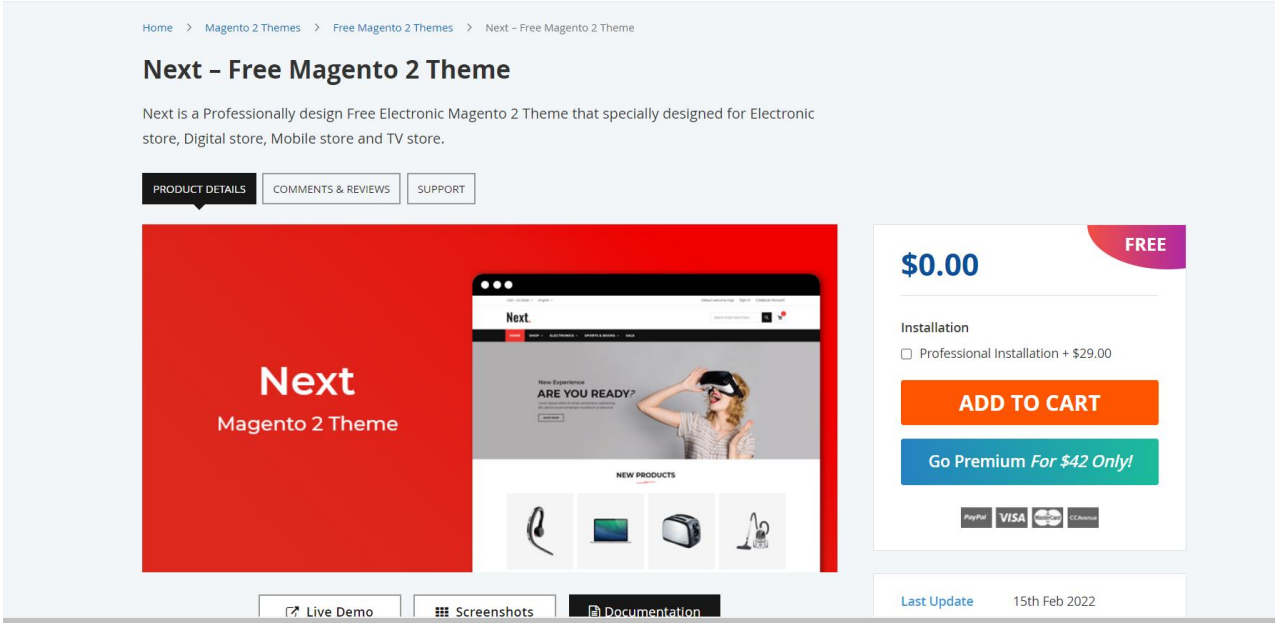

Figura 59. Descarga del tema Next - Free Magento 2 Theme.

Tras descomprimir el archivo, encontramos dos directorios \app y \pub, que copiaremos a la raíz de nuestro proyecto "C:\xampp\htdocs\megagameshop"., sobrescribiendo los datos existentes.

A continuación, debemos dirigirnos a la consola del servidor, que se encuentra en el panel de control y ejecutar los siguientes comandos sobre la raíz del proyecto:

```
php bin/magento setup:upgrade

php bin/magento setup:static-content:deploy -f
```

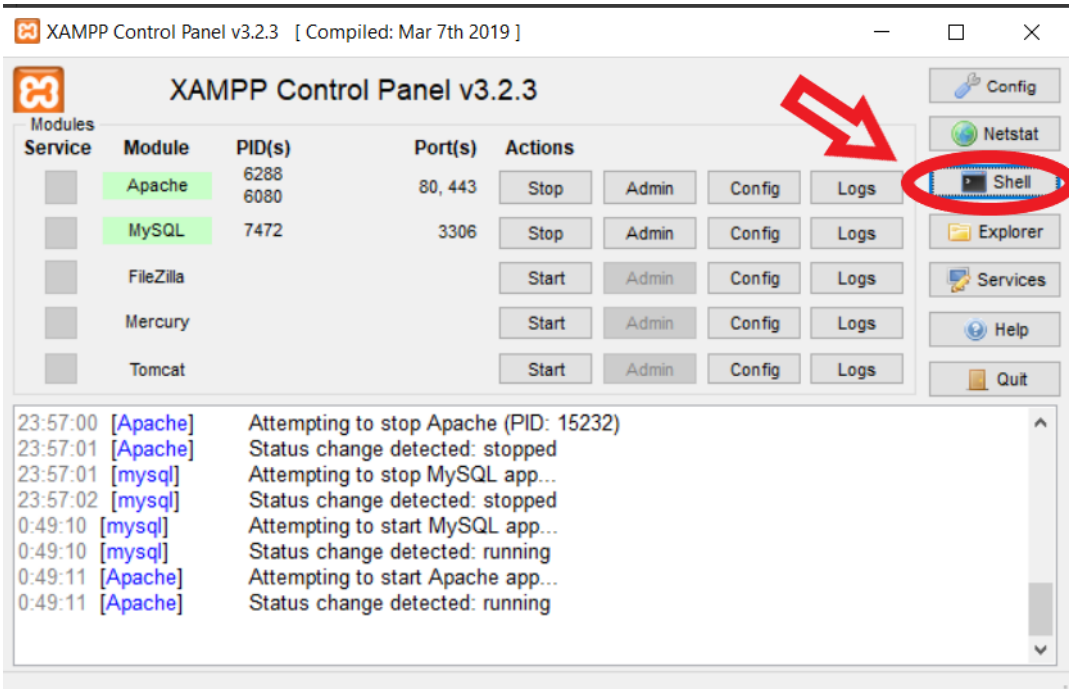

Figura 60. Consola de XAMPP.

Tras ejecutar los comandos, nos dirigiremos al panel de administración de Magento y en la sección **Content < Configuration**, pulsaremos sobre **Edit.** Después, seleccionaremos el tema “Hiddentechies Next” y guardaremos la configuración.

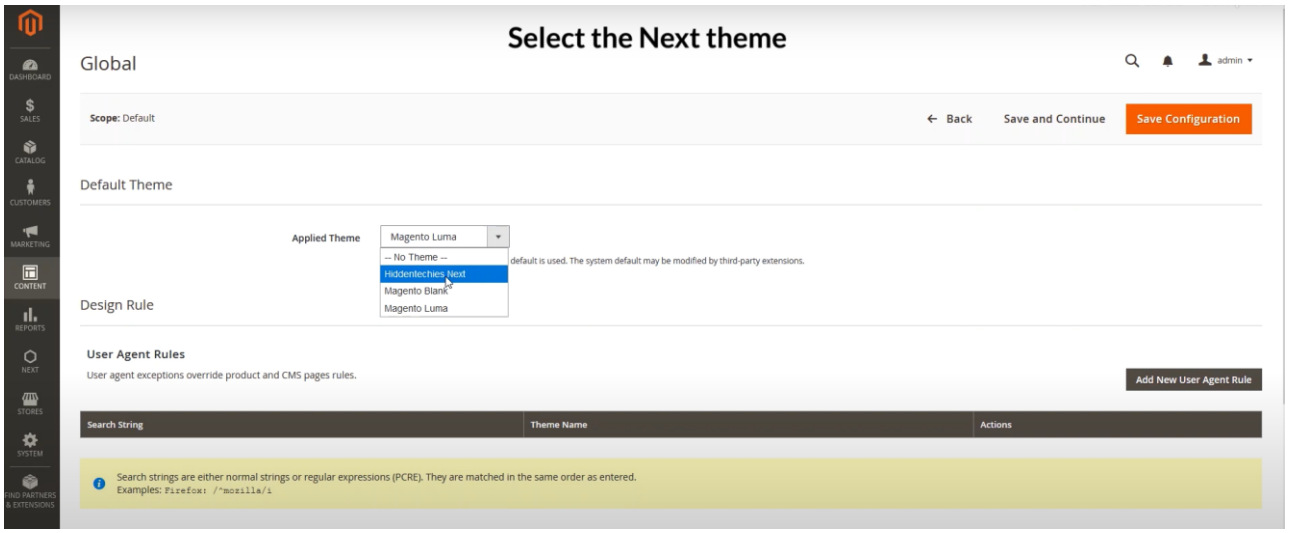

Figura 61. Selección del tema Next en Magento.

Por último, volveremos a la consola de XAMPP y para ver los cambios reflejados en el frontend, siempre debemos ejecutar en orden los siguientes comandos al hacer cambios en el backend:

-Reindexar: php htdocs/megagameshop/bin/magento indexer:reindex
-Limpiar la caché: php htdocs/megagameshop/bin/magento cache:clean
-Realizar un flush: php htdocs/megagameshop/bin/magento cache:flush

Tras ejecutar estos comandos, visualizamos el tema “Next” en el frontend.

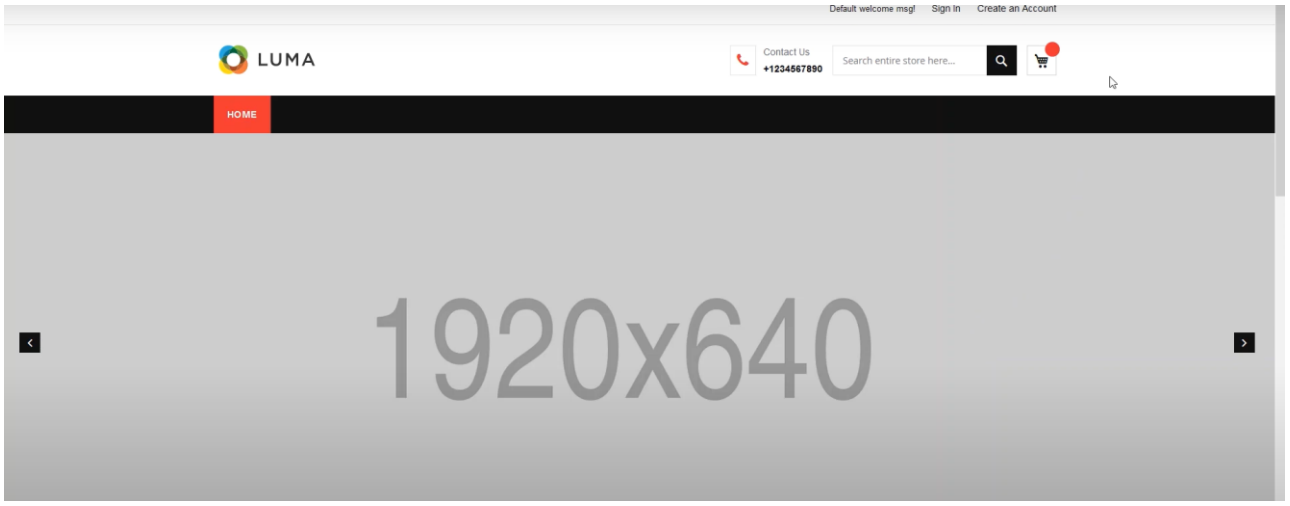

Figura 62. Visualización del tema Next en el frontend.

4.6.2.2. Accesibilidad en varios idiomas.

Uno de los requisitos que debía cumplir nuestro sitio web era que debía tener accesibilidad en varios idiomas. En este caso, podremos visualizar el sitio web en dos idiomas, inglés y español. Por defecto, el idioma será el español.

Para ello debemos dirigirnos al backend y dirigirnos a la sección **Store < All Stores < Create Store View**. Crearemos una vista para inglés y otra para español.

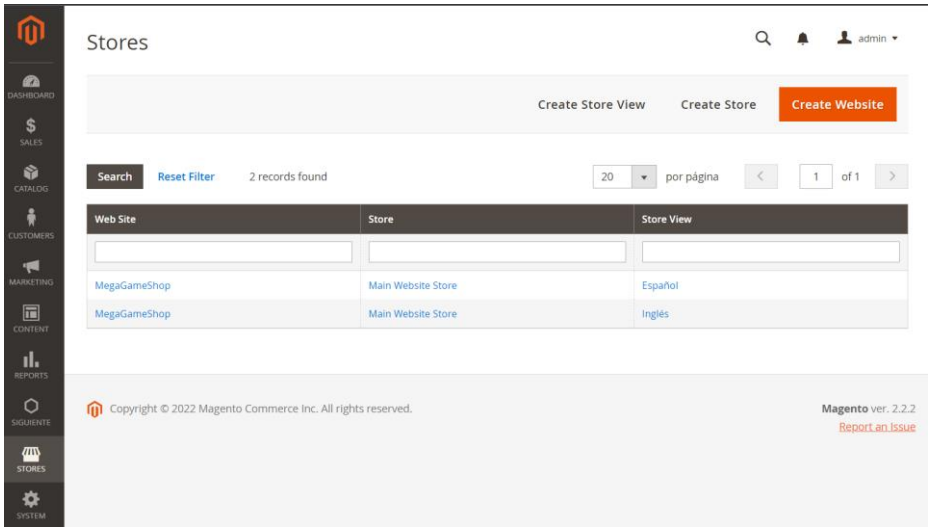

Figura 63. Creación de vistas en distintos idiomas.

Al guardar los cambios y reindexar la web, nos aparecerá en el frontend un desplegable en la cabecera donde podremos alternar entre los dos idiomas.

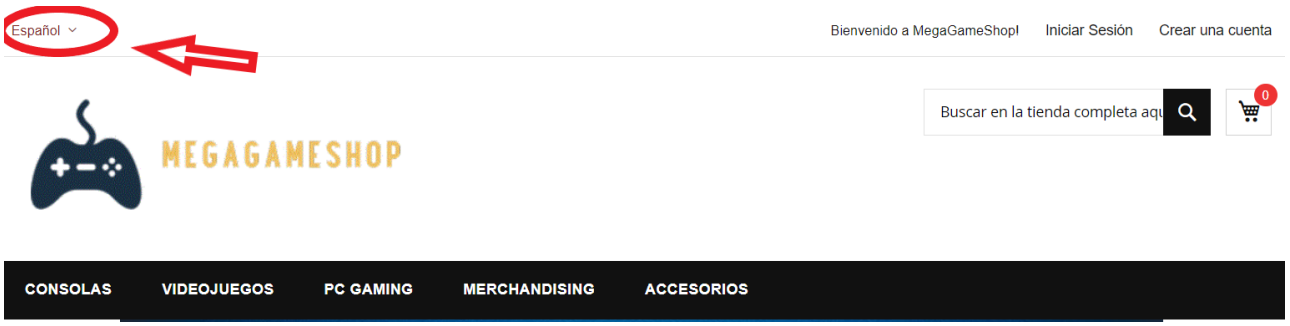

Figura 64. Desplegable con los idiomas de MegaGameShop.

4.6.2.3. Personalización de la cabecera y pie de página.

El siguiente paso es personalizar la cabecera y pie de página (header y footer), donde modificaremos el logo de la empresa, títulos, mensaje de bienvenida, icono web, datos de contacto, etc. Esto debemos hacerlo tanto para la vista en inglés como en español.

Primero accederemos a la sección **Content < Configuration** y pulsaremos en editar la vista en castellano. En la sección **HTML Head** cargaremos nuestro icono de empresa y el título por defecto de nuestra web. Esto aparecerá en la pestaña del frontend de nuestro navegador. En la siguiente figura se muestra con más detalle.

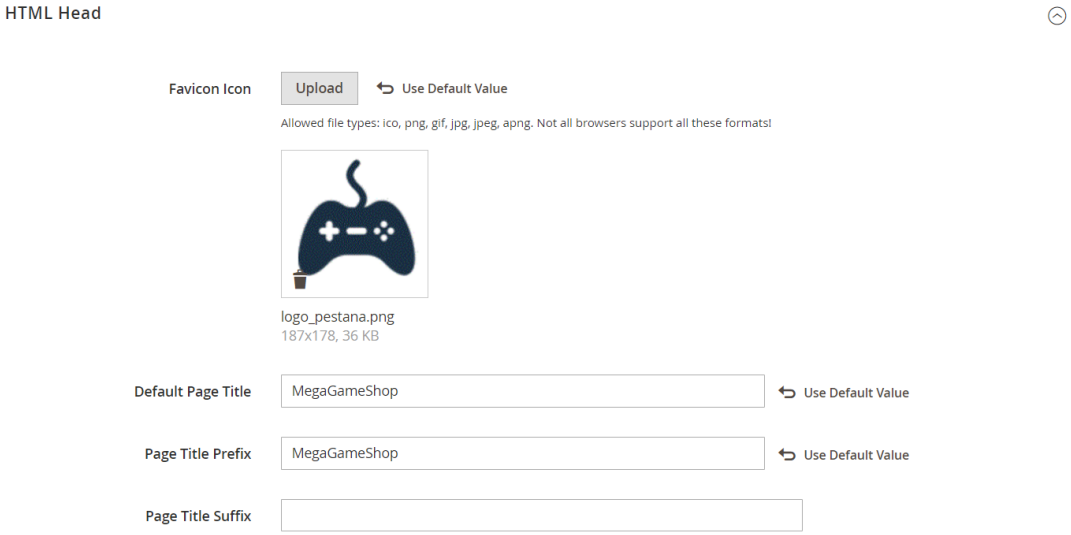

Figura 65. Configuración del icono y título de la pestaña de MegaGameShop.

Esto nos mostrará el siguiente resultado en la pestaña del frontend después de reindexar los datos.

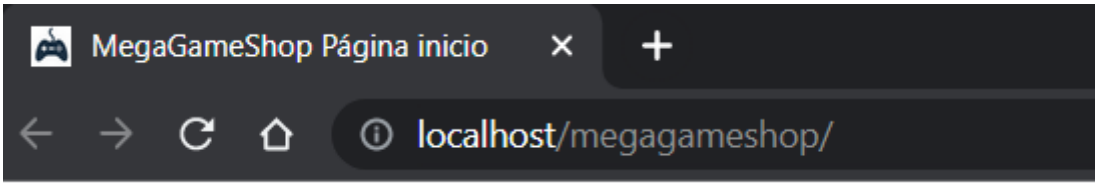

Figura 66. Icono y título de la pestaña de MegaGameShop.

Pasamos a la sección **Header**, donde cargaremos el logo de la empresa y sus dimensiones (largo y ancho). Además, personalizaremos el mensaje de bienvenida.

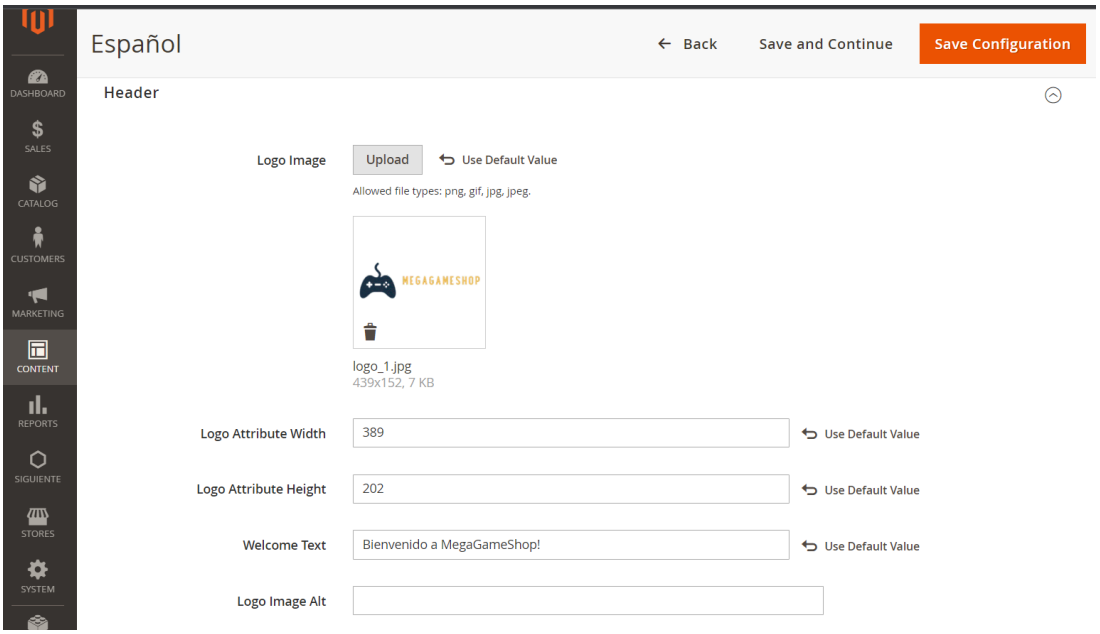

Figura 67. Configuración de la cabecera de MegaGameShop.

Esto nos mostrará el siguiente resultado en la cabecera del frontend después de reindexar los datos.

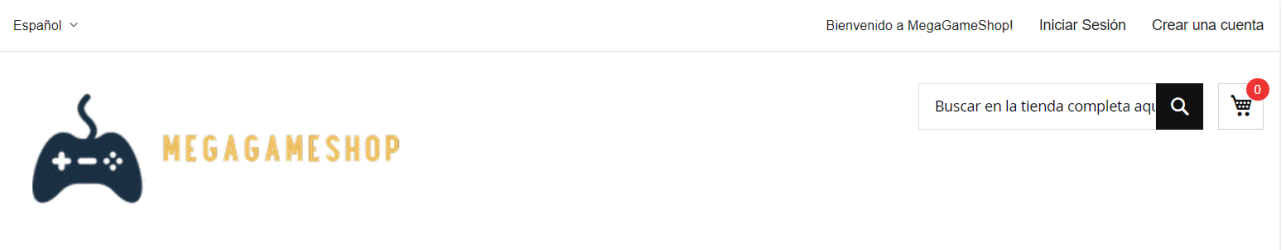

Figura 68. Cabecera de MegaGameShop.

Por último, modificaremos los datos de contacto de la empresa. Esto lo haremos desde la sección **Store < Configuration < Hiddentechies < Next Settings < Footer Contact Block** donde añadiremos nuestra dirección, email y número de teléfono de la empresa.

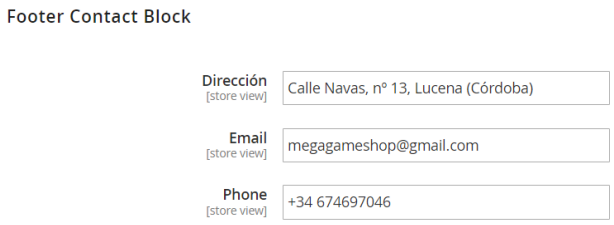

Figura 69. Datos de contacto del footer de MegaGameShop.

4.6.2.4. Administración de categorías y subcategorías.

El siguiente paso es agregar las categorías que va a tener nuestra tienda, así como las subcategorías que encontraremos dentro de ellas.

Para ello nos vamos a la sección **Catalog < Categories** y, una vez allí añadiremos las categorías y subcategorías de las mismas. Estas serían las categorías y subcategorías que vamos a desarrollar a lo largo de nuestro proyecto.

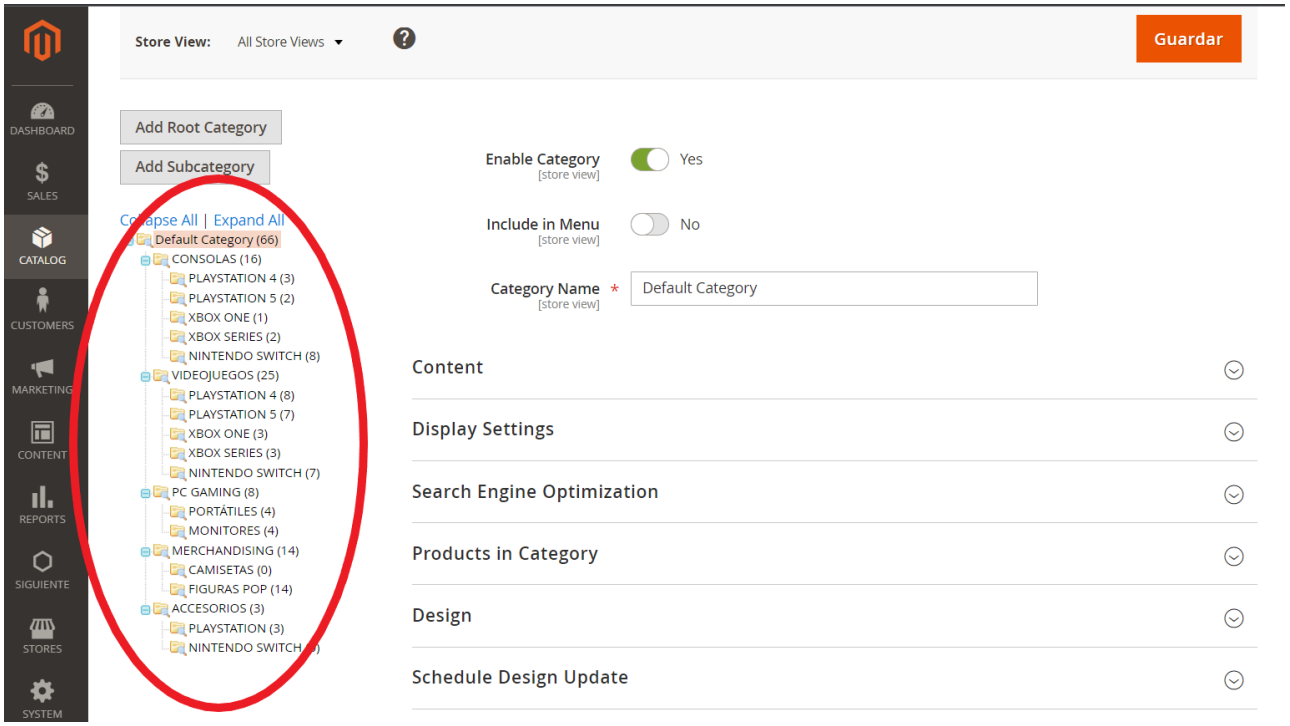

Figura 70. Configuración de categorías y subcategorías de MegaGameShop.

Como vemos, hemos dividido nuestros productos en cinco categorías:

-Consolas: en esta categoría hemos dividido las consolas dependiendo de la plataforma a la que pertenezcan (Playstation 4, Playstation 5, Xbox One, Xbox Series y Nintendo Switch).

-Videojuegos: esta categoría la hemos dividido en las mismas categorías que la categoría "Consolas" dependiendo de la consola a la que pertenezca el videojuego.

-PC Gaming: esta categoría venderemos productos relacionados con el PC y el mundo de los videojuegos. La hemos dividido en dos, portátiles y monitores.

-Merchandising: en esta categoría venderemos artículos basados en videojuegos o películas relacionadas con el mundo de los videojuegos. Lo hemos dividido en camisetas y figuras POP.

-Accesorios: en esta categoría venderemos accesorios (cascos, bases de carga para mandos, fundas). La hemos dividido en dos secciones dependiendo de la plataforma para la que sea el accesorio (Playstation y Nintendo Switch).

Después de añadir estas categorías y subcategorías, reindexaremos nuestra web y nos quedará el resultado de la siguiente figura. Vemos desplegadas las subcategorías de una categoría como ejemplo:

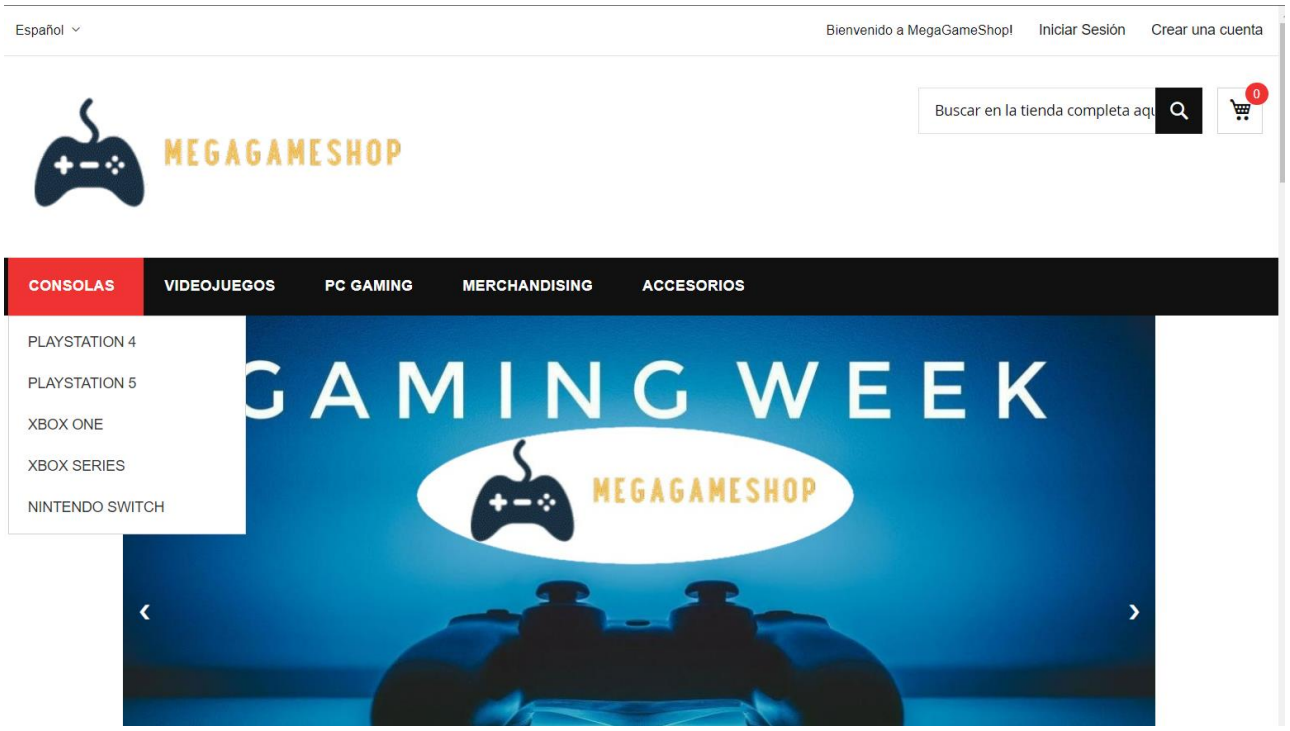

Figura 71. Categorías y subcategorías en el frontend versión web de MegaGameShop.

En la siguiente figura se muestra cómo quedarían las categorías y subcategorías para la vista de Smartphone:

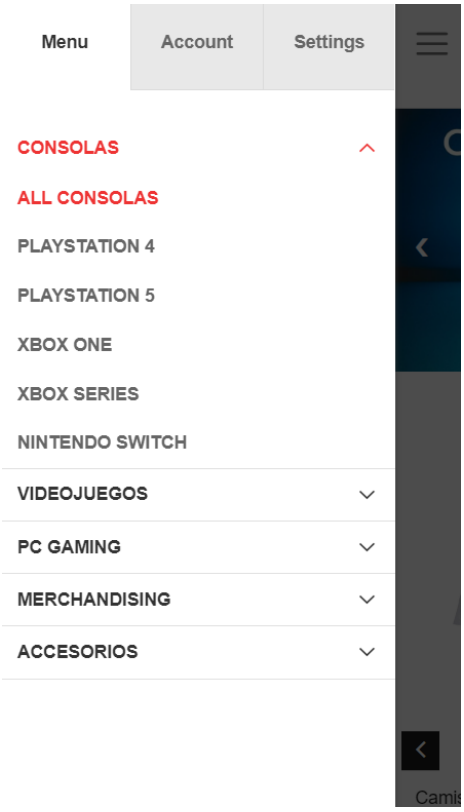

Figura 72. Categorías y subcategorías en el frontend versión Smartphone de MegaGameShop.

4.6.2.5. Configuración de impuestos.

Antes de empezar a añadir productos a nuestra web, necesitaremos configurar los impuestos de España, que es donde nuestra web va a tener actividad. Por lo tanto, en España nuestros productos tendrán un IVA del 21%, ya que nuestra web vende productos electrónicos, figuras y ropa.

Primero, debemos dirigirnos a la sección **Stores < Tax Zones and Rates < Add New Tax Rate** y rellenarlo con los siguientes datos añadiendo un 21% de IVA.

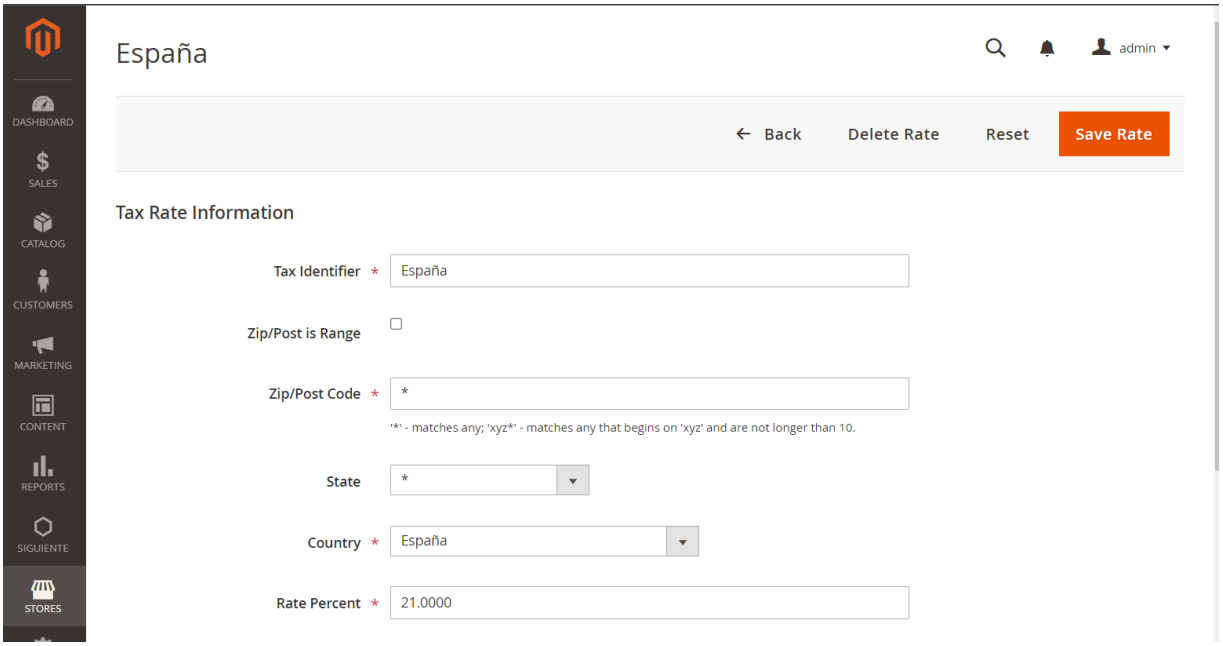

Figura 73. Configuración de Tasas en el backend.

Por último, iremos a la sección **Stores < Tax Rules < Add New Tax Rule** y rellenarlo con los siguientes datos:

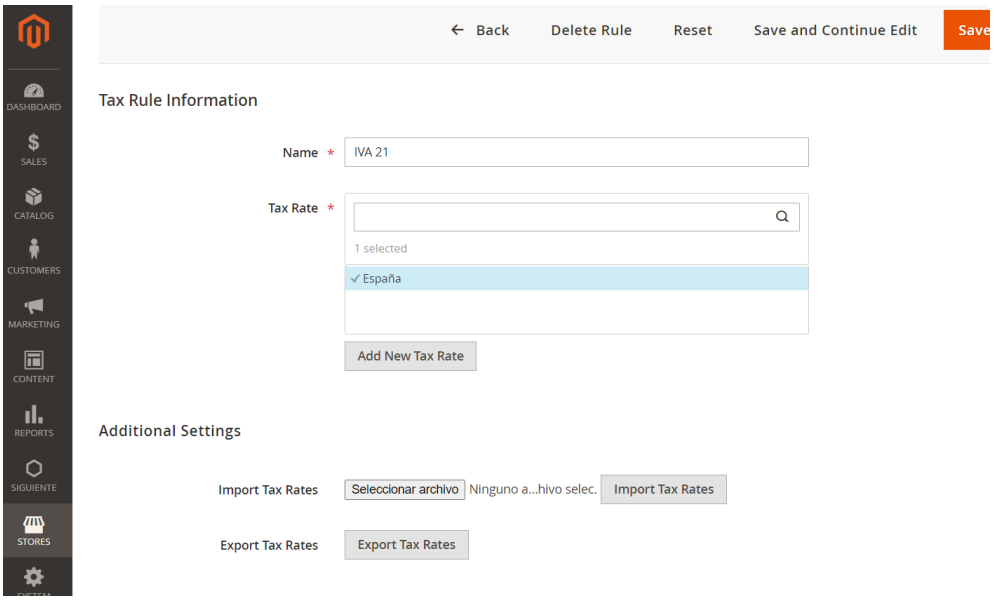

Figura 74. Configuración de regla de impuestos en el backend.

Tras esta configuración, ya tendremos creados correctamente los impuestos pertenecientes a España. Posteriormente, debemos aplicar este IVA del 21% a los productos en el frontend para que el cliente tenga claro en todo momento el precio final que va a pagar por el producto.

4.6.2.5. Administración de productos.

Una vez vistas las categorías y subcategorías de nuestro sitio web y creadas las tasas impositivas de nuestro país, debemos añadir todos los productos clasificándolos en estas categorías y subcategorías.

Para ello, debemos dirigirnos a la sección **Catalog < Product < Add Product**, donde tendremos que rellenar los siguientes campos, como se ve en el siguiente ejemplo repartidas en 3 figuras, donde añadimos una consola Playstation 5 con lector. Explicaremos los apartados más relevantes después de ver las figuras.

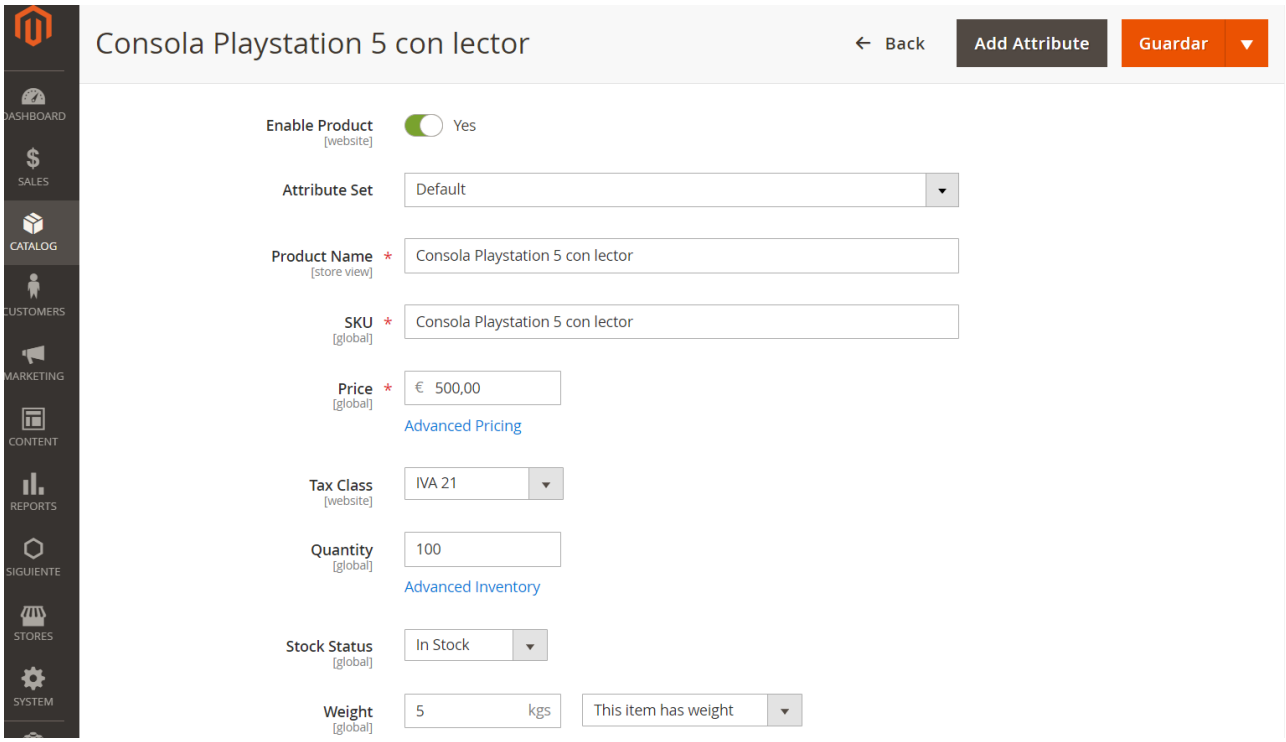

Figura 75. Creación de un producto en el backend (parte I).

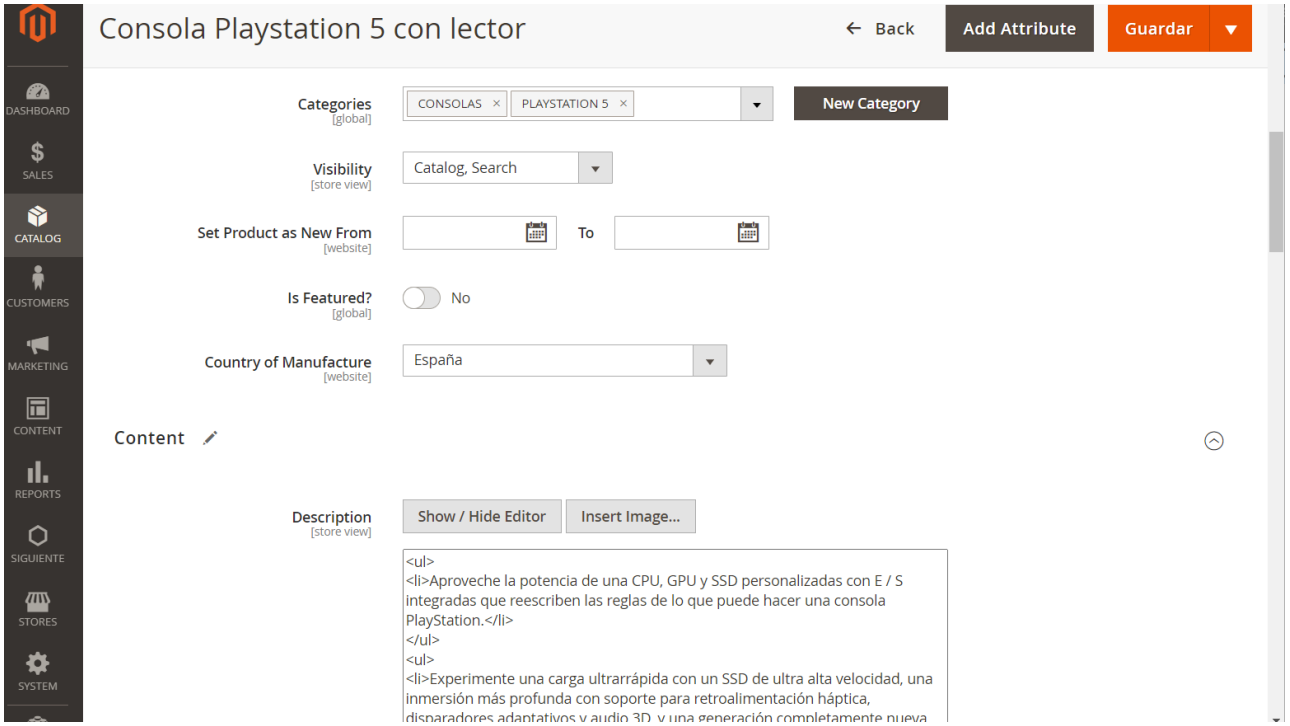

Figura 76. Creación de un producto en el backend (parte II).

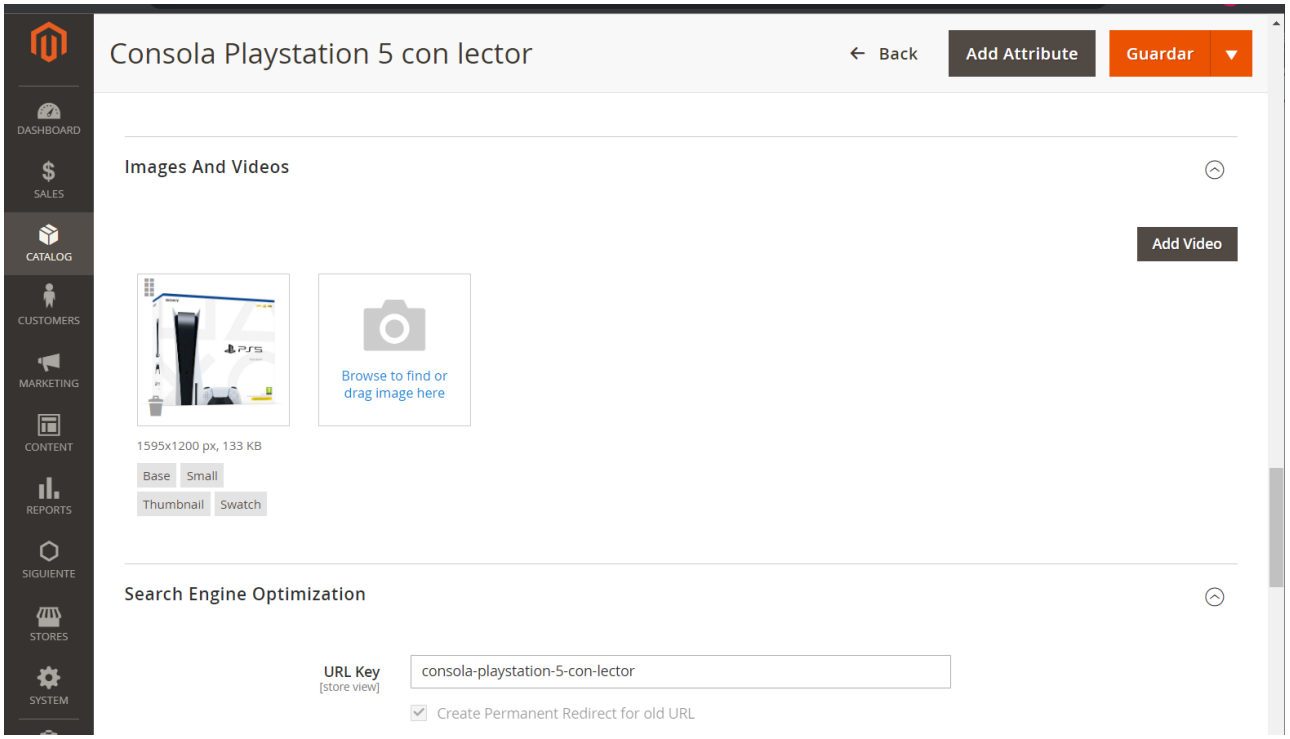

Figura 77. Creación de un producto en el backend (parte III).

-Product Name: es el nombre del producto, el que se mostrará al cliente en el frontend.

-SKU: es el nombre que utilizamos en el backend para identificar al producto como único.

-Price: es el precio del producto, exento de IVA.

-Tax Class: es el tipo de tasa a aplicar al producto. En este caso el 21% de IVA creado anteriormente.

-Quantity: es la cantidad de stock de la que disponemos en el almacén.

-Stock Status: es el estado del stock para este producto. En este caso, el producto se encuentra en stock.

-Weight: es el peso del producto. En este caso, 5 kilogramos.

-Categories: son las categorías y subcategorías a las que este producto pertenece.

-Visibility: es la visibilidad que tendrá nuestro producto. En este caso, nuestro producto podrá ser buscado tanto desde el catálogo como realizando una búsqueda con la barra de búsqueda del footer.

-Description: en él haremos una descripción del producto de manera detallada. En este campo se permite incluir algunos lenguajes de programación bastante conocidos como HTML, PHP y JavaScript.

-Images And Videos: en este campo podremos cargar imágenes o videos relacionados con el producto.

-URL Key: en este apartado se personaliza la URL que el cliente verá en el navegador al estar visualizando el producto.

Tras finalizar de subir el producto, mostramos en la siguiente figura la vista en el frontend para la versión web:

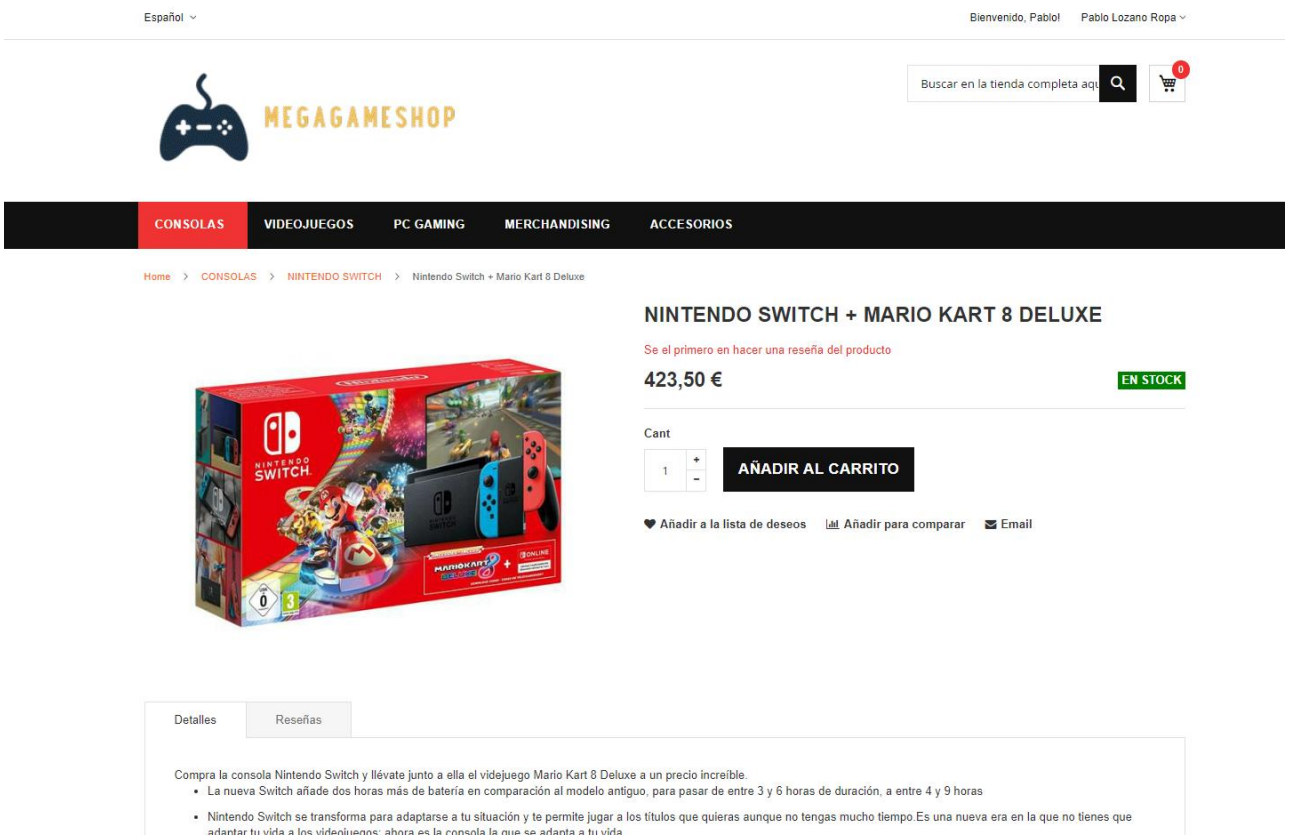

Figura 78. Producto en el frontend versión web de MegaGameShop.

En la siguiente figura se muestra cómo quedaría un producto para la vista para Smartphone:

Figura 79. Producto en el frontend versión Smartphone de MegaGameShop.

Al terminar de subir todos los productos, vamos a pasar a ver los distintos bloques con las vistas que tendremos en la tienda.

4.6.2.6. Administración de bloques.

Los bloques son especialmente útiles en cuanto a la personalización de nuestra tienda, ya que nos permite modificar el código fuente de cada bloque a nuestro gusto. Esto nos posibilita fragmentar el código en bloques y hacer más legible el código.

Para visualizar los bloques de la tienda, accederemos mediante la ruta **Content < Blocks** y nos aparecerán los bloques de los que está compuesta nuestra web.

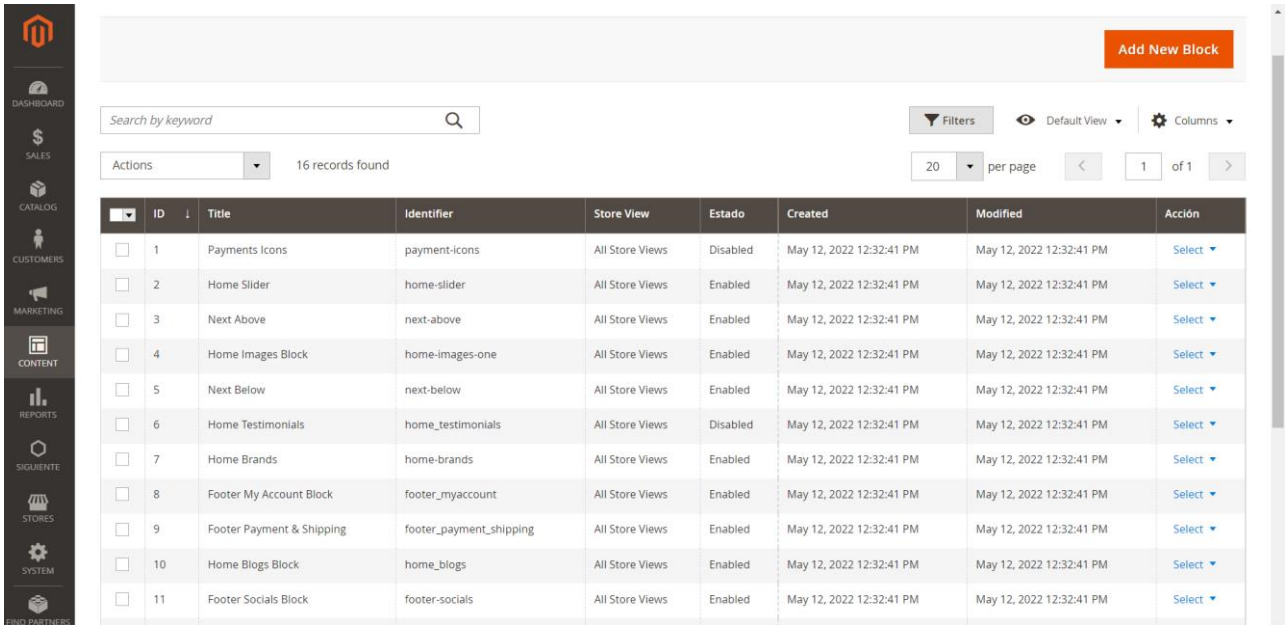

Figura 80. Administración de bloques en el backend.

Los bloques que más me he enfocado en realizar modificaciones son los referentes a la página principal, donde he modificado el carrusel de la parte superior, los banners de publicidad, los bloques de productos que se muestran y el carrusel de marcas que trabajan con MegaGameShop. Además, cada uno de los accesos rápidos del footer ha sido modificado internamente adaptándolo a nuestra empresa.

A continuación, mostraremos uno de ellos, el bloque Home Slider. En él podremos ver el apartado en el que podemos ir escribiendo el código fuente para hacer el carrusel que nos encontramos justo debajo del header de nuestra web.

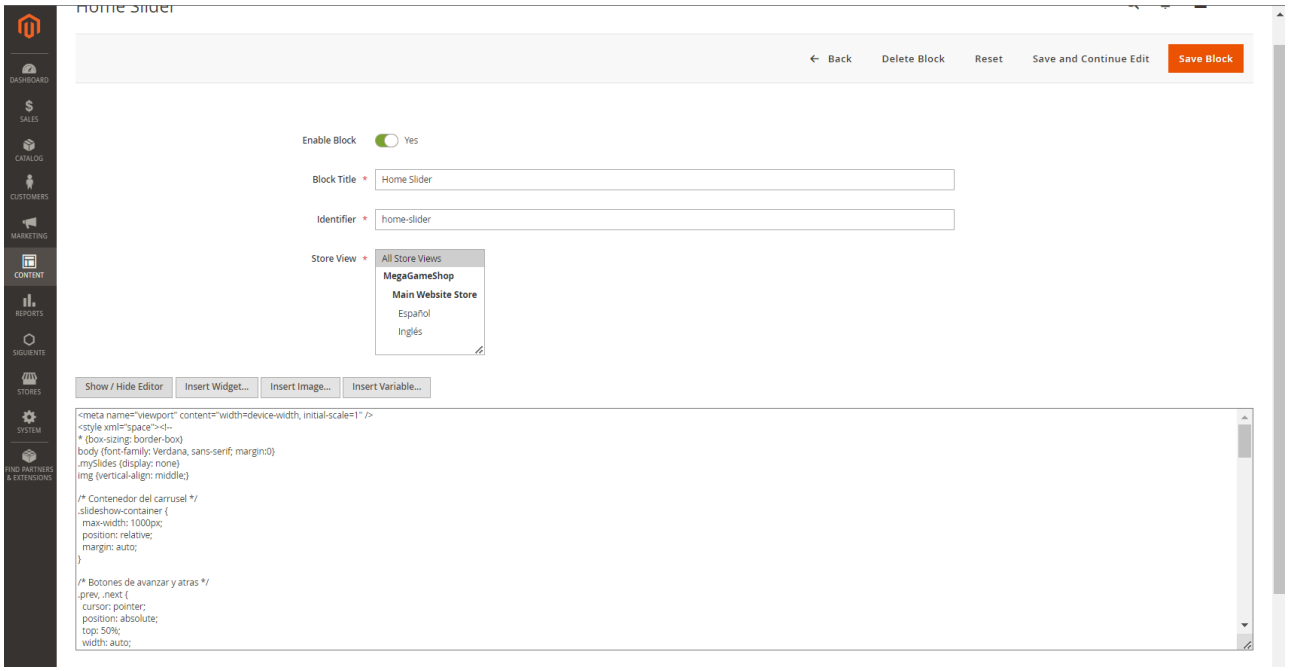

Figura 81. Administración del bloque Home Slider en el backend.

4.6.2.7. Modificaciones en el código.

En este apartado veremos las modificaciones más relevantes que hemos realizado en el código fuente a través de estos bloques. Estas modificaciones se han realizado para personalizar la página de inicio

4.6.2.7.1. Bloque Home Slider

Este bloque ha sido creado desde cero, ya que se estableció que nuestra web iba a tener un carrusel en la parte superior. Comentamos las funciones en las siguientes figuras:

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
<style xml="space"><!--
* {box-sizing: border-box}
body {font-family: Verdana, sans-serif; margin:0}
.mySlides {display: none}
img {vertical-align: middle;}

/* Contenedor del carrusel */
.slideshow-container {
  max-width: 1000px;
  position: relative;
  margin: auto;
}

/* Botones de avanzar y atras */
.prev, .next {
  cursor: pointer;
  position: absolute;
  top: 50%;
  width: auto;
  padding: 16px;
  margin-top: -22px;
  color: white;
  font-weight: bold;
  font-size: 18px;
  transition: 0.6s ease;
  border-radius: 0 3px 3px 0;
  user-select: none;
}

/* Colocar el botón de siguiente a la derecha */
.next {
  right: 0;
  border-radius: 3px 0 0 3px;
}

/* Color del background al pasar el ratón */
.prev:hover, .next:hover {
  background-color: rgba(0,0,0,0.8);
}
```

Establecemos las dimensiones del marco del carrusel y su posicionamiento (centrado)

Creamos los botones para desplazarnos por el carrusel

Figura 82. Código fuente del bloque Home Slider (parte I).

```
/* Indicadores de paso de imagen */
.dot {
  cursor: pointer;
  height: 15px;
  width: 15px;
  margin: 0 2px;
  background-color: #bbb;
  border-radius: 50%;
  display: inline-block;
  transition: background-color 0.6s ease;
}

.active, .dot:hover {
  background-color: #717171;
}

/* Animacion */
.fade {
  animation-name: fade;
  animation-duration: 1.5s;
}

@keyframes fade {
  from {opacity: .4}
  to {opacity: 1}
}

/* Decrementar tamaño de texto para pantallas mas pequeñas
@media only screen and (max-width: 300px) {
  .prev, .next,.text {font-size: 11px}
}
--></style>
<div class="slideshow-container">
<div class="mySlides fade"><img style="width: 100%;" src="{{media url="wysiwyg/image1Banner.jpeg"}}" /></div>
<div class="mySlides fade"><img style="width: 100%;" src="{{media url="wysiwyg/image2Banner.jpg"}}" /></div>
<div class="mySlides fade"><a href="http://localhost/megagameshop/merchandising/figuras-pop.html?product_list_limit=15"><img style="width: 100%;" src="{{media url="wysiwyg/image3Banner.jpg"}}" /></a></div>
<a class="prev" onclick="plusSlides(-1)"><</a> <a class="next" onclick="plusSlides(1)">></a></div>
<p></p>
<div style="text-align: center;"><span class="dot" onclick="currentSlide(1)"></span> <span class="dot" onclick="currentSlide(2)"></span> <span class="dot" onclick="currentSlide(3)"></span></div>
<script type="text/javascript" xml="space">// <![CDATA[
//
//
//
//
```

Creamos los indicadores de imagen actual y las restantes

Configuración de las animaciones al pasar de imágenes

Decrementar tamaño del texto para pantallas más pequeñas

Establecemos las rutas de las imágenes y los tamaños de las imágenes

Figura 83. Código fuente del bloque Home Slider (parte II).

```
let slideIndex = 1;
showSlides(slideIndex);

function plusSlides(n) {
  showSlides(slideIndex += n);
}

function currentSlide(n) {
  showSlides(slideIndex = n);
}

function showSlides(n) {
  let i;
  let slides = document.getElementsByClassName("mySlides");
  let dots = document.getElementsByClassName("dot");
  if (n > slides.length) {slideIndex = 1}
  if (n < 1) {slideIndex = slides.length}
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
}
//
//
//
//
// ]]></script>
  let i;
  let slides = document.getElementsByClassName("mySlides");
  let dots = document.getElementsByClassName("dot");
  if (n > slides.length) {slideIndex = 1}
  if (n < 1) {slideIndex = slides.length}
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active", "");
  }
  slides[slideIndex-1].style.display = "block";
  dots[slideIndex-1].className += " active";
}
```

Función que avanza la diapositiva en la que nos encontramos

Función que nos devuelve la diapositiva en la que nos encontramos actualmente.

Función que realiza los desplazamientos de imagen.

Script donde el carrusel se pone en movimiento.

Figura 84. Código fuente del bloque Home Slider (parte III).

Esto nos ofrece como resultado un carrusel compuesto por tres imágenes, dos botones a los lados para realizar el desplazamiento y tres indicadores, que nos sirven de guía para indicarnos la imagen en la que nos encontramos actualmente.

A continuación, vemos cómo quedaría el carrusel en el frontend para la versión web:

Figura 85. Carrusel de imágenes en el frontend para versión web de MegaGameShop.

En la siguiente figura se muestra cómo quedaría el carrusel para la vista para Smartphone:

Figura 86. Carrusel de imágenes en el frontend para versión Smartphone de MegaGameShop.

4.6.2.7.2. Bloque Home Images Block.

En este pequeño bloque hemos creado dos banners publicitarios, que hemos diseñado nosotros mismos:

-El primero lo situaremos a la izquierda de la vista, haciendo referencia a una de nuestras subcategorías. En concreto, se trata de Consolas Nintendo Switch. Al pulsar en este banner, nos redireccionará a esta subcategoría.

-El segundo lo situaremos a la derecha de la vista, haciendo referencia a nuestro código de descuento “ACCESORIOS30”. Este banner nos redireccionará a una subcategoría mostrándonos los productos donde es efectivo este código.

El código fuente para estos banners es el siguiente:

```
<div class="home-images-wrapper above-padding below-padding">
<div class="page-main">
<div class="home-image-main">
<div class="home-image-inner left"><a href="http://localhost/megagameshop/consolas/nintendo-switch.html"> <img class="image-responsive middle-block" src="{{media url="wysiwyg/switch.jpg"}}" alt="adv-image" /> </a>
</div>
<div class="home-image-inner right"><a href="http://localhost/megagameshop/accesorios/playstation.html"> <img class="image-responsive middle-block" src="{{media url="wysiwyg/bannerAccesorios.png"}}" alt="adv-image"
/> </a></div>
</div>
</div>
</div>
```

Figura 87. Código fuente del bloque Home Images Block.

A continuación, vemos cómo quedarían estos banners en el frontend para la versión web:

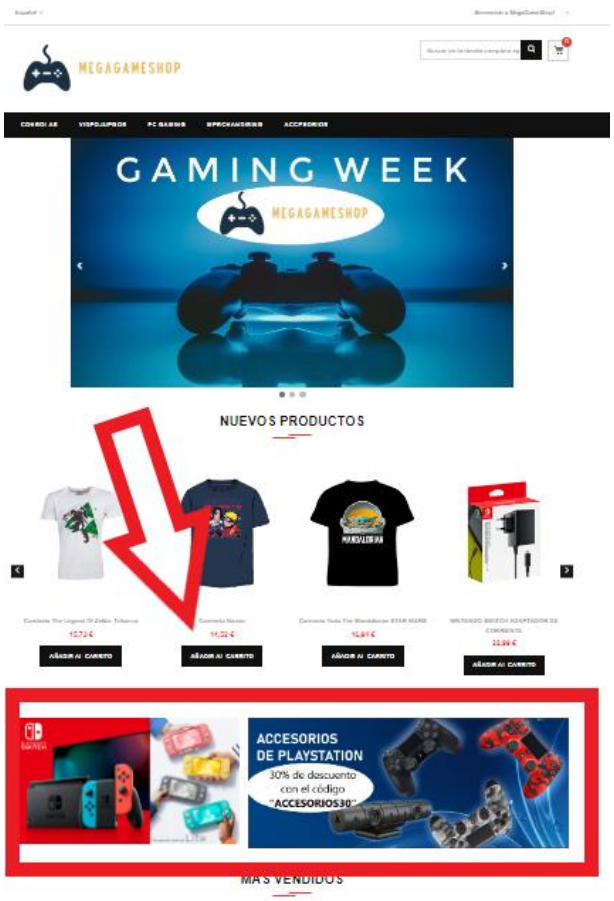

Figura 88. Banners publicitarios en el frontend para versión web de MegaGameShop.

En la siguiente figura se muestra cómo quedaría el carrusel para la vista para Smartphone:

Figura 89. Banners publicitarios en el frontend para versión Smartphone de MegaGameShop.

4.6.2.7.3. Bloque Home Best Sellers.

En este bloque hemos diseñado un carrusel formado por cuatro componentes, donde se muestran los productos más vendidos de nuestro sitio web. Cada componente está compuesto por: imagen, título y descripción del producto.

En las dos siguientes figuras se muestra el código fuente explicando los fragmentos más relevantes:

```
<div class="hb-wrapper above-padding below-padding" style="background: #ffffff;">
<div class="page-main">
<div class="page-title-main"><strong>Más Vendidos</strong></div>
<div class="hb-row">
<div class="hb-blog">
<ul id="hb-blogs">
<li style="display: none;">
<div class="hb-img"><a href="http://localhost/megagameshop/resident-evil-village.html"><img src="{{media url="wysiwyg/blog/residentEvil8.jpg"}}" alt="blog-image" /></a></div>
<div class="hb-content">
<h3>Resident Evil Village PS5</h3>
<p align="justify">Ambientado después de los terroríficos eventos del aclamado Resident Evil 7 Biohazard la historia comienza con Ethan Winters y su mujer Mia
viviendo pacíficamente en una nueva localidad, libre de sus pasadas pesadillas. Justo cuando empiezan a construir su nueva vida juntos la tragedia se ceba con ellos de nuevo.
Chris Redfield el héroe legendario de la saga Resident Evil vuelve a ponerse en contacto con la pareja y entra abruptamente en sus vidas, sumergiendo a Ethan en el caos absoluto
Devastado, Ethan se encuentra en un remoto pueblo cubierto por la nieve buscando respuestas después de ser arrojado a una pesadilla completamente nueva.</p>
</div>
</li>
<li style="display: none;">
<div class="hb-img"><a href="http://localhost/megagameshop/god-of-war.html"><img src="{{media url="wysiwyg/blog/godOfWar.jpg"}}" alt="blog-image" /></a></div>
<div class="hb-content">
<h3>God Of War PS4</h3>
<p align="justify">Vuelve el héroe más icónico de PlayStation de la historia, PS4 God Of War en edición Hits. La motivación de Kratos ha dejado de ser la venganza.
Ahora quiere tomarse el tiempo necesario para ser padre, hacer por fin algo bueno sin ninguna maldad. Adéntrate en una aventura compleja y desconocida en la que explorarás una
impresionante versión alternativa de una historia clásica. Únete a Kratos en un profundo viaje de reflexión, crecimiento, violencia y el deseo de convertirse en un hombre mejor
por su hijo. En un mundo de monstruos, dragones y dioses, el peso de la responsabilidad recae sobre tus hombros. Y cuando tu vida está marcada por la sangre y el caos, ¿cómo
puedes evitar los errores del pasado para proteger tu futuro?</p>
</div>
</li>
<ul>
<li style="display: none;">
<div class="hb-img"><a href="http://localhost/megagameshop/marvel-spider-man.html"><img src="{{media url="wysiwyg/blog/spidermanPs4.jpg"}}" alt="blog-image" /></a></div>
<div class="hb-content">
<h3>Marvel Spider-Man PS4</h3>
<p align="justify">Marvel Spider-Man incluye las habilidades acrobáticas, la improvisación y el lanzamiento de redes por los que es conocido el trepamuros, al tiempo que
introduce nuevos elementos nunca antes vistos en un juego de Spider-Man. Muévete haciendo parkour y utilizando el entorno o prueba el nuevo sistema de combate y escenas
repletas de acción. Este es un juego de Spider-Man como nunca antes has visto.</p>
</div>
</li>
</ul>
<li style="display: none;">
<div class="hb-img"><a href="http://localhost/megagameshop/horizon-forbidden-west.html"><img src="{{media url="wysiwyg/blog/horizon.jpg"}}" alt="blog-image" /></a></div>
<div class="hb-content">
<h3>Horizon Forbidden West PS5</h3>
<p align="justify">Acompaña a Aloy en su aventura por las peligrosas y fantásticas tierras del "Oeste Prohibido" y enfréntate a nuevas y misteriosas amenazas.
Explora páramos remotos, enfréntate a máquinas más grandes e imponentes y descubre increíbles tribus en tu regreso a un futuro lejano en el mundo apocalíptico de Horizon.
La tierra se muere. Las rugientes tormentas y una desolación imparable causan estragos entre lo que queda de la humanidad, unas cuantas tribus dispersas, mientras nuevas y
temibles máquinas merodean por sus fronteras. La vida en la Tierra se enfrenta a una nueva extinción y nadie sabe por qué. Solo Aloy es capaz de descubrir los secretos que
hay detrás de estas amenazas, y restablecer el orden y el equilibrio en el mundo. Por el camino, se reunirá con viejos amigos, forjará alianzas con nuevas facciones en guerra
y descubrirá la herencia de un antiguo pasado, al tiempo que intenta permanecer un paso por delante de un nuevo y aparentemente invencible enemigo.</p>
</div>
</li>
</ul>
```

Título del carrusel de productos más vendidos

Producto 1

Producto 2

Producto 3

Producto 4

Figura 90. Código fuente del bloque Home Best Sellers Block (parte I).

```
</li>
</ul>
<script type="text/javascript" xml="space">// <![CDATA[
//

                    require(['jquery', 'nextowlcarousel'], function() {
                        jQuery(document).ready(function() {
                            jQuery("#hb-blogs").owlCarousel({
                                items: 3,
                                itemsDesktop: [1198, 3],
                                itemsDesktopSmall: [1023, 2],
                                itemsTablet: [768, 1],
                                itemsTabletSmall: [767, 1],
                                itemsMobile: [479, 1],
                                pagination: false,
                                navigationText: ["<div class='left-arrow'><i class='fa fa-angle-left'></i></div>", "<div class='right-arrow'><i class='fa fa-angle-right'></div>"],
                                navigation: true,
                            });
                            jQuery("#hb-blogs li").show();
                        });
                    });

// ]]></script>
</div>
</div>
</div>
</div>
</div>
```

Tamaños de los items para cada dispositivo, al ser responsive design

Figura 91. Código fuente del bloque Home Best Sellers Block (parte II).

A continuación, vemos cómo quedaría este slider en el frontend para la versión web:

Figura 92. Slider de los productos más vendidos en el frontend para versión web de MegaGameShop.

Figura 93. Slider de los productos más vendidos en el frontend para versión Smartphone de MegaGameShop.

4.6.2.7.4. Bloque Home Brands.

En este bloque hemos diseñado un slider formado por siete componentes, donde cada uno de ellos muestra una imagen de cada marca que trabaja con nosotros. Este slider se situará en la parte baja de nuestra vista de inicio.

En la siguiente figura se muestra el código fuente explicando los fragmentos más importantes:

```
<div class="brands-wrapper above-padding below-padding below-margin">
<div class="page-main">
<div class="page-title-main"><strong>Nuestras Marcas</strong></div>
<div class="brands-main">
<ul class="brands-slider">
<li style="display: none;"><img src="{{media url="wysiwyg/brand/logoPlaystation.jpg"}}" alt="brand-image" /></li>
<li style="display: none;"><img src="{{media url="wysiwyg/brand/logoXbox.jpg"}}" alt="brand-image" /></li>
<li style="display: none;"><img src="{{media url="wysiwyg/brand/logoNintendo.jpg"}}" alt="brand-image" /></li>
<li style="display: none;"><img src="{{media url="wysiwyg/brand/logoSamsung.png"}}" alt="brand-image" /></li>
<li style="display: none;"><img src="{{media url="wysiwyg/brand/logoMsi.jpg"}}" alt="brand-image" /></li>
<li style="display: none;"><img src="{{media url="wysiwyg/brand/logoLenovo.jpg"}}" alt="brand-image" /></li>
<li style="display: none;"><img src="{{media url="wysiwyg/brand/logoLg.jpg"}}" alt="brand-image" /></li>
</ul>
<script type="text/javascript" xml="space">// <![CDATA[

                require(['jquery', 'nextowlcarousel'], function(
                    jQuery(document).ready(function() {
                        jQuery(".brands-slider").owlCarousel({
                            items: 7,
                            itemsDesktop: [1198, 5],
                            itemsDesktopSmall: [1023, 5],
                            itemsTablet: [768, 4],
                            itemsTabletSmall: [767, 4],
                            itemsMobile: [479, 2],
                            pagination: false,
                            navigationText: ["<div class='left-arrow'><i class='fa fa-angle-left'></i></div>", "<div class='right-arrow'><i class='fa fa-angle-right'></div>"],
                            navigation: true,
                        });
                        jQuery(".brands-slider li").show();
                    });
                });

// ]]></script>
</div>
</div>
</div>
```

Título del slider

Cargamos las imágenes de nuestro slider

Tamaños de los items para cada dispositivo, al ser responsive design

Figura 94. Código fuente del bloque Home Brands.

A continuación, vemos cómo se muestra este slider en el frontend para la versión web:

Figura 95. Slider de Nuestras Marcas en el frontend para versión web de MegaGameShop.

En la siguiente figura se muestra cómo quedaría el slider para la vista para Smartphone:

Figura 96. Slider de Nuestras Marcas en el frontend para versión Smartphone de MegaGameShop.

4.6.2.7.5. Bloque con política de privacidad y cookies.

En este bloque hemos redactado toda la documentación de política de privacidad de la empresa y cookies, accesible desde el footer de nuestra web en el apartado “Política de Privacidad y Cookies”. La hemos dividido en las siguientes secciones tal y como vemos en el código fuente:

```
<dl><dt><span>Política de Privacidad de MegaGameShop</span></dt><dd>
<p>Esta Política de privacidad describe cómo se recopila, utiliza y comparte su información personal cuando visita o hace una compra en MegaGameShop.</p>
<dl><dt><span>INFORMACIÓN PERSONAL QUE RECOPILAMOS</span></dt><dd>
<p>Cuando visita el Sitio, recopilamos automáticamente cierta información sobre su dispositivo, incluida información sobre su navegador web, dirección IP,
zona horaria y algunas de las cookies que están instaladas en su dispositivo. Además, a medida que navega por el Sitio, recopilamos información sobre las
páginas web individuales o los productos que ve, las páginas web o los términos de búsqueda que lo remitieron al Sitio e información sobre cómo interactúa
usted con el Sitio. Nos referimos a esta información recopilada automáticamente como “Información del dispositivo”.</p>
<p>Recopilamos Información del dispositivo mediante el uso de las siguientes tecnologías:</p>

<dl><dt><span>COOKIES</span></dt><dd>
<p>Aquí tienes una lista de cookies que utilizamos. Las listamos para que puedas elegir si quieres optar por quitarlas o no.</p>
<p>_session_id, unique token, sessional, Allows Shopify to store information about your session (referrer, landing page, etc).</p>
<p>_magento_visit, no data held, Persistent for 30 minutes from the last visit, Used by our website provider’s internal stats tracker
to record the number of visits</p>
<p>_magento_uniq, no data held, expires midnight (relative to the visitor) of the next day, Counts the number of visits to a store by a single customer.</p>
<p>cart, unique token, persistent for 2 weeks, Stores information about the contents of your cart.</p>
<p>_secure_session_id, unique token, sessional</p>
<p>storefront_digest, unique token, indefinite If the shop has a password, this is used to determine if the current visitor has access.<br />
- Los “Archivos de registro” rastrean las acciones que ocurren en el Sitio y recopilan datos, incluyendo su dirección IP, tipo de navegador,
proveedor de servicio de Internet, páginas de referencia/salida y marcas de fecha/horario.<br /> - Las “balizas web”, las “etiquetas” y los
“píxeles” son archivos electrónicos utilizados para registrar información sobre cómo navega usted por el Sitio.<br /><br />Además, cuando hace
una compra o intenta hacer una compra a través del Sitio, recopilamos cierta información de usted, entre la que se incluye su nombre, dirección
de facturación, dirección de envío, información de pago (incluidos los números de la tarjeta de crédito), dirección de correo electrónico y número
de teléfono. Nos referimos a esta información como “Información del pedido”.</p>
<p>Cuando hablamos de “Información personal” en la presente Política de privacidad, nos referimos tanto a la Información del dispositivo como a la Información
del pedido.</p>

<dl><dt><span>¿CÓMO UTILIZAMOS SU INFORMACIÓN PERSONAL?</span></dt><dd>
<p>Usamos la Información del pedido que recopilamos en general para preparar los pedidos realizados a través del Sitio (incluido el procesamiento de su información
de pago, la organización de los envíos y la entrega de facturas y/o confirmaciones de pedido). Además, utilizamos esta Información del pedido para: comunicarnos con
 usted;<br />examinar nuestros pedidos en busca de fraudes o riesgos potenciales; y cuando de acuerdo con las preferencias que usted compartió con nosotros, le
proporcionamos información o publicidad relacionada con nuestros productos o servicios.</p>
<p><br />Utilizamos la Información del dispositivo que recopilamos para ayudarnos a detectar posibles riesgos y fraudes (en particular, su dirección IP) y, en general,
 para mejorar y optimizar nuestro Sitio (por ejemplo, al generar informes y estadísticas sobre cómo nuestros clientes navegan e interactúan con el Sitio y para evaluar
el éxito de nuestras campañas publicitarias y de marketing).</p>

<br /><dl><dt><span>COMPARTIR SU INFORMACIÓN PERSONAL</span></dt><dd>
<p>Compartimos su Información personal con terceros para que nos ayuden a utilizar su Información personal, tal como se describió anteriormente. Por ejemplo, utilizamos
la tecnología de Shopify en nuestra tienda online. Puede obtener más información sobre cómo Shopify utiliza su Información personal aquí: https://www.shopify.com/legal/privacy.
También utilizamos Google Analytics para ayudarnos a comprender cómo usan nuestros clientes el Sitio. Puede obtener más información sobre cómo Google utiliza su Información personal
aquí: https://www.google.com/intl/es/policies/privacy/. Puede darse de baja de Google Analytics aquí: https://tools.google.com/dlpage/gaoptout.</p>
<p>Finalmente, también podemos compartir su Información personal para cumplir con las leyes y regulaciones aplicables, para responder a una citación,
orden de registro u otra solicitud legal de información que recibamos, o para proteger nuestros derechos.<br /><br /></p>
<dl><dt><span>PUBLICIDAD ORIENTADA POR EL COMPORTAMIENTO</span></dt><dd><br />Como se describió anteriormente, utilizamos su Información personal para proporcionarle anuncios
publicitarios dirigidos o comunicaciones de marketing que creemos que pueden ser de su interés. Para más información sobre cómo funciona la publicidad dirigida, puede visitar la
página educativa de la Iniciativa Publicitaria en la Red (“NAI” por sus siglas en inglés) en http://www.networkadvertising.org/understanding-online-advertising/how-does-it-work.</dd>
</dl><dl><dt><span>SUS DERECHOS</span></dt><dd><br />Si usted es un residente europeo, tiene derecho a acceder a la información personal que tenemos sobre usted y a solicitar que su
información personal sea corregida, actualizada o eliminada. Si desea ejercer este derecho, comuníquese con nosotros a través de la información de contacto que se encuentra a continuación.<br />
Además, si usted es un residente europeo, tenemos en cuenta que estamos procesando su información para cumplir con los contratos que podamos tener con usted (por ejemplo, si realiza un pedido
 a través del Sitio) o para perseguir nuestros intereses comerciales legítimos enumerados anteriormente. Además, tenga en cuenta que su información se transferirá fuera de Europa, incluidos
Canadá y los Estados Unidos.</dd></dl>
<p></p>

<dl><dt><span>RETENCIÓN DE DATOS</span></dt><dd><br />Cuando realiza un pedido a través de MegaGameShop, mantendremos su Información del pedido para nuestros registros a menos que y hasta que
nos pida que eliminemos esta información.</dd></dl><dl><dt><span>CAMBIOS</span></dt><dd><br />Podemos actualizar esta política de privacidad periódicamente para reflejar, por ejemplo, cambios
en nuestras prácticas o por otros motivos operativos, legales o reglamentarios.</dd></dl><dl><dt><span>CONTÁCTENOS</span></dt><dd><br />Para obtener más información sobre nuestras prácticas de
privacidad, si tiene alguna pregunta o si desea presentar una queja, contáctenos por correo electrónico a megagameshop@gmail.com o por correo mediante el uso de la información que se proporciona
 a continuación:</dd></dl>
<p>Calle Navas, n 13, Lucena, CO, 14900, España</p>
</dd></dl></dd></dl></dd></dl></dd></dl>
```

Política de Privacidad

Cookies

¿Cómo utilizamos su información personal?

Compartir su información personal

Publicidad orientada por el comportamiento

Retención de datos

Figura 97. Código fuente de política de privacidad y cookies.

A continuación, vemos cómo se muestra en el frontend para la versión web:

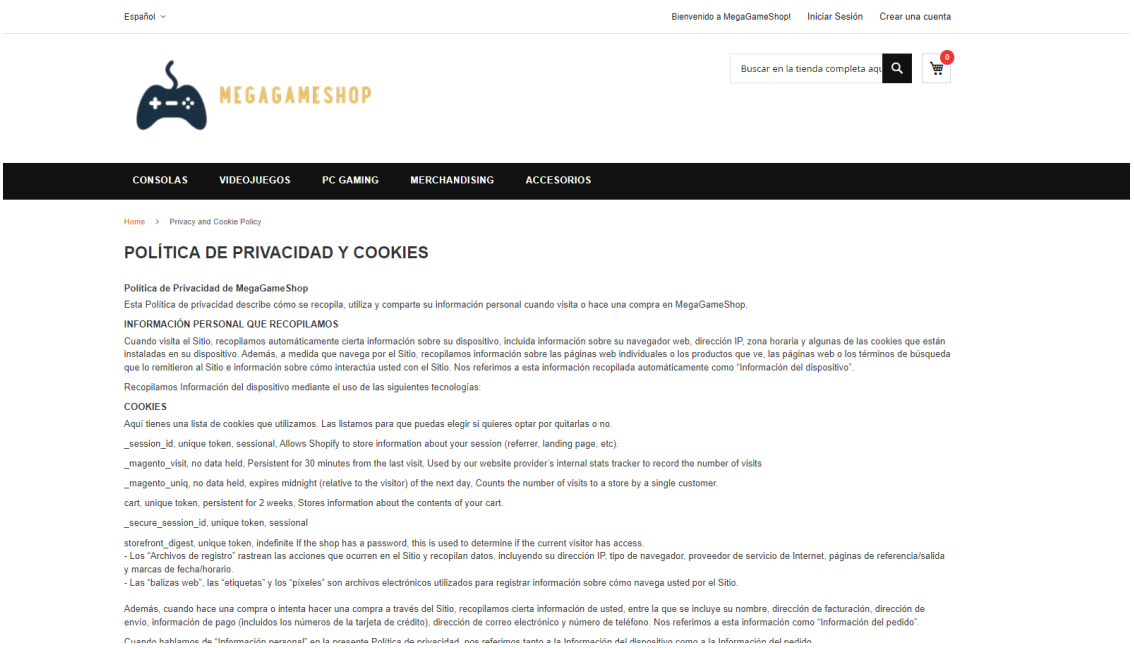

Figura 98. Política de privacidad y cookies en el frontend para versión web de MegaGameShop.

En la siguiente figura se muestra cómo se mostraría para la vista para Smartphone:

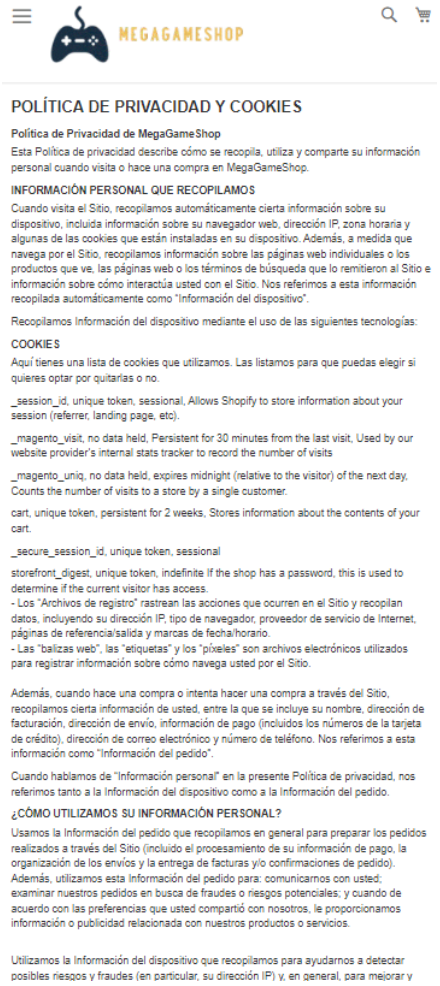

Figura 99. Política de privacidad y cookies en el frontend para versión Smartphone de MegaGameShop.

4.6.2.8. Administración de cupones de descuento.

En esta sección vamos a detallar cómo hemos creado el cupón de descuento “ACCESORIOS30”, para que el cliente pueda obtener un 30% de descuento en accesorios de Playstation. Para ello, debemos ir a la sección **Marketing < Cart Price Rules < Add New Rule.** Debemos completar los siguientes campos, donde estableceremos los clientes que pueden utilizarlo (en este caso, todos los clientes), el código de descuento del cupón, el número de usos del mismo y la categoría a la que queremos aplicarlo (a la subcategoría “Playstation”, dentro de la categoría “ACCESORIOS”).

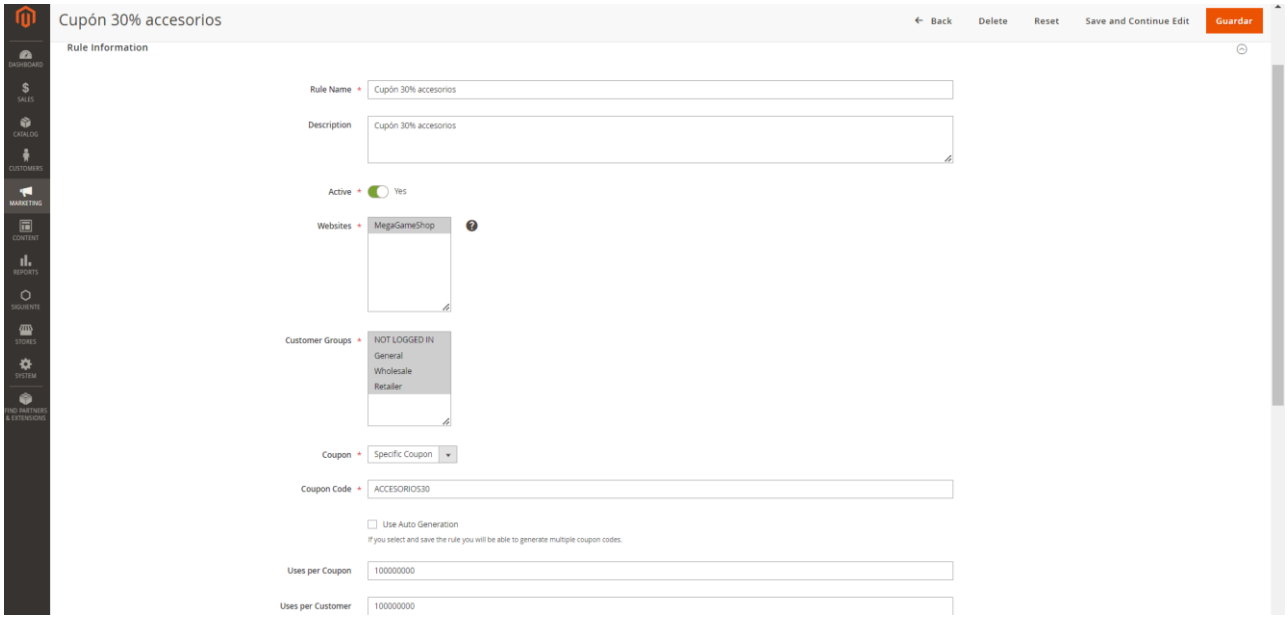

Figura 100. Administración de cupones de descuento en el backend.

4.6.2.9. Administración de métodos de envío.

Para administrar los métodos de envío, debemos entrar en la sección **Stores < Configuration < Sales < Shipping Methods.** Una vez dentro, como nuestra tienda dispone solamente de envío gratuito a través de Correos, inhabilitamos todos los métodos de envío y habilitamos solamente el envío gratuito. En la siguiente figura se muestran los campos que debemos rellenar:

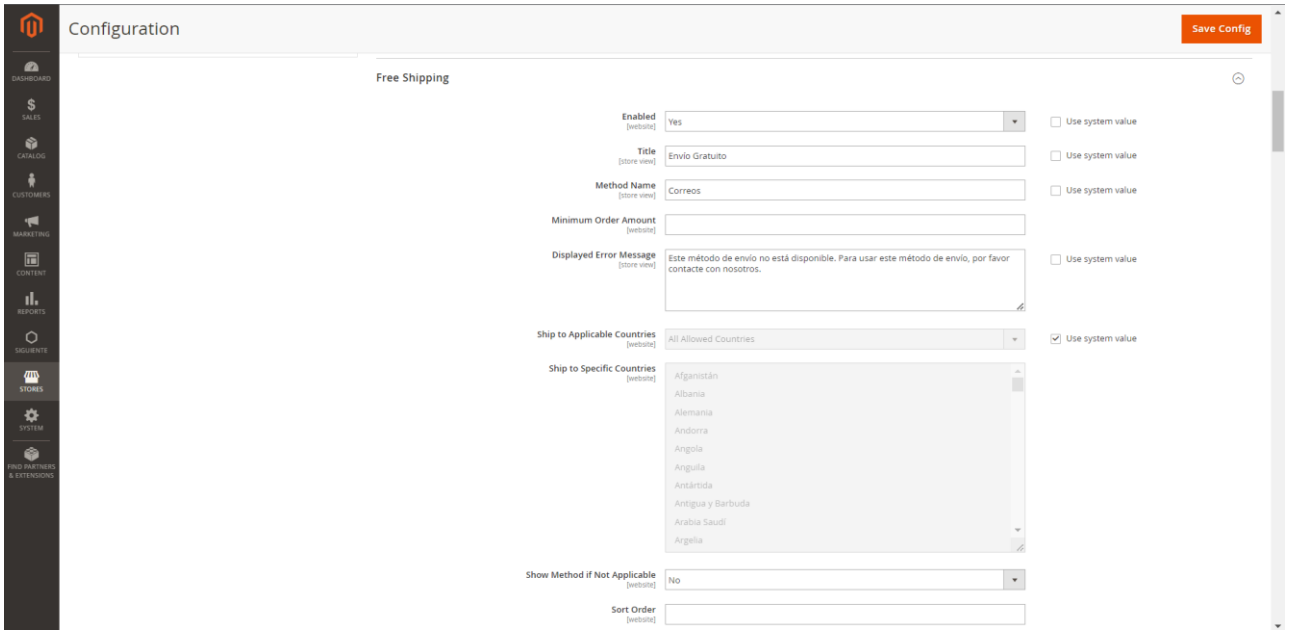

Figura 101. Administración de métodos de envío en el backend.

4.6.2.10. Administración de métodos de pago.

En este apartado, configuraremos los distintos métodos de pago que ofreceremos al usuario. Esto se realizará desde la sección **Stores < Configuration < Sales < Payment Methods**. Aquí encontramos los métodos de pago disponibles para nuestra tienda. Configuraremos el pago mediante Paypal, contrareembolso, transferencia bancaria y tarjeta de crédito o débito.

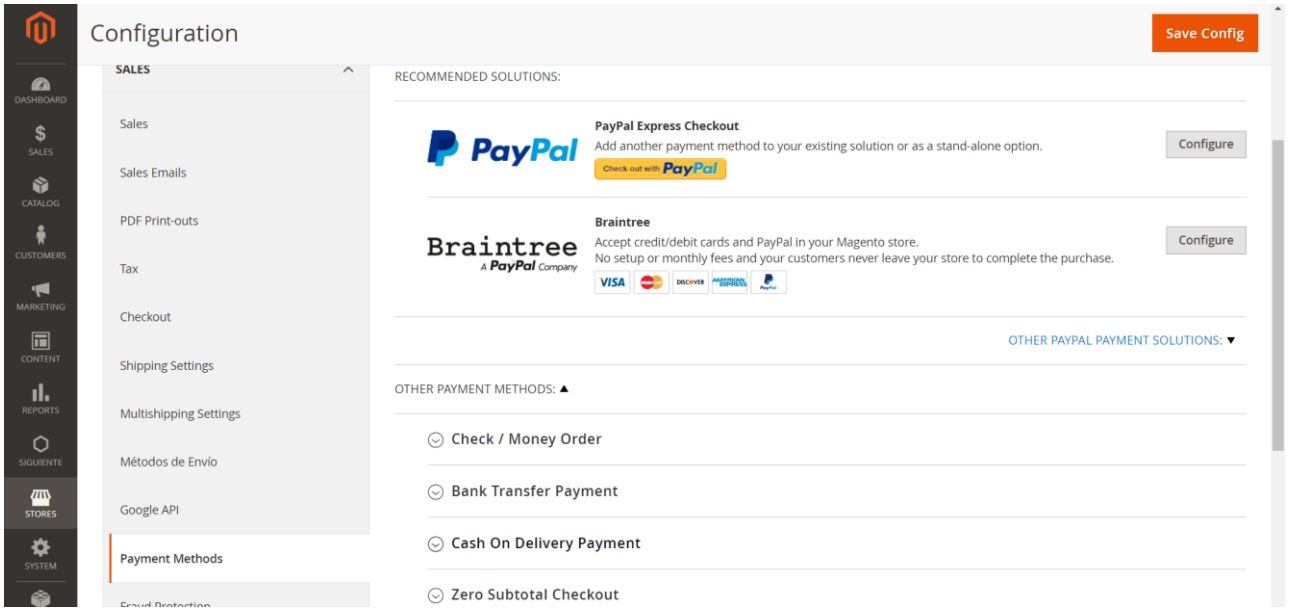

Figura 102. Administración de métodos de pago en el backend.

Esto ofrecerá como resultado los siguientes métodos de pago al cliente en el frontend para la versión web:

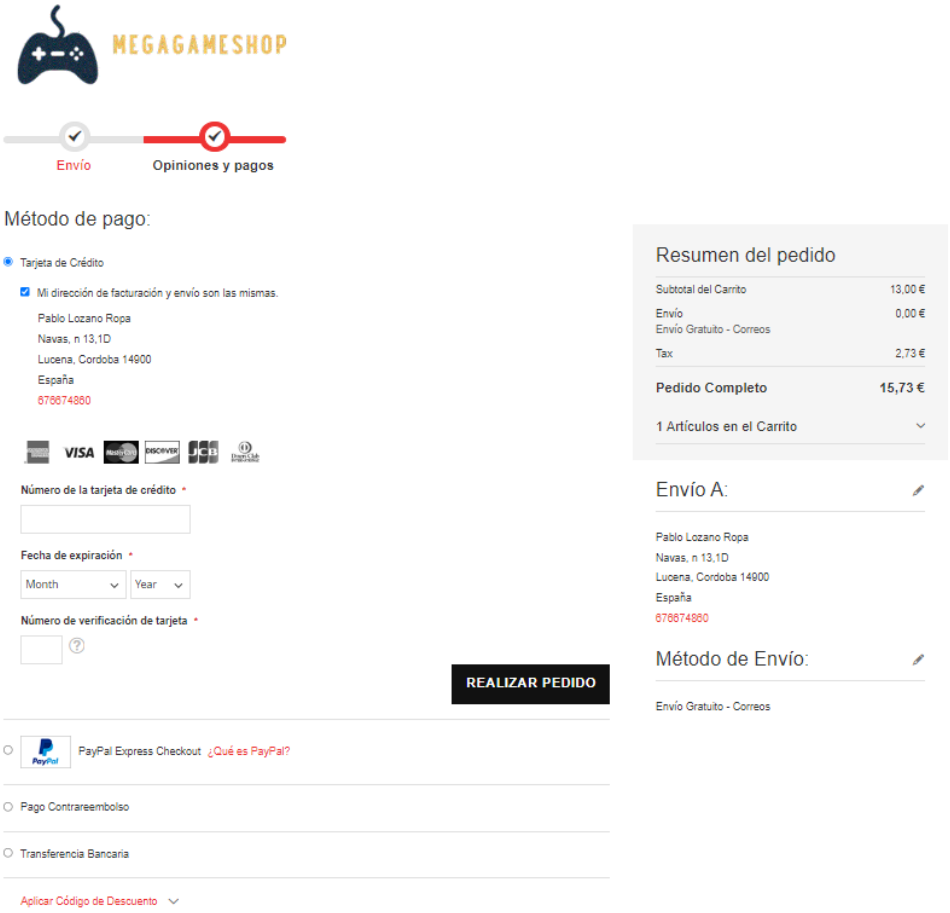

Figura 103. Vista de métodos de pago en el frontend para versión web de MegaGameShop.

En la siguiente figura se muestra cómo se mostraría para la vista para Smartphone:

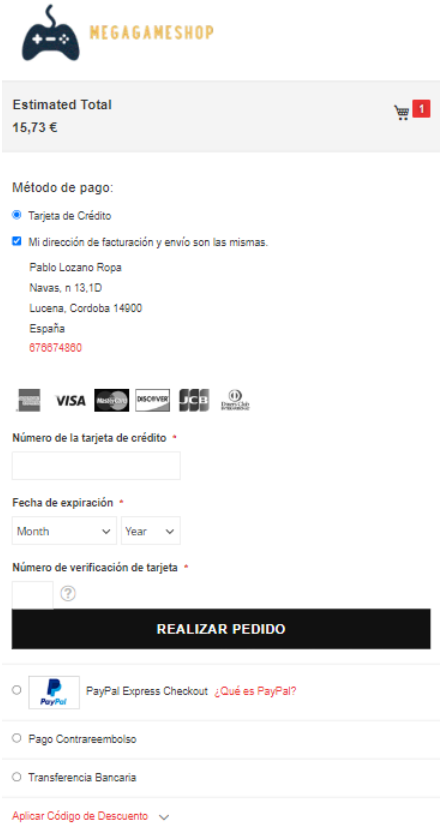

Figura 104. Vista de métodos de pago en el frontend para versión Smartphone de MegaGameShop.

4.7. Pruebas.

En este apartado realizaremos pruebas automáticas de funcionamiento mediante la herramienta Selenium IDE. Estas pruebas nos proporcionarán un feedback sobre el correcto funcionamiento de nuestra aplicación web. Hemos realizado las siguientes pruebas:

4.7.1. Prueba de añadir un producto al carrito.

En este apartado, añadiremos un producto al carrito de manera correcta. Esto podrá realizarlo tanto un usuario registrado como un usuario invitado. Hemos agregado un videojuego al carrito para realizar la prueba. Para ello, necesitamos hacer la siguiente secuencia de acciones en Selenium:

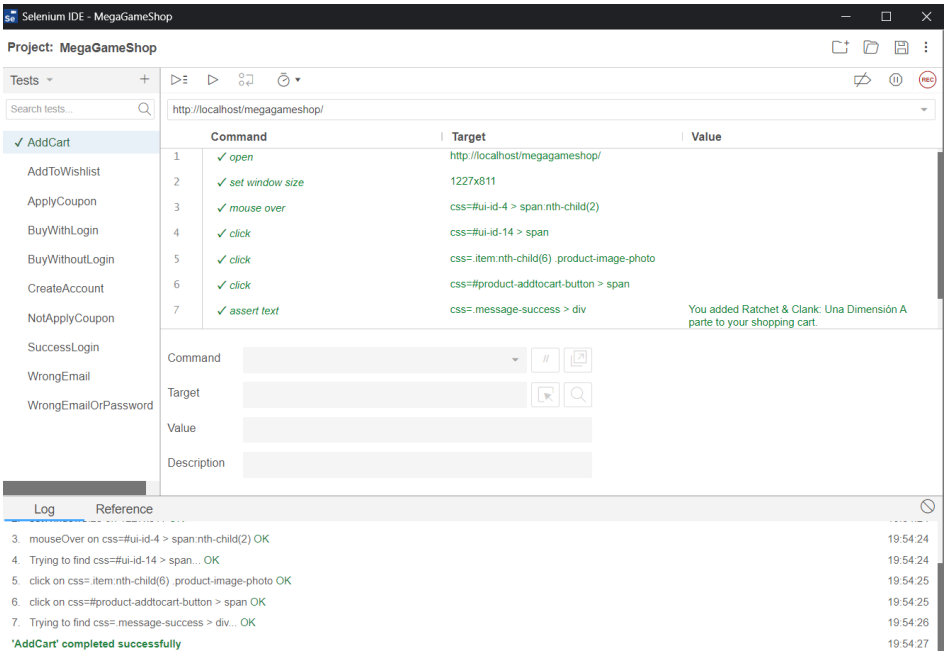

Figura 105. Prueba de agregar un producto al carrito en Selenium.

En el log del programa vemos cómo se cumple toda la secuencia de acciones hasta que se muestra el siguiente mensaje de éxito en el frontend. En la siguiente figura se muestra este mensaje:

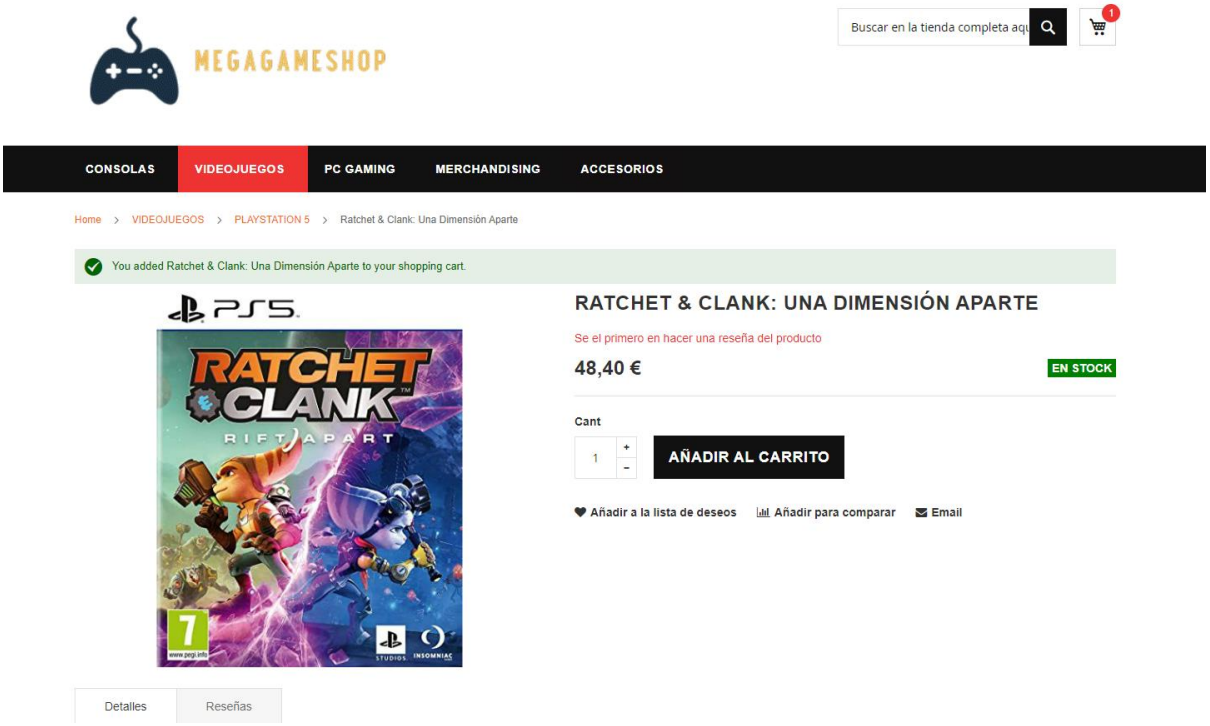

Figura 106. Prueba de agregar un producto al carrito en el frontend de MegaGameShop.

También destacar la posibilidad de exportar el código fuente Java generado por Selenium. En la siguiente figura mostramos el código generado:

```java
*AddCartTest.java
// Generated by Selenium IDE
import org.junit.Test;
public class AddCartTest {
  private WebDriver driver;
  private Map<String, Object> vars;
  JavascriptExecutor js;
  @Before
  public void setUp() {
    driver = new ChromeDriver();
    js = (JavascriptExecutor) driver;
    vars = new HashMap<String, Object>();
  }
  @After
  public void tearDown() {
    driver.quit();
  }
  @Test
  public void addCart() {
    driver.get("http://localhost/megagameshop/");
    driver.manage().window().setSize(new Dimension(1227, 811));
    {
      WebElement element = driver.findElement(By.cssSelector("#ui-id-4 > span:nth-child(2)"));
      Actions builder = new Actions(driver);
      builder.moveToElement(element).perform();
    }
    driver.findElement(By.cssSelector("#ui-id-14 > span")).click();
    driver.findElement(By.cssSelector(".item:nth-child(6) .product-image-photo")).click();
    driver.findElement(By.cssSelector("#product-addtocart-button > span")).click();
    assertThat(driver.findElement(By.cssSelector(".message-success > div")).getText(), is("You added Ratchet & Clank: Una Dimensión Aparte to your shopping cart."));
  }
}
```

Figura 107. Código fuente de la prueba de agregar un producto al carrito en Selenium.

4.7.2. Prueba de añadir un producto a la lista de deseos.

En este apartado, añadiremos un producto a la lista de deseos de manera correcta. Esto solamente podrá realizarlo un usuario registrado. El proceso consiste en realizar el inicio de sesión con una cuenta válida y, a continuación, hemos agregado un videojuego a la lista de deseos para realizar la prueba. Para ello, necesitamos hacer la siguiente secuencia de acciones en Selenium:

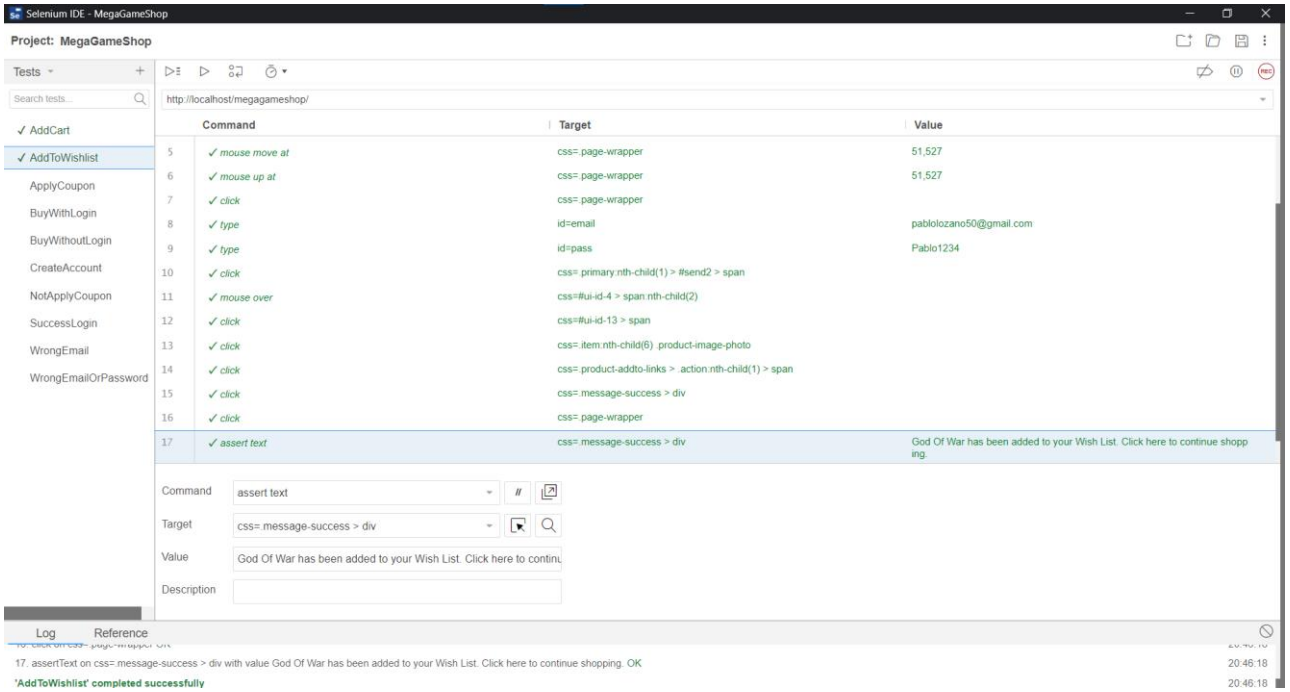

Figura 108. Prueba de agregar un producto a la lista de deseos en Selenium.

En el log del programa vemos cómo se cumple toda la secuencia de acciones hasta que se muestra el siguiente mensaje de éxito en el frontend. En la siguiente figura se muestra este mensaje:

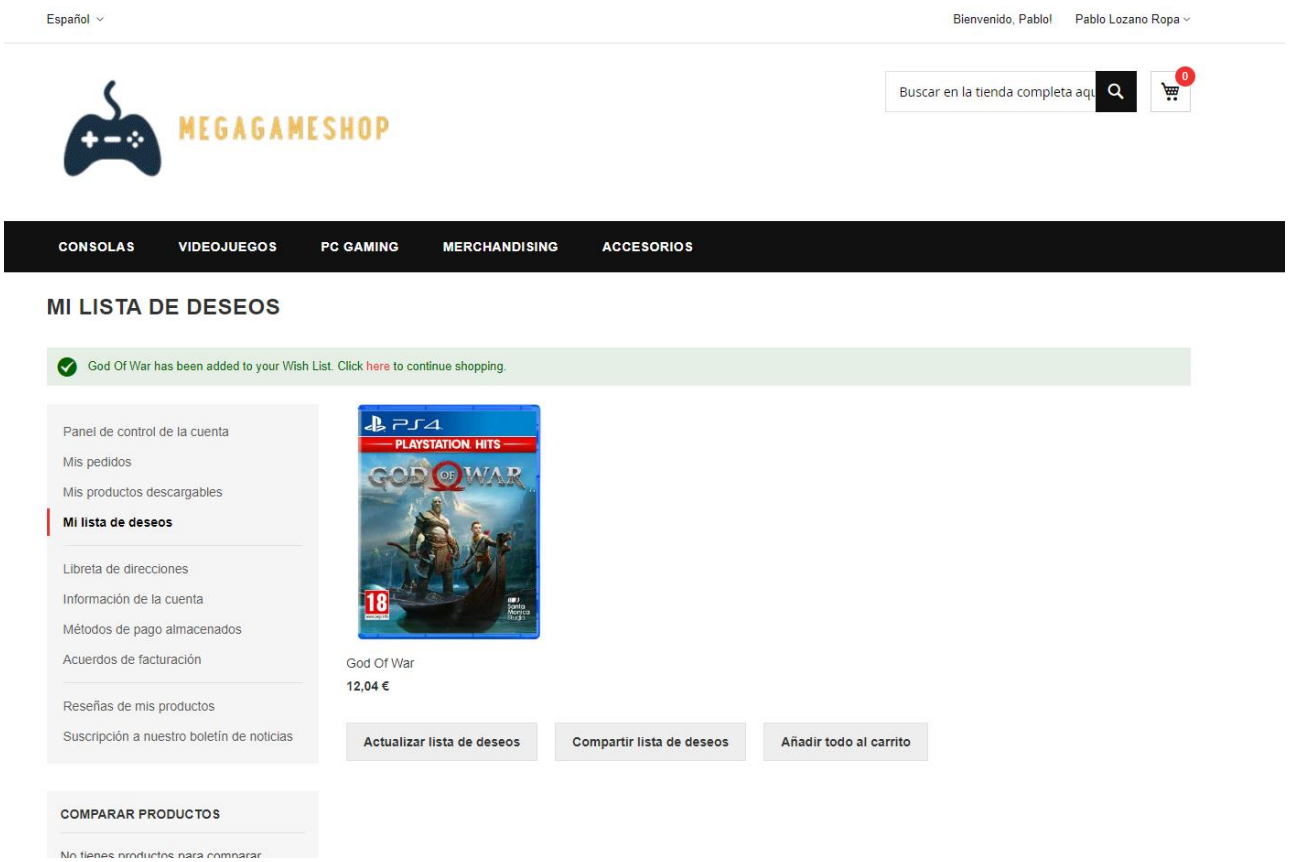

Figura 109. Prueba de agregar un producto a la lista de deseos en el frontend de MegaGameShop.

Exportamos el código fuente Java generado por Selenium, en la siguiente figura mostramos el código generado:

```
AddToWishlistTest.java
2  import org.junit.Test;
25 public class AddToWishlistTest {
26   private WebDriver driver;
27   private Map<String, Object> vars;
28   JavascriptExecutor js;
29   @Before
30   public void setUp() {
31     driver = new ChromeDriver();
32     js = (JavascriptExecutor) driver;
33     vars = new HashMap<String, Object>();
34   }
35   @After
36   public void tearDown() {
37     driver.quit();
38   }
39   @Test
40   public void addToWishlist() {
41     driver.get("http://localhost/megagameshop//megagameshop/");
42     driver.manage().window().setSize(new Dimension(1227, 811));
43     driver.findElement(By.linkText("Iniciar Sesión")).click();
44     {
45       WebElement element = driver.findElement(By.cssSelector(".page-wrapper"));
46       Actions builder = new Actions(driver);
47       builder.moveToElement(element).clickAndHold().perform();
48     }
49     {
50       WebElement element = driver.findElement(By.cssSelector(".page-wrapper"));
51       Actions builder = new Actions(driver);
52       builder.moveToElement(element).perform();
53     }
54     {
55       WebElement element = driver.findElement(By.cssSelector(".page-wrapper"));
56       Actions builder = new Actions(driver);
57       builder.moveToElement(element).release().perform();
58     }
59     driver.findElement(By.cssSelector(".page-wrapper")).click();
60     driver.findElement(By.id("email")).sendKeys("pablolozano50@gmail.com");
61     driver.findElement(By.id("pass")).sendKeys("Pablo1234");
62     driver.findElement(By.cssSelector(".primary:nth-child(1) > #send2 > span")).click();
63     {
64       WebElement element = driver.findElement(By.cssSelector("#ui-id-4 > span:nth-child(2)"));
65       Actions builder = new Actions(driver);
66       builder.moveToElement(element).perform();
67     }
68     driver.findElement(By.cssSelector("#ui-id-13 > span")).click();
69     driver.findElement(By.cssSelector(".item:nth-child(6) .product-image-photo")).click();
70     driver.findElement(By.cssSelector(".product-addto-links > .action:nth-child(1) > span")).click();
71     driver.findElement(By.cssSelector(".message-success > div")).click();
72     driver.findElement(By.cssSelector(".page-wrapper")).click();
73     assertThat(driver.findElement(By.cssSelector(".message-success > div")).getText(), is("God Of War has been added to your Wish List. Click here to continue shopping."));
74   }
```

Figura 110. Código fuente de la prueba de agregar un producto a la lista de deseos en Selenium.

4.7.3. Prueba de aplicar cupón de descuento a un producto con éxito.

En este apartado, aplicaremos el cupón de descuento “ACCESORIOS30” a un producto de manera correcta. Este cupón nos ofrecerá un 30% de descuento en accesorios de Playstation. Primero, debemos agregar al carrito un producto de la categoría y subcategoría para la que hemos creado este cupón. Tras agregar el producto, se podrá ingresar el código de descuento y, si es correcto, se aplicará correctamente y podrá verse reflejado en el resumen del pedido. Para ello, necesitamos hacer la siguiente secuencia de acciones en Selenium:

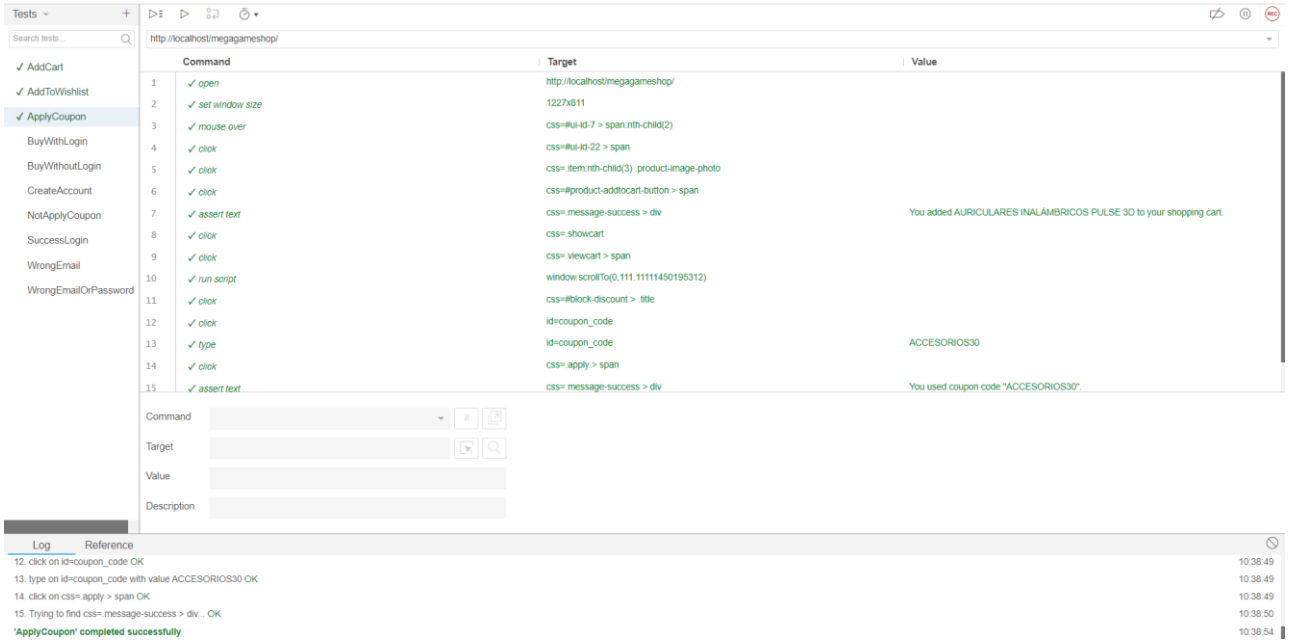

Figura 111. Prueba de aplicar cupón de descuento a un producto con éxito en Selenium.

En el log del programa vemos cómo se cumple toda la secuencia de acciones hasta que se muestra el siguiente mensaje de éxito en el frontend. En la siguiente figura observamos cómo se aplica correctamente el descuento y queda reflejado en el resumen del pedido:

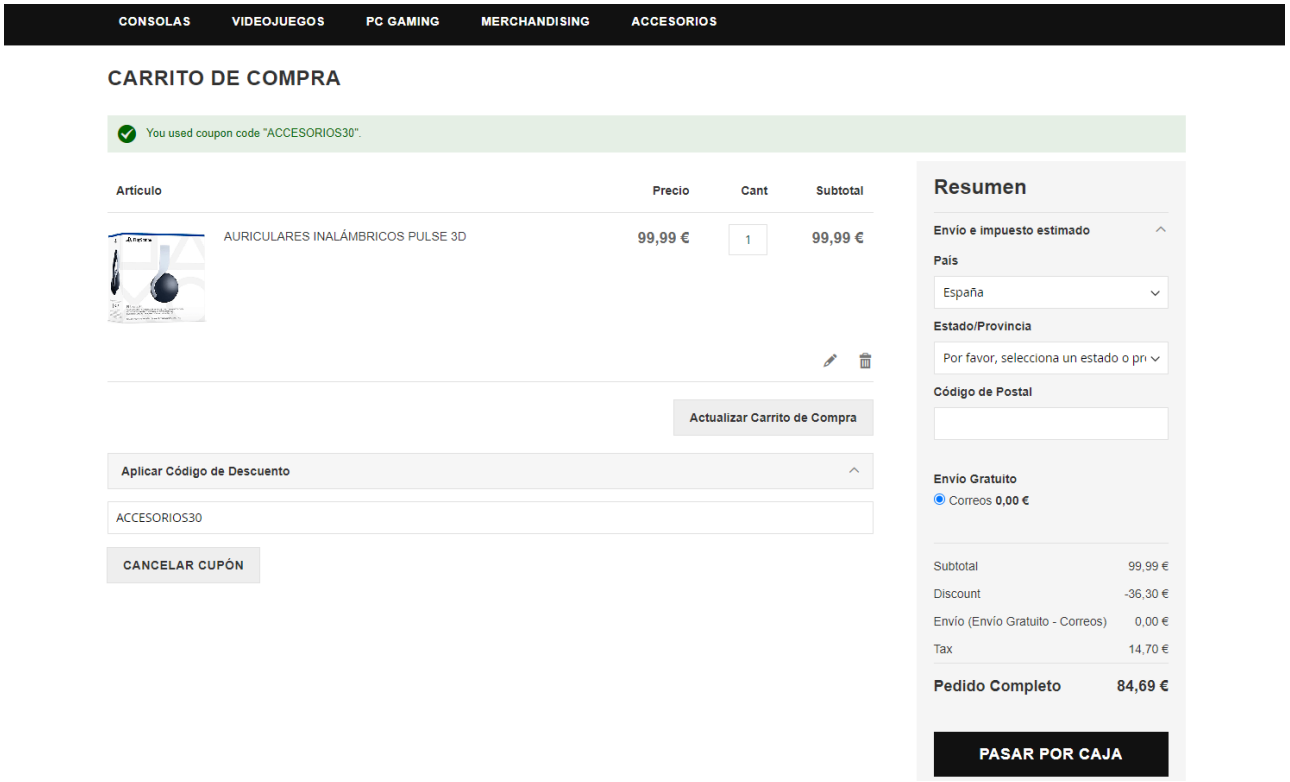

Figura 112. Prueba de aplicar cupón de descuento a un producto con éxito en el frontend de MegaGameShop.

Exportamos el código fuente Java generado por Selenium, en la siguiente figura mostramos el código generado:

```
ApplyCouponTest.java
import org.junit.Test;
public class ApplyCouponTest {
  private WebDriver driver;
  private Map<String, Object> vars;
  JavascriptExecutor js;
  @Before
  public void setUp() {
    driver = new ChromeDriver();
    js = (JavascriptExecutor) driver;
    vars = new HashMap<String, Object>();
  }
  @After
  public void tearDown() {
    driver.quit();
  }
  @Test
  public void applyCoupon() {
    driver.get("http://localhost/megagameshop/");
    driver.manage().window().setSize(new Dimension(1227, 811));
    {
      WebElement element = driver.findElement(By.cssSelector("#ui-id-7 > span:nth-child(2)"));
      Actions builder = new Actions(driver);
      builder.moveToElement(element).perform();
    }
    driver.findElement(By.cssSelector("#ui-id-22 > span")).click();
    driver.findElement(By.cssSelector(".item:nth-child(3) .product-image-photo")).click();
    driver.findElement(By.cssSelector("#product-addtocart-button > span")).click();
    assertThat(driver.findElement(By.cssSelector(".message-success > div")).getText(), is("You added AURICULARES INALÁMBRICOS PULSE 3D to your shopping cart."));
    driver.findElement(By.cssSelector(".showcart")).click();
    driver.findElement(By.cssSelector(".viewcart > span")).click();
    js.executeScript("window.scrollTo(0,111.1111450195312)");
    driver.findElement(By.cssSelector("#block-discount > .title")).click();
    driver.findElement(By.id("coupon_code")).click();
    driver.findElement(By.id("coupon_code")).sendKeys("ACCESORIOS30");
    driver.findElement(By.cssSelector(".apply > span")).click();
    assertThat(driver.findElement(By.cssSelector(".message-success > div")).getText(), is("You used coupon code \\\"ACCESORIOS30\\\".."));
  }
}
```

Figura 113. Código fuente de la prueba de aplicar cupón de descuento a un producto con éxito en Selenium.

4.7.4. Prueba de aplicar cupón de descuento a un producto sin éxito.

El proceso a seguir es el mismo que en el apartado anterior, pero en este caso, al introducir un código erróneo como “ACCESORIOS” se nos mostrará un mensaje de error.

Para ello, necesitamos hacer la siguiente secuencia de acciones en Selenium.

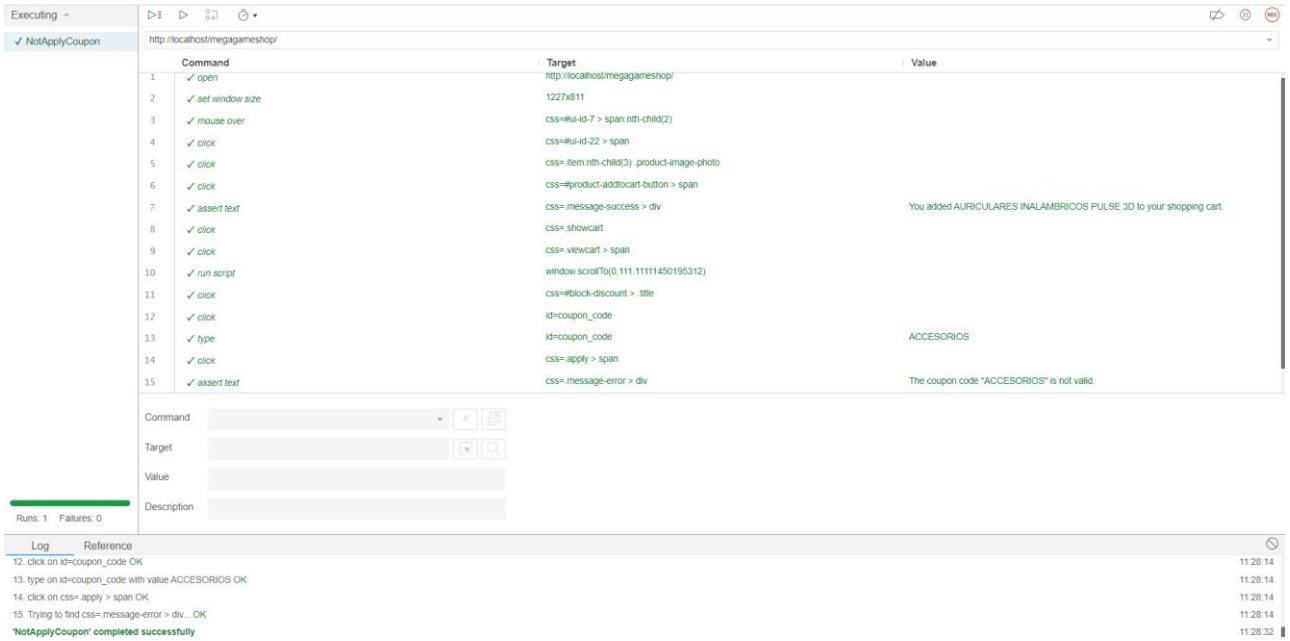

Figura 114. Prueba de aplicar cupón de descuento a un producto sin éxito en Selenium.

En el log del programa vemos cómo se cumple toda la secuencia de acciones hasta que se muestra el siguiente mensaje de error en el frontend. En la siguiente figura observamos cómo se muestra un mensaje de error y en el resumen del pedido no se realiza ningún descuento:

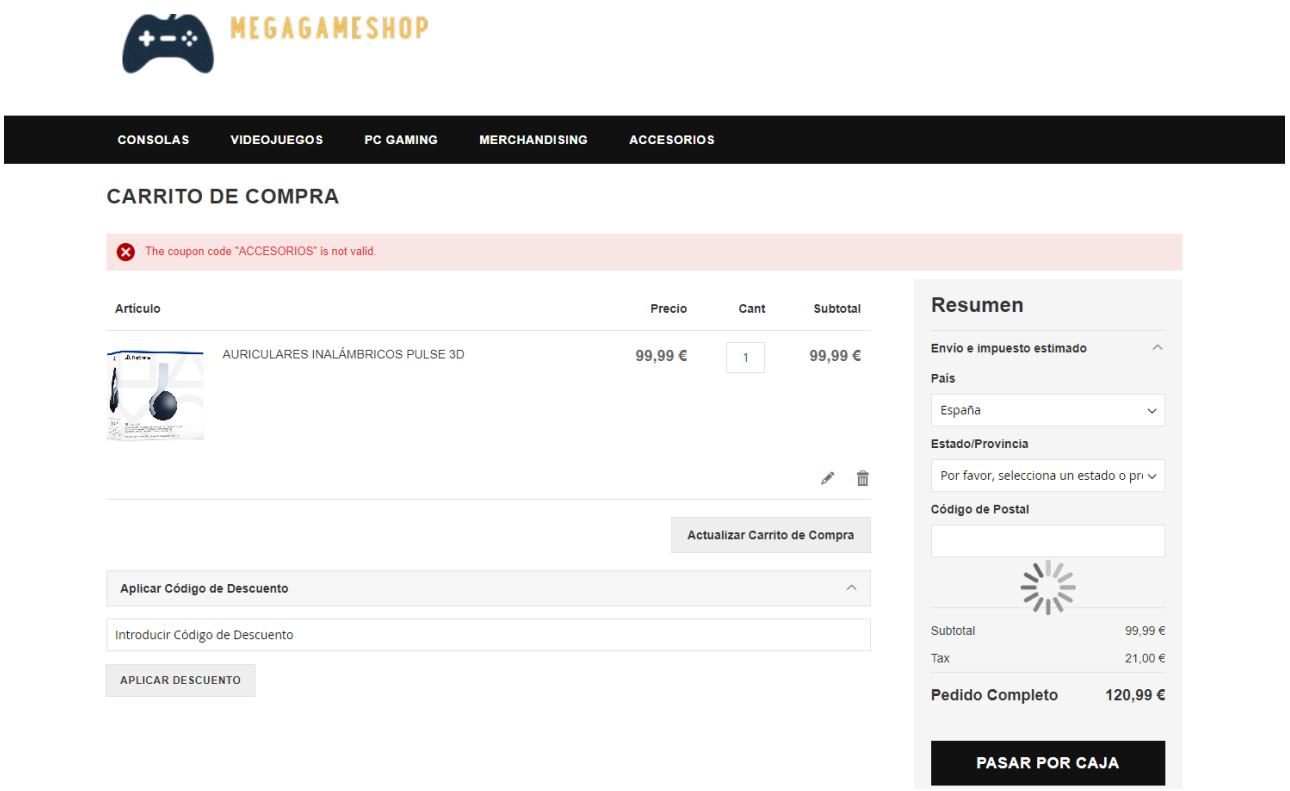

Figura 115. Prueba de aplicar cupón de descuento a un producto sin éxito en el frontend de MegaGameShop.

Exportamos el código fuente Java generado por Selenium, en la siguiente figura mostramos el código generado:

```
NotApplyCouponTest.java
1  // Generated by Selenium IDE
2  import org.junit.Test;
25 public class NotApplyCouponTest {
26   private WebDriver driver;
27   private Map<String, Object> vars;
28   JavascriptExecutor js;
29   @Before
30   public void setUp() {
31     driver = new ChromeDriver();
32     js = (JavascriptExecutor) driver;
33     vars = new HashMap<String, Object>();
34   }
35   @After
36   public void tearDown() {
37     driver.quit();
38   }
39   @Test
40   public void notApplyCoupon() {
41     driver.get("http://localhost/megagameshop/");
42     driver.manage().window().setSize(new Dimension(1227, 811));
43     {
44       WebElement element = driver.findElement(By.cssSelector("#ui-id-7 > span:nth-child(2)"));
45       Actions builder = new Actions(driver);
46       builder.moveToElement(element).perform();
47     }
48     driver.findElement(By.cssSelector("#ui-id-22 > span")).click();
49     driver.findElement(By.cssSelector(".item:nth-child(3) .product-image-photo")).click();
50     driver.findElement(By.cssSelector("#product-addtocart-button > span")).click();
51     assertThat(driver.findElement(By.cssSelector(".message-success > div")).getText(), is("You added AURICULARES INALÁMBRICOS PULSE 3D to your shopping cart."));
52     driver.findElement(By.cssSelector(".showcart")).click();
53     driver.findElement(By.cssSelector(".viewcart > span")).click();
54     js.executeScript("window.scrollTo(0,111.11111450195312)");
55     driver.findElement(By.cssSelector("#block-discount > .title")).click();
56     driver.findElement(By.id("coupon_code")).click();
57     driver.findElement(By.id("coupon_code")).sendKeys("ACCESORIOS");
58     driver.findElement(By.cssSelector(".apply > span")).click();
59     assertThat(driver.findElement(By.cssSelector(".message-error > div")).getText(), is("The coupon code \\\"ACCESORIOS\\\" is not valid."));
60   }
```

Figura 116. Código fuente de la prueba de aplicar cupón de descuento a un producto sin éxito en Selenium.

4.7.5. Prueba de inicio de sesión con éxito.

En este apartado, realizaremos un inicio de sesión con éxito. Para ello, nos dirigiremos a la esquina superior derecha y pulsaremos en “Iniciar Sesión”. Tras introducir un email y una contraseña correcta, tendremos la sesión iniciada con éxito. Para ello, necesitamos hacer la siguiente secuencia de acciones en Selenium:

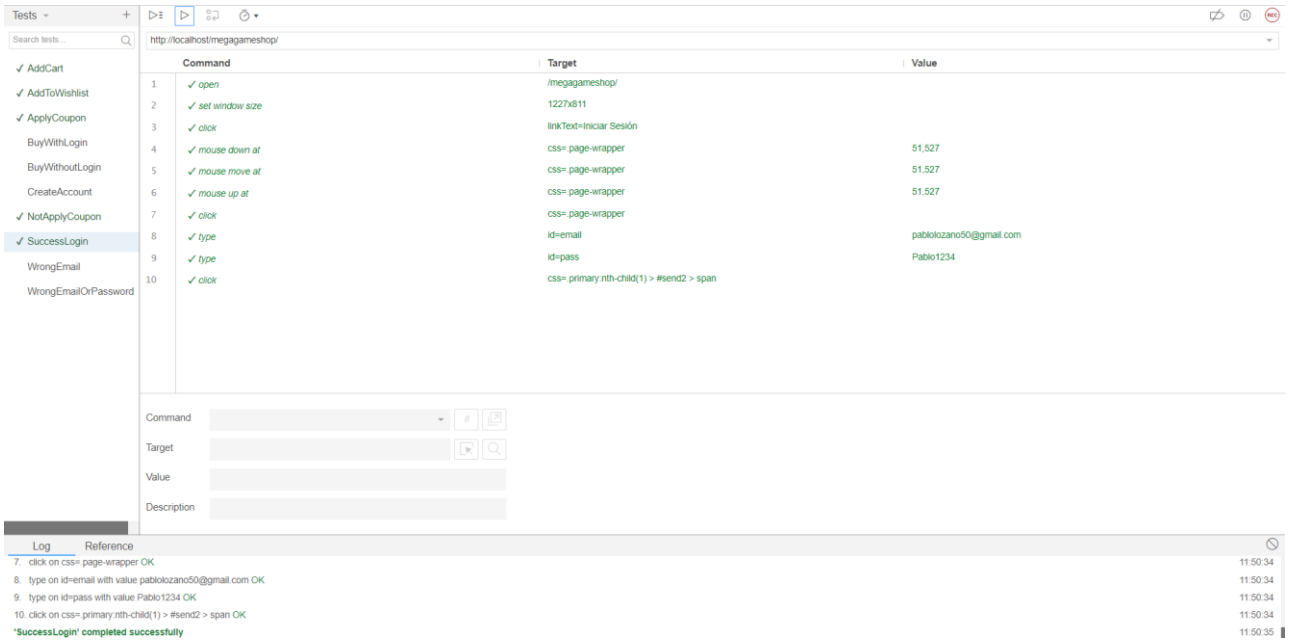

Figura 117. Prueba de iniciar sesión con éxito en Selenium.

En el log del programa vemos cómo se cumple toda la secuencia de acciones hasta que se realiza el inicio de sesión. En la siguiente figura observamos cómo en la cabecera

de nuestro frontend se personaliza el mensaje de bienvenida y se visualiza un desplegable donde disponemos de un enlace con acceso al panel de nuestra cuenta y podremos cerrar la sesión en todo momento:

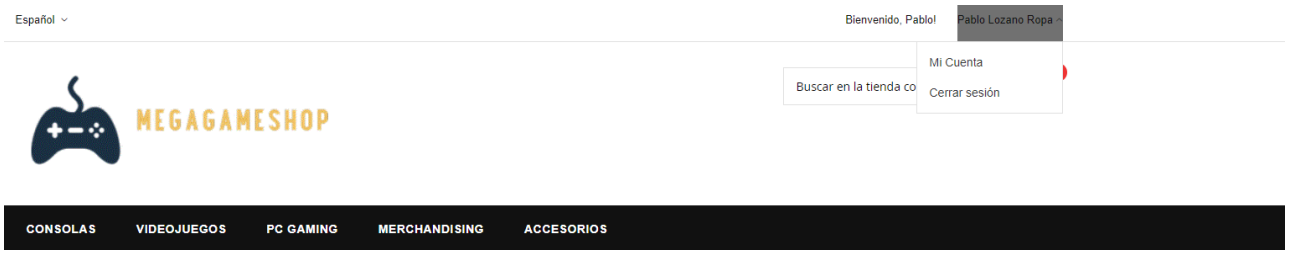

Figura 118. Prueba de iniciar sesión con éxito en el frontend de MegaGameShop.

Exportamos el código fuente Java generado por Selenium, en la siguiente figura mostramos el código generado:

```
SuccessLoginTest.java
1  // Generated by Selenium IDE
2  import org.junit.Test;
25 public class SuccessLoginTest {
26   private WebDriver driver;
27   private Map<String, Object> vars;
28   JavascriptExecutor js;
29   @Before
30   public void setUp() {
31     driver = new ChromeDriver();
32     js = (JavascriptExecutor) driver;
33     vars = new HashMap<String, Object>();
34   }
35   @After
36   public void tearDown() {
37     driver.quit();
38   }
39   @Test
40   public void successLogin() {
41     driver.get("http://localhost/megagameshop//megagameshop/");
42     driver.manage().window().setSize(new Dimension(1227, 811));
43     driver.findElement(By.linkText("Iniciar Sesión")).click();
44     {
45       WebElement element = driver.findElement(By.cssSelector(".page-wrapper"));
46       Actions builder = new Actions(driver);
47       builder.moveToElement(element).clickAndHold().perform();
48     }
49     {
50       WebElement element = driver.findElement(By.cssSelector(".page-wrapper"));
51       Actions builder = new Actions(driver);
52       builder.moveToElement(element).perform();
53     }
54     {
55       WebElement element = driver.findElement(By.cssSelector(".page-wrapper"));
56       Actions builder = new Actions(driver);
57       builder.moveToElement(element).release().perform();
58     }
59     driver.findElement(By.cssSelector(".page-wrapper")).click();
60     driver.findElement(By.id("email")).sendKeys("pablolozano50@gmail.com");
61     driver.findElement(By.id("pass")).sendKeys("Pablo1234");
62     driver.findElement(By.cssSelector(".primary:nth-child(1) > #send2 > span")).click();
63   }
64 }
```

Figura 119. Código fuente de la prueba de iniciar sesión con éxito en Selenium.

4.7.6. Prueba de inicio de sesión con formato de email erróneo.

En este apartado, realizaremos un inicio de sesión con formato de email erróneo. Para ello, nos dirigiremos a la esquina superior derecha y pulsaremos en “Iniciar Sesión”. Tras introducir un email con un formato erróneo como, por ejemplo, “pablo” se nos mostrará un

mensaje de error pidiéndonos que introduzcamos un email válido como, por ejemplo, “johndoe@domain.com”. Para ello, necesitamos hacer la siguiente secuencia de acciones en Selenium:

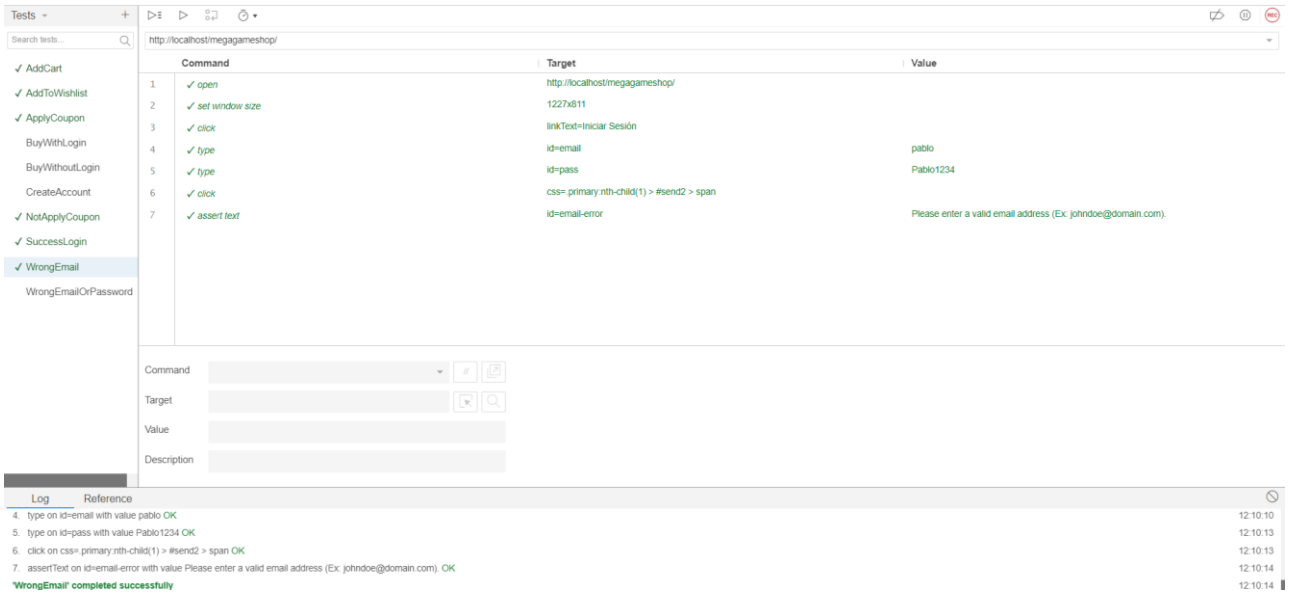

Figura 120. Prueba de inicio de sesión con formato de email erróneo en Selenium.

En el log del programa vemos cómo se cumple toda la secuencia de acciones hasta que se muestra el siguiente mensaje de error. En la siguiente figura observamos cómo se muestra este mensaje:

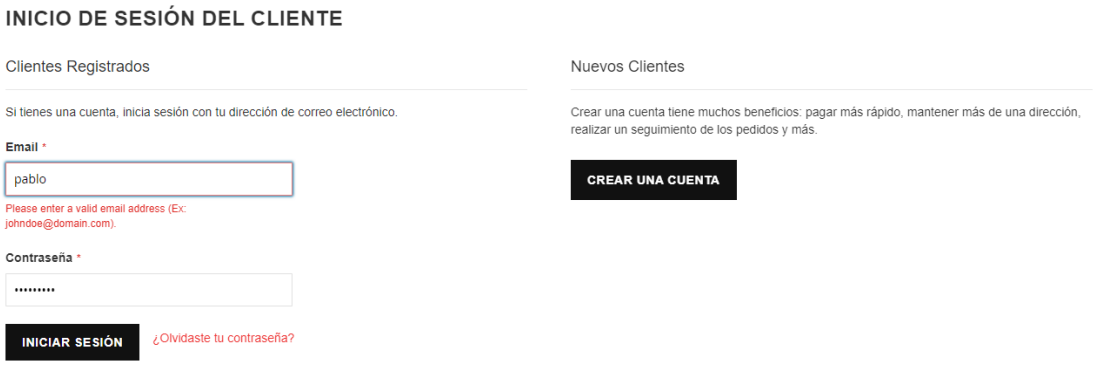

Figura 121. Prueba de inicio de sesión con formato de email erróneo en el frontend de MegaGameShop.

Exportamos el código fuente Java generado por Selenium, en la siguiente figura mostramos el código generado:

```
WrongEmailTest.java
1  // Generated by Selenium IDE
2  import org.junit.Test;
25 public class WrongEmailTest {
26   private WebDriver driver;
27   private Map<String, Object> vars;
28   JavascriptExecutor js;
29   @Before
30   public void setUp() {
31     driver = new ChromeDriver();
32     js = (JavascriptExecutor) driver;
33     vars = new HashMap<String, Object>();
34   }
35   @After
36   public void tearDown() {
37     driver.quit();
38   }
39   @Test
40   public void wrongEmail() {
41     driver.get("http://localhost/megagameshop/");
42     driver.manage().window().setSize(new Dimension(1227, 811));
43     driver.findElement(By.linkText("Iniciar Sesión")).click();
44     driver.findElement(By.id("email")).sendKeys("pablo");
45     driver.findElement(By.id("pass")).sendKeys("Pablo1234");
46     driver.findElement(By.cssSelector(".primary:nth-child(1) > #send2 > span")).click();
47     assertThat(driver.findElement(By.id("email-error")).getText(), is("Please enter a valid email address (Ex: johndoe@domain.com)."));
48   }
49 }
```

Figura 122. Código fuente de la prueba de inicio de sesión con formato de email erróneo en Selenium.

4.7.7. Prueba de inicio de sesión con email o contraseña errónea.

En este apartado, realizaremos un inicio de sesión con email o contraseña errónea. Para ambos casos se mostrará el mismo mensaje de error. Para ello, nos dirigiremos a la esquina superior derecha y pulsaremos en “Iniciar Sesión”. Tras introducir un email que no está registrado como, por ejemplo, “pablolozano60@gmail.com” se nos mostrará un mensaje de error. Para ello, necesitamos hacer la siguiente secuencia de acciones en Selenium:

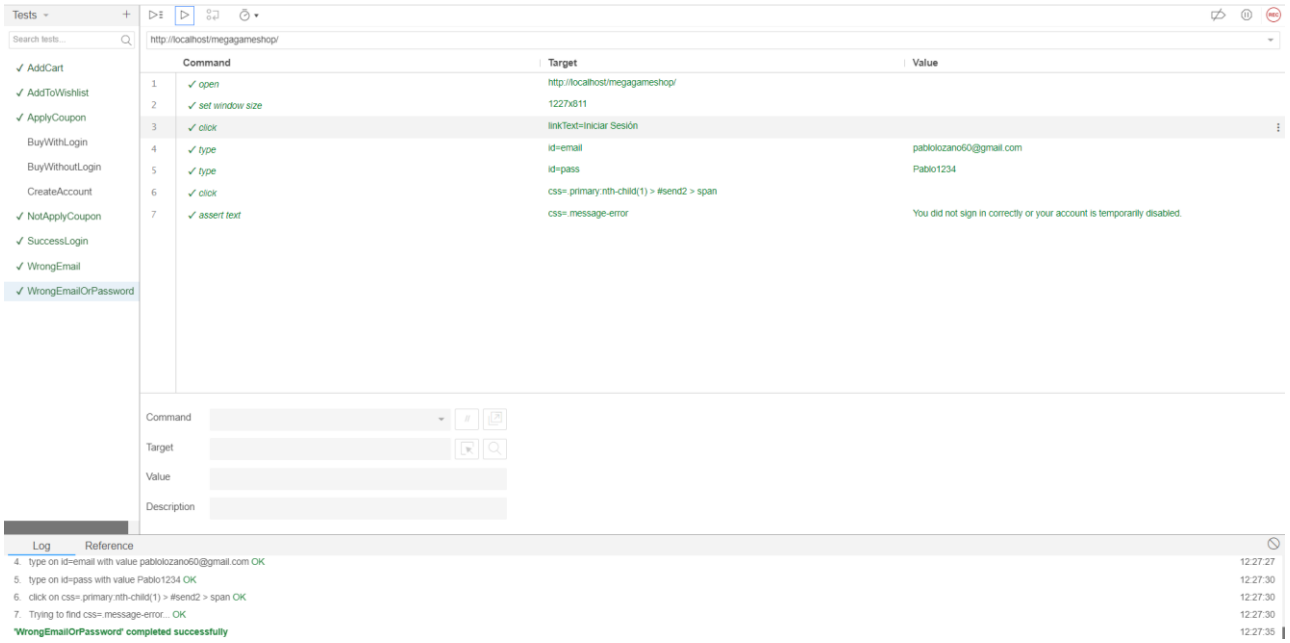

Figura 123. Prueba de inicio de sesión con email o contraseña errónea en Selenium.

En el log del programa vemos cómo se cumple toda la secuencia de acciones hasta que se muestra el siguiente mensaje de error. En la siguiente figura observamos cómo se muestra este mensaje:

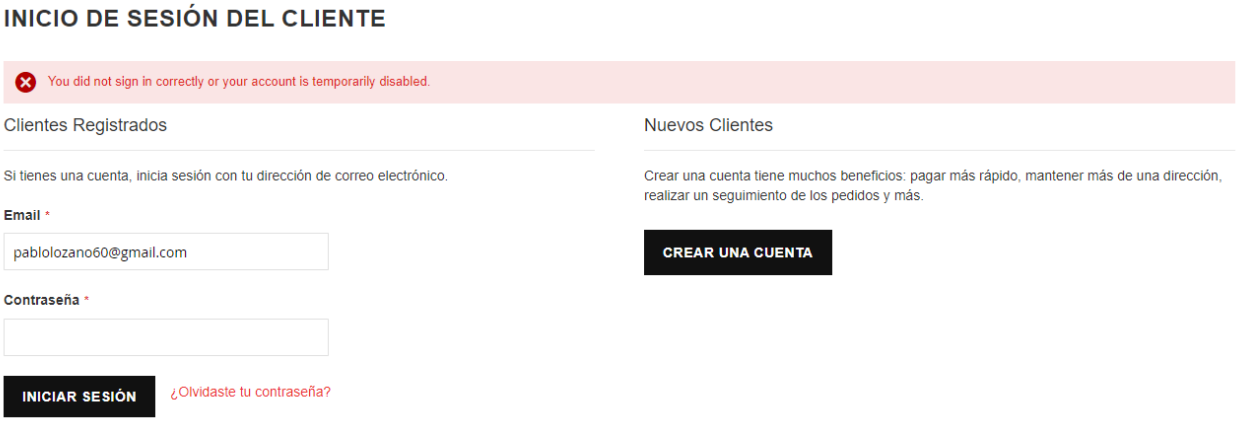

Figura 124. Prueba de inicio de sesión con email o contraseña errónea en el frontend de MegaGameShop.

Exportamos el código fuente Java generado por Selenium, en la siguiente figura mostramos el código generado:

```
WrongEmailOrPasswordTest.java
1  // Generated by Selenium IDE
2  import org.junit.Test;
25 public class WrongEmailOrPasswordTest {
26   private WebDriver driver;
27   private Map<String, Object> vars;
28   JavascriptExecutor js;
29   @Before
30   public void setUp() {
31     driver = new ChromeDriver();
32     js = (JavascriptExecutor) driver;
33     vars = new HashMap<String, Object>();
34   }
35   @After
36   public void tearDown() {
37     driver.quit();
38   }
39   @Test
40   public void wrongEmailOrPassword() {
41     driver.get("http://localhost/megagameshop/");
42     driver.manage().window().setSize(new Dimension(1227, 811));
43     driver.findElement(By.linkText("Iniciar Sesión")).click();
44     driver.findElement(By.id("email")).sendKeys("pablolozano60@gmail.com");
45     driver.findElement(By.id("pass")).sendKeys("Pablo1234");
46     driver.findElement(By.cssSelector(".primary:nth-child(1) > #send2 > span")).click();
47     assertThat(driver.findElement(By.cssSelector(".message-error")).getText(), is("You did not sign in correctly or your account is temporarily disabled."));
48   }
49 }
```

Figura 125. Código fuente de la prueba de inicio de sesión con email o contraseña errónea en Selenium.

4.7.8. Prueba de creación de cuenta de cliente.

En este apartado, realizaremos la creación de cliente. Nos dirigiremos a la esquina superior derecha y pulsaremos en “Crear una cuenta”. Tras rellenar el formulario que se nos muestra, tendremos la cuenta con éxito. Para ello, necesitamos hacer la siguiente secuencia de acciones en Selenium:

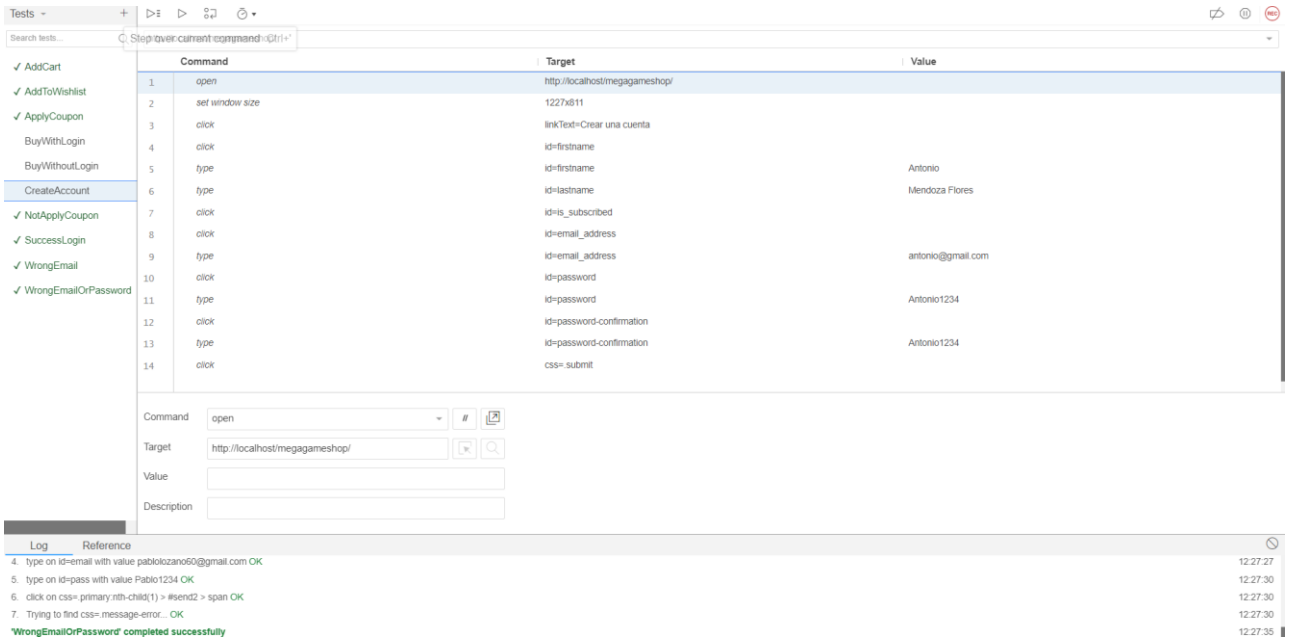

Figura 126. Prueba de creación de cuenta de cliente en Selenium.

En el log del programa vemos cómo se cumple toda la secuencia de acciones hasta que se rellenan todos los campos y se crea una cuenta de cliente. En la siguiente figura observamos el formulario relleno:

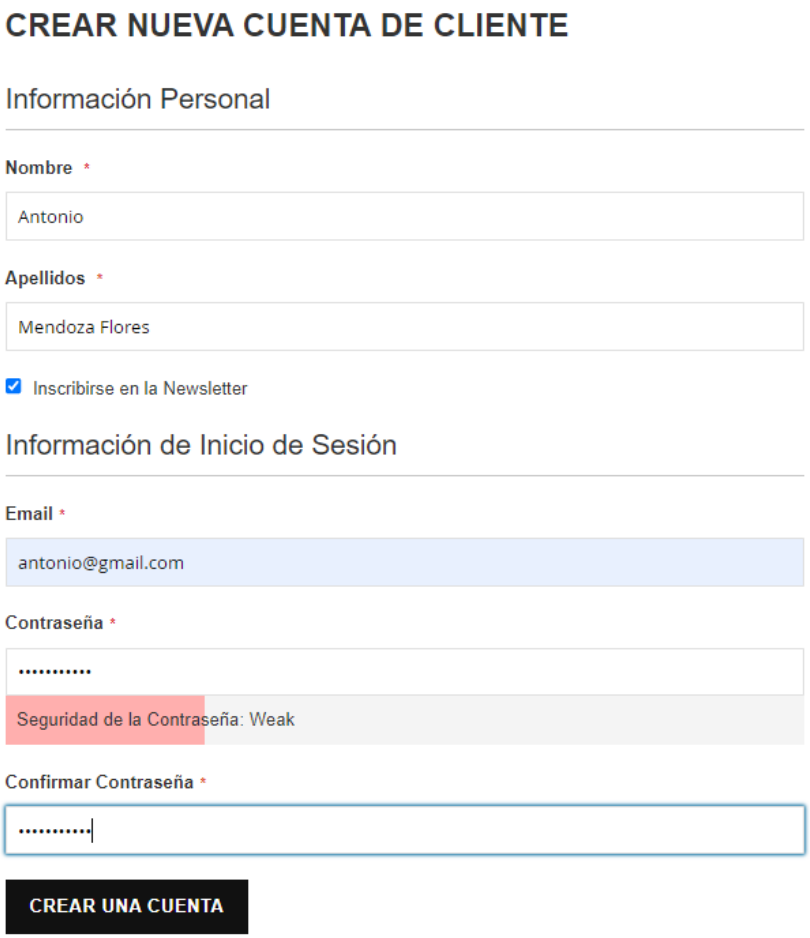

Figura 127. Prueba de creación de cuenta de cliente en el frontend de MegaGameShop.

Exportamos el código fuente Java generado por Selenium, en la siguiente figura mostramos el código generado:

```
CreateAccountTest.java
1  // Generated by Selenium IDE
2  import org.junit.Test;
25 public class CreateAccountTest {
26   private WebDriver driver;
27   private Map<String, Object> vars;
28   JavascriptExecutor js;
29   @Before
30   public void setUp() {
31     driver = new ChromeDriver();
32     js = (JavascriptExecutor) driver;
33     vars = new HashMap<String, Object>();
34   }
35   @After
36   public void tearDown() {
37     driver.quit();
38   }
39   @Test
40   public void createAccount() {
41     driver.get("http://localhost/megagameshop/");
42     driver.manage().window().setSize(new Dimension(1227, 811));
43     driver.findElement(By.linkText("Crear una cuenta")).click();
44     driver.findElement(By.id("firstname")).click();
45     driver.findElement(By.id("firstname")).sendKeys("Antonio");
46     driver.findElement(By.id("lastname")).sendKeys("Mendoza Flores");
47     driver.findElement(By.id("is_subscribed")).click();
48     driver.findElement(By.id("email_address")).click();
49     driver.findElement(By.id("email_address")).sendKeys("antonio@gmail.com");
50     driver.findElement(By.id("password")).click();
51     driver.findElement(By.id("password")).sendKeys("Antonio1234");
52     driver.findElement(By.id("password-confirmation")).click();
53     driver.findElement(By.id("password-confirmation")).sendKeys("Antonio1234");
54     driver.findElement(By.cssSelector(".submit")).click();
55   }
56 }
```

Figura 128. Código fuente de la prueba de creación de cuenta de cliente en Selenium.

4.7.9. Prueba de compra de cliente registrado.

En este apartado, realizaremos una compra de un producto para un cliente que se encuentra registrado en la web. Nos dirigiremos a la esquina superior derecha y pulsaremos en “Iniciar Sesión”. Tras introducir un email y una contraseña correcta, tendremos la sesión iniciada con éxito. A continuación, añadiremos un producto a la cesta de compra y realizaremos el pago del mismo. Para ello, necesitamos hacer la siguiente secuencia de acciones en Selenium:

Tests

Search tests...

AddCart
AddToWishlist
✓ ApplyCoupon
✓ BuyWithLogin
BuyWithoutLogin
CreateAccount
NotApplyCoupon
SuccessLogin
WrongEmail
WrongEmailOrPassword

http://localhost/megagameshop/

	Command	Target	Value
1	✓ open	http://localhost/megagameshop/	
2	✓ set window size	1227x811	
3	✓ click	linkText=Iniciar Sesión	
4	✓ mouse down at	css=.page-wrapper	51,527
5	✓ mouse move at	css=.page-wrapper	51,527
6	✓ mouse up at	css=.page-wrapper	51,527
7	✓ click	css=.page-wrapper	
8	✓ type	id=email	pablolozano50@gmail.com
9	✓ type	id=pass	Pablo1234
10	✓ click	css=.primary:nth-child(1) > #send2 > span	
11	✓ mouse over	css=#ui-id-4 > span:nth-child(2)	
12	✓ click	css=#ui-id-14 > span	
13	✓ click	css=.item:nth-child(6) .product-image-photo	
14	✓ click	css=#product-addtocart-button > span	
15	✓ assert text	css=.message-success > div	You added Ratchet & Clank: Una Dimensión Aparte to your shopping cart.
16	✓ click	css=.showcart	
17	✓ click	id=top-cart-btn-checkout	
18	✓ click	css=.button > span	
19	✓ run script	window.scrollTo(0,0)	
20	✓ click	css=.payment-method:nth-child(5) > .payment-method-title span	
21	✓ click	css=._active > .payment-method-content > .actions-toolbar span	

Command
Target
Value
Description

Log Reference

20. click on css=.payment-method:nth-child(5) > .payment-method-title span OK 18:51:25
21. click on css=._active > .payment-method-content > .actions-toolbar span OK 18:51:26
'BuyWithLogin' completed successfully 18:51:27

Figura 129. Prueba de compra de cliente registrado en Selenium.

En el log del programa vemos cómo se cumple toda la secuencia de acciones hasta que se completa un pedido. En la siguiente figura observamos que el pedido se ha realizado con éxito:

Español

Bienvenido, Pablo! Pablo Lozano Ropa

Buscar en la tienda completa aq

MEGAGAMESHOP

CONSOLAS VIDEOJUEGOS PC GAMING MERCHANDISING ACCESORIOS

THANK YOU FOR YOUR PURCHASE!

Print receipt

Your order number is: 000000011.

We'll email you an order confirmation with details and tracking info.

CONTINUE SHOPPING

Figura 130. Prueba de compra de cliente registrado en el frontend de MegaGameShop.

Tras finalizar el pedido, podremos acceder a la factura de compra para conocer todos

los detalles de nuestra compra:

MEGAGAMESHOP

PEDIDO # 000000011 PENDIENTE

26 de junio de 2022

Nombre del producto	SKU	Precio	Cant	Subtotal
Ratchet & Clank: Una Dimensión Aparte	Ratchet & Clank: Una Dimensión Aparte	40,00 €	Pedidos: 1	40,00 €
			Subtotal	40,00 €
			Envío y Manejo	0,00 €
			Tax	8,40 €
			Totalidad	**48,40 €**

Order Information

Dirección de envío
Pablo Lozano Ropa
Navas, n 13
1D
Lucena, Cordoba, 14900
España
T: 676674860

Método de Envío
Envío Gratuito - Correos

Dirección de facturación
Pablo Lozano Ropa
Navas, n 13
1D
Lucena, Cordoba, 14900
España
T: 676674860

Método de pago
Transferencia Bancaria
Nuestro número de cuenta es :

MegaGameShop
2100 2931 9001 0090 1542
IBAN: ES44 2100 2931 9001 0090 1542
BIC/SWIFT: CAIXESBBXXX

Una vez realizada la transferencia, por favor enviar comprobante a megagameshop@gmail.com

Figura 131. Factura de pedido de compra de cliente registrado de MegaGameShop.

Exportamos el código fuente Java generado por Selenium, en la siguiente figura mostramos el código generado:

```java
BuyWithLoginTest.java
// Generated by Selenium IDE
import org.junit.Test;
public class BuyWithLoginTest {
  private WebDriver driver;
  private Map<String, Object> vars;
  JavascriptExecutor js;
  @Before
  public void setUp() {
    driver = new ChromeDriver();
    js = (JavascriptExecutor) driver;
    vars = new HashMap<String, Object>();
  }
  @After
  public void tearDown() {
    driver.quit();
  }
  @Test
  public void buyWithLogin() {
    driver.get("http://localhost/megagameshop/");
    driver.manage().window().setSize(new Dimension(1227, 811));
    driver.findElement(By.linkText("Iniciar Sesión")).click();
    {
      WebElement element = driver.findElement(By.cssSelector(".page-wrapper"));
      Actions builder = new Actions(driver);
      builder.moveToElement(element).clickAndHold().perform();
    }
    {
      WebElement element = driver.findElement(By.cssSelector(".page-wrapper"));
      Actions builder = new Actions(driver);
      builder.moveToElement(element).perform();
    }
    {
      WebElement element = driver.findElement(By.cssSelector(".page-wrapper"));
      Actions builder = new Actions(driver);
      builder.moveToElement(element).release().perform();
    }
    driver.findElement(By.cssSelector(".page-wrapper")).click();
    driver.findElement(By.id("email")).sendKeys("pablolozano50@gmail.com");
    driver.findElement(By.id("pass")).sendKeys("Pablo1234");
    driver.findElement(By.cssSelector(".primary:nth-child(1) > #send2 > span")).click();
    {
      WebElement element = driver.findElement(By.cssSelector("#ui-id-4 > span:nth-child(2)"));
      Actions builder = new Actions(driver);
      builder.moveToElement(element).perform();
    }
    driver.findElement(By.cssSelector("#ui-id-14 > span")).click();
    driver.findElement(By.cssSelector(".item:nth-child(6) .product-image-photo")).click();
    driver.findElement(By.cssSelector("#product-addtocart-button > span")).click();
    assertThat(driver.findElement(By.cssSelector(".message-success > div")).getText(), is("You added Ratchet & Clank: Una Dimensión Aparte to your shopping cart."));
    driver.findElement(By.cssSelector(".showcart")).click();
    driver.findElement(By.id("top-cart-btn-checkout")).click();
    driver.findElement(By.cssSelector(".button > span")).click();
    js.executeScript("window.scrollTo(0,0)");
    driver.findElement(By.cssSelector(".payment-method:nth-child(5) > .payment-method-title span")).click();
    driver.findElement(By.cssSelector(".\\_active > .payment-method-content > .actions-toolbar span")).click();
  }
}
```

Figura 132. Código fuente de la prueba de compra de cliente registrado en Selenium.

4.7.10. Prueba de compra de cliente invitado.

En este apartado, realizaremos una compra de un producto para un cliente invitado que no se encuentra registrado en la web. La secuencia de pasos a seguir es la misma que la de una compra para un cliente registrado, pero sin tener que realizar el paso de iniciar sesión. Justo antes de realizar el pago, se nos mostrará un formulario similar al de crear una nueva cuenta y, tras finalizar el pago, habremos hecho el pedido con éxito. Para ello, necesitamos hacer la siguiente secuencia de acciones en Selenium:

Tests

Search tests...

- AddCart
- AddToWishlist
- ✓ ApplyCoupon
- ✓ BuyWithLogin
- BuyWithoutLogin
- CreateAccount
- NotApplyCoupon
- SuccessLogin
- WrongEmail
- WrongEmailOrPassword

http://localhost/megagameshop/

	Command	Target	Value
1	open	http://localhost/megagameshop/	
2	set window size	1227x811	
3	mouse over	css=#ui-id-4 > span:nth-child(2)	
4	click	css=#ui-id-14 > span	
5	click	css=.item:nth-child(6) .product-image-photo	
6	click	css=#product-addtocart-button > span	
7	assert text	css=.message-success > div	You added Ratchet & Clank: Una Dimensión Aparte to your shopping cart.
8	click	css=.showcart	
9	click	id=top-cart-btn-checkout	
10	type	id=customer-email	juan23@gmail.com
11	click	name=firstname	
12	type	name=firstname	Juan
13	click	name=lastname	
14	type	name=lastname	Gómez Pérez
15	click	name=street[0]	
16	type	name=street[0]	Avenida Andalucía, 23
17	click	name=city	
18	type	name=city	Jaén
19	click	name=region_id	
20	select	name=region_id	label=Jaen
21	click	name=postcode	
22	type	name=postcode	23009
23	click	name=telephone	
24	type	name=telephone	675791234
25	click	css=.button	
26	run script	window.scrollTo(0,0)	
27	click	css=.payment-method:nth-child(5) > .payment-method-title	

Command: open

Target: http://localhost/megagameshop/

Value:

Description:

Figura 133. Prueba de compra de cliente invitado en Selenium.

En el log del programa vemos cómo se cumple toda la secuencia de acciones hasta que se completa un pedido. Tras realizar el pedido, podemos crear una cuenta con la dirección de correo que utilizamos para realizar el pedido, de tal forma que podremos realizar el seguimiento del mismo. En la siguiente figura observamos que el pedido se ha realizado con éxito:

Español

Bienvenido a MegaGameShop! Iniciar Sesión Crear una cuenta

Buscar en la tienda completa aq

0

MEGAGAMESHOP

CONSOLAS VIDEOJUEGOS PC GAMING MERCHANDISING ACCESORIOS

THANK YOU FOR YOUR PURCHASE!

Your order # is: 000000005.

We'll email you an order confirmation with details and tracking info.

CONTINUE SHOPPING

You can track your order status by creating an account.

Dirección de correo: juan23@gmail.com

CREAR UNA CUENTA

Figura 134. Prueba de compra de cliente invitado en el frontend de MegaGameShop.

Exportamos el código fuente Java generado por Selenium, en la siguiente figura mostramos el código generado:

```
1  // Generated by Selenium IDE
2  import org.junit.Test;
25 public class BuyWithoutLoginTest {
26   private WebDriver driver;
27   private Map<String, Object> vars;
28   JavascriptExecutor js;
29   @Before
30   public void setUp() {
31     driver = new ChromeDriver();
32     js = (JavascriptExecutor) driver;
33     vars = new HashMap<String, Object>();
34   }
35   @After
36   public void tearDown() {
37     driver.quit();
38   }
39   @Test
40   public void buyWithoutLogin() {
41     driver.get("http://localhost/megagameshop/");
42     driver.manage().window().setSize(new Dimension(1227, 811));
43     {
44       WebElement element = driver.findElement(By.cssSelector("#ui-id-4 > span:nth-child(2)"));
45       Actions builder = new Actions(driver);
46       builder.moveToElement(element).perform();
47     }
48     driver.findElement(By.cssSelector("#ui-id-14 > span")).click();
49     driver.findElement(By.cssSelector(".item:nth-child(6) .product-image-photo")).click();
50     driver.findElement(By.cssSelector("#product-addtocart-button > span")).click();
51     assertThat(driver.findElement(By.cssSelector(".message-success > div")).getText(), is("You added Ratchet & Clank: Una Dimensión Aparte to your shopping cart."));
52     driver.findElement(By.cssSelector(".showcart")).click();
53     driver.findElement(By.id("top-cart-btn-checkout")).click();
54     driver.findElement(By.id("customer-email")).sendKeys("juan23@gmail.com");
55     driver.findElement(By.name("firstname")).click();
56     driver.findElement(By.name("firstname")).sendKeys("Juan");
57     driver.findElement(By.name("lastname")).click();
58     driver.findElement(By.name("lastname")).sendKeys("Gómez Pérez");
59     driver.findElement(By.name("street[0]")).click();
60     driver.findElement(By.name("street[0]")).sendKeys("Avenida Andalucía, 23");
61     driver.findElement(By.name("city")).click();
62     driver.findElement(By.name("city")).sendKeys("Jaén");
63     driver.findElement(By.name("region_id")).click();
64     {
65       WebElement dropdown = driver.findElement(By.name("region_id"));
66       dropdown.findElement(By.xpath("//option[. = 'Jaen']")).click();
67     }
68     driver.findElement(By.name("postcode")).click();
69     driver.findElement(By.name("postcode")).sendKeys("23009");
70     driver.findElement(By.name("telephone")).click();
71     driver.findElement(By.name("telephone")).sendKeys("675791234");
72     driver.findElement(By.cssSelector(".button")).click();
73     js.executeScript("window.scrollTo(0,0)");
74     driver.findElement(By.cssSelector(".payment-method:nth-child(5) > .payment-method-title")).click();
75     driver.findElement(By.cssSelector(".payment-method:nth-child(5) > .payment-method-title span")).click();
76     driver.findElement(By.cssSelector(".\\_active > .payment-method-content > .actions-toolbar span")).click();
77   }
78 }
```

Figura 135. Código fuente de la prueba de compra de cliente invitado en Selenium.

CAPÍTULO 5. CONCLUSIÓN.

5.1. Conclusiones.

Para concluir este Trabajo de Fin de Grado, debemos extraer nuestras conclusiones acerca del desarrollo de este proyecto.

Hemos desarrollado con éxito nuestra tienda online MegaGameShop, logrando implementar cada uno de los requisitos marcados antes de comenzar su desarrollo. Conseguimos un sitio web con una gran facilidad de uso haciendo una interfaz intuitiva y adaptándonos al dispositivo que esté utilizando el cliente al ser un sitio web responsive design (Smartphone, ordenador o tablet). Además, hemos implementado dos idiomas para llegar a un mayor número de personas (inglés y español).

Respecto a la instalación del servidor local, me ha otorgado una gran cantidad de conocimientos utilizando la línea de comandos mediante el Shell que trae integrado, además de aprender a realizar acciones que a través de la interfaz de Magento iba a resultar más tedioso.

Debo destacar la realización de la tienda online a través del CMS Magento, ya que es un sistema de gestión de contenidos bastante potente y nos ha permitido modificar el código fuente para poder adaptarnos a los requisitos que teníamos para realizar nuestro sitio web. Además, nos ofrece un gran número de ajustes y configuraciones para cuidar cada detalle de nuestra web, ajustándolo a nuestros requisitos.

Respecto a la utilización de la metodología en cascada, nos ha permitido realizar cada fase por separado, contactando con el tutor del proyecto para realizar una reunión al finalizar cada fase y obtener un feedback que me ayudara a mejorar su desarrollo sin arrastrar posibles errores.

Para finalizar, comentar que este proyecto me ha resultado de gran utilidad para conocer y aprender cómo crear y gestionar una tienda online a través de un Sistema de Gestión de Contenidos tan completo como Magento. Además, me ha servido para afianzar mis conocimientos en base de datos, HTML, CSS o PHP para la utilización de la línea de comandos del servidor local.

5.2. Posibles desarrollos futuros.

Para posibles desarrollos futuros, se podría implementar nuestro sitio web en un servicio de alojamiento web, para llevar nuestro sitio web a Internet y así poder ofrecer nuestros productos a un público real.

Con respecto al sitio web, podríamos incrementar el número de funcionalidades o ampliar los tipos de filtros dentro de las categorías, para ofrecer al cliente una mejor experiencia de usuario.

Como conclusión, considero que la realización de este Trabajo de Fin de Grado puede ayudarme de cara al mundo laboral, ya que nuestro sitio web podría ser ampliado o podría desarrollar otro sitio web para alguna empresa o cliente en particular, adaptándome a sus necesidades, ahora que dispongo de un mayor conocimiento en este ámbito.

BIBLIOGRAFÍA

[1] “Crecimiento del comercio electrónico en España en 2020”, eccommercerentrable.es. Disponible: https://ecommercerentable.es/ecommerce-espana-2021/ [Accedido por última vez: 27 de junio de 2022]

[2] “Comercio electrónico en España”, eccommercerentrable.es. Disponible: https://ecommercerentable.es/ecommerce-espana-2021/ [Accedido por última vez: 27 de junio de 2022]

[3] “Sector de los videojuegos en España”, artline.es. Disponible: https://www.artline.es/el-sector-de-los-videojuegos-en-espana-caracteristicas-datos-y-estrategias-de-marketing/ [Accedido por última vez: 27 de junio de 2022]

[4] “Descripción de CMS”, webempresa.com. Disponible: https://www.webempresa.com/blog/que-es-cms-los-mejores-gestores-de-contenido.html [Accedido por última vez: 27 de junio de 2022]

[5] “Ventajas e inconvenientes de los CMS”, tactic-center.com. Disponible: https://tactic-center.com/desarrollo-web/ventajas-desventajas-usar-cms-desarrollar-sitio-web/?locale=es [Accedido por última vez: 27 de junio de 2022]

[6] “CMS más utilizados para el comercio electrónico”, wappalyzer.com. Disponible: https://www.wappalyzer.com/technologies/ecommerce [Accedido por última vez: 27 de junio de 2022]

[7] ”CMS WooCommerce, ventajas y desventajas”, webempresa.com. Disponible: https://www.webempresa.com/blog/ventajas-desventajas-woocommerce.html [Accedido por última vez: 27 de junio de 2022]

[8] “CMS Shopify, ventajas y desventajas”, marketing4ecommerce.net. Disponible: https://marketing4ecommerce.net/como-funciona-shopify-ventajas-desventajas-como-cms-para-tu-ecommerce/ [Accedido por última vez: 27 de junio de 2022]

[9] “CMS Squarespace Commerce, ventajas y desventajas”, coworkings.co. Disponible: https://coworkings.co/squarespace-ventajas-desventajas-reviews-y-precios/ [Accedido por última vez: 27 de junio de 2022]

[10] “CMS Wix eCommerce, ventajas y desventajas”, proun.es. Disponible: https://www.proun.es/blog/ventajas-desventajas-wix/ [Accedido por última vez: 27 de junio de 2022]

[11] “CMS PrestaShop, ventajas y desventajas”, infranetworking.com. Disponible: https://blog.infranetworking.com/prestashop-ventajas-desventajas/ [Accedido por última vez: 27 de junio de 2022]

[12] “CMS Magento, ventajas y desventajas”, packlink.es. Disponible: https://pro.packlink.es/becommerce/ventajas-y-desventajas-de-crear-una-tienda-online-con-magento/ [Accedido por última vez: 27 de junio de 2022]

[13] “Servicios de alojamiento web”, marketing4ecommerce.net. Disponible: https://marketing4ecommerce.net/top-mejores-herramientas-de-hosting/ [Accedido por última vez: 27 de junio de 2022]

[14] “Cuánto cobra un programador web”, zaask.es. Disponible: https://www.zaask.es/cuanto-cuesta/programador-php [Accedido por última vez: 27 de junio de 2022]

[15] “IVA en España”, n26.com. Disponible: https://www.n26.com/es-es/blog/tipos-iva [Accedido por última vez: 27 de junio de 2022]