PONTIFICIA UNIVERSIDAD CATOLICA DEL ECUADOR

FACULTAD DE INGENIERIA

ESCUELA DE SISTEMAS



Trabajo de Titulación

Tema: DESARROLLO DE UNA APLICACIÓN WEB E-COMMERCE CASO DE ESTUDIO: SAINT FOUNDER.

AUTOR:

FREDDY ALEXANDER CHAVEZ TENECELA

QUITO DM, 2022

Dedicatoria:

A mis padres Freddy y Rosa quienes me enseñaron que nunca debo rendirme antes las adversidades que da la vida, creyeron siempre en mí y supieron aconsejarme sabiamente.

A mi tía Miriam y a mi abuela Hirma por acogerme en su hogar durante mis tiempos de estudio, por brindarme su simpatía y apoyo.

Agradecimientos:

Agradezco Dios por darme la vida, a mis padres por darme amor y educación, a mi familia por siempre brindarme su apoyo, a mis profesores por todo el conocimiento que me supieron transmitir, a mis amigos por alegrar aquellos días de estudio.

Contenido

Capítulo 1: Introducción. 9
1.1 Caso de estudio 9
1.1 Justificación 9
1.2 Planteamiento del problema. 10
1.3 Objetivos 11
1.3.1 Objetivo Principal 11
1.4 Antecedentes 11
1.5 Alcance 11
Capítulo 2 Marco teórico. 12
2.1 Aplicaciones web 12
2.2 E-commerce 12
2.3 ¿Cómo se puede crear una aplicación web de tipo e-commerce? 14
2.4 Tecnología seleccionada. 14
2.4.1 Lenguajes de programación. 14
2.4.2 Frameworks de PHP 16
2.4.3 Bases de datos: 18
2.4.4 Herramientas de desarrollo 18
2.4.5 Herramientas de producción 19
2.4.6 Pasarelas de pago. 19
2.4.7 Resumen de la tecnología seleccionada 19
2.5 Selección de metodología: 20
2.5.1 Metodologías Tradicionales. 20
2.5.2 Metodologías Agiles 21
Capítulo 3: Desarrollo de la aplicación web. 23
3.1 Establecer requerimientos 23
3.2 Diseño de la base de datos 25
3.2 Configuración de entorno e inicialización de proyecto: 25
3.2. Administrar Usuarios y roles 26
3.3 Administrar Categorías 33
3.4 Administrar Productos 37
3.5 Implementar carrito de compras 44
3.6 Administrar Ordenes 47
3.7 Implementar pasarela de pago 53
3.8 Administrar direcciones de envió. 57
3.9 Administrar servicios 63

Capítulo 4: Subir a producción. 68
Capítulo 5: Conclusiones y Recomendaciones 72
Conclusiones. 72
Recomendaciones. 73
Bibliografía 74
Anexo 1: 75

Lista de figuras

Figura 1. Inicialización del tablero Kanban (Chávez, 2022) 24
Figura 2. Diseño de la base de datos (Chávez, 2022) 25
Figura 3. Creación del proyecto (Chávez, 2022) 26
Figura 4. Tablero Kanban-Diseño de administrar usuarios y roles (Chávez, 2022) 26
Figura 5. Diagrama de casos de uso de administración de usuarios y roles (Chávez, 2022) 27
Figura 6. Tablero Kanban – Desarrollo administración de usuarios y roles (Chávez, 2022) 27
Figura 7. Tabla para administrar usuarios (Chávez, 2022) 28
Figura 8. Formulario para crear nuevo usuario (Chávez, 2022) 29
Figura 9. Formulario para editar un usuario (Chávez, 2022) 29
Figura 10. Confirmación para editar un usuario (Chávez, 2022) 30
Figura 11. Formulario para iniciar sesión de usuario (Chávez, 2022) 30
Figura 12. Formulario para registrar un nuevo usuario (Chávez, 2022) 31
Figura 13. Tablero Kanban –Pruebas de administración de usuarios y roles (Chávez, 2022) 32
Figura 14. Tablero Kanban -Diseño administrar categorías (Chávez, 2022) 33
Figura 15. Diseño administrar categorías (Chávez, 2022) 33
Figura 16. Tablero Kanban -Desarrollo administrar categorías (Chávez, 2022) 34
Figura 17. Tabla para administrar categorías (Chávez, 2022) 34
Figura 18. Formulario para crear una categoría (Chávez, 2022) 35
Figura 19. Formulario para editar una categoría (Chávez, 2022) 36
Figura 20. Confirmación para eliminar una categoría (Chávez, 2022) 36
Figura 21. Tablero Kanban – Pruebas de administración de categorías (Chávez, 2022) 37
Figura 22. Tablero Kanban – Diseño de administrar productos (Chávez, 2022) 37
Figura 23. Diagrama de casos de uso de administrar productos (Chávez, 2022) 38
Figura 24. Tablero Kanban – Desarrollo de administrar productos (Chávez, 2022) 38
Figura 25. Tabla de administración de productos (Chávez, 2022) 39
Figura 26. Formulario de creación de un nuevo producto (Chávez, 2022) 40
Figura 27. Formulario para editar un producto (Chávez, 2022) 40

Figura 28. Formulario para agregar imágenes a un producto (Chávez, 2022) 41
Figura 29. Confirmación para eliminar un producto (Chávez, 2022) 41
Figura 30. Vista de las categorías y productos (Chávez, 2022) 42
Figura 31. Vista de productos filtrado por categoría (Chávez, 2022) 42
Figura 32. Vista de un producto con sus características (Chávez, 2022) 43
Figura 33. Tablero Kanban – Pruebas de administración de productos (Chávez, 2022) 43
Figura 35. Diagrama de casos de uso de implementar carrito de compras (Chávez, 2022) 45
Figura 36. Tablero Kanban – Desarrollo de implementar carrito de compras (Chávez, 2022) 45
Figura 37. Carrito de compras pequeño con productos dentro de este (Chávez, 2022) 46
Figura 38. Carrito de compras con productos dentro de este (Chávez, 2022) 46
Figura 39. Tablero Kanban – Pruebas de implementar carrito de compras (Chávez, 2022) 47
Figura 40. Tablero Kanban – Diseño de administrar órdenes (Chávez, 2022) 47
Figura 41. Diagrama de casos de uso de administrar órdenes (Chávez, 2022) 48
Figura 42. Tablero Kanban – Desarrollo de administrar órdenes (Chávez, 2022) 48
Figura 43 Generar Ordenes con retiro en tienda (Chávez, 2022) 49
Figura 44. Generar Ordenes con envío a domicilio (Chávez, 2022) 49
Figura 45. Visualización de orden generada (Chávez, 2022) 50
Figura 46. Visualización de orden generada con estado cambiado (Chávez, 2022) 50
Figura 47. Recordatorio de parte de la aplicación de que una orden no ha sido pagada (Chávez, 2022) 51
Figura 48. Visualización de la ordenes que ha generado el usuario (Chávez, 2022) 51
Figura 49. Visualización de las órdenes que se han generado en la aplicación (Chávez, 2022) 52
Figura 50. Visualización de una orden y formulario para cambiar el estado de la orden (Chávez, 2022) 52
Figura 51. Tablero Kanban – Pruebas administrar órdenes (Chávez, 2022) 53
Figura 52. Tablero Kanban – Diseño implementar pasarela de pago (Chávez, 2022) 53
Figura 53. Diagrama de casos de uso de implementar pasarela de pago (Chávez, 2022) 54
Figura 54. Tablero Kanban – Desarrollo implementar pasarela de pago (Chávez, 2022) 54
Figura 55. Implementación del botón de PayPal (Chávez, 2022) 55
Figura 56. Funcionamiento del botón de PayPal (Chávez, 2022) 55
Figura 57. Continuación del funcionamiento del botón de PayPal (Chávez, 2022) 56
Figura 58. Vista de orden tras confirmación del botón de PayPal (Chávez, 2022) 56
Figura 59. Tablero Kanban – Pruebas implementar pasarela de pago (Chávez, 2022) 57
Figura 60. Tablero Kanban – Diseño administrar ordenes de envío (Chávez, 2022) 57
Figura 61. Diagrama de casos de uso de administrar direcciones de envío (Chávez, 2022) 58
Figura 62 Tablero Kanban – Desarrollo administrar direcciones de envío (Chávez, 2022) 58

Figura 63. Vista de administración de provincias (Chávez, 2022) 59
Figura 64. Confirmación de eliminar provincia (Chávez, 2022) 60
Figura 65. Vista para edición de provincias (Chávez, 2022) 60
Figura 66. Vista para agregar nueva ciudad (Chávez, 2022) 61
Figura 67. Vista para editar ciudad (Chávez, 2022) 61
Figura 68. Confirmación para eliminar una ciudad (Chávez, 2022) 62
Figura 69. Pruebas Administrar Direcciones de envío (Chávez, 2022) 62
Figura 70. Tablero Kanban – Diseño administrar servicios (Chávez, 2022) 63
Figura 71. Diagrama de casos de uso de administrar servicios (Chávez, 2022) 64
Figura 72. Tablero Kanban – Desarrollo administrar servicios (Chávez, 2022) 64
Figura 73. Tabla de administrar servicios (Chávez, 2022) 65
Figura 74. Formulario para crear un servicio nuevo (Chávez, 2022) 65
Figura 75. Formulario para editar servicio y agregar imágenes (Chávez, 2022) 66
Figura 76. Confirmación para eliminar un servicio (Chávez, 2022) 67
Figura 77- Vista de usuario para servicios (Chávez, 2022) 67
Figura 78. Tablero Kanban – Pruebas administrar servicios (Chávez, 2022) 68
Figura 79. Configuración de servidor de Laravel Forge (Chávez, 2022) 69
Figura 80. Proceso de creación de un servidor con Laravel Forge (Chávez, 2022) 70
Figura 81. Servidor creando con Laravel Forge (Chávez, 2022) 70
Figura 82. Opciones de servidor (Chávez, 2022) 71
Figura 83. Instalación de un proyecto desde Github (Chávez, 2022) 71
Figura 84. Sitio desplegado en un servidor (Chávez, 2022) 72

Lista de tablas

Tabla 1. Comparativa entre Frameworks de Php (Chávez, 2022) 17
Tabla 2. Resumen de la tecnología seleccionada (Chávez, 2022) 20
Tabla 3. Casos de prueba de administración de usuarios (Chávez, 2022) 32
Tabla 4. Casos de prueba de administración de categorías (Chávez, 2022) 37
Tabla 5. Casos de prueba de administrar productos (Chávez, 2022). 44
Tabla 6. Casos de prueba de implementar carrito de compras (Chávez, 2022) 47
Tabla 7. Casos de prueba de administrar órdenes (Chávez, 2022) 53
Tabla 8. Casos de prueba implementar pasarela de pago (Chávez, 2022) 57
Tabla 9. Casos de prueba de administrar direcciones de envío (Chávez, 2022) 63
Tabla 10. Casos de prueba de administrar servicios (Chávez, 2022) 68
Tabla 11. Tabla roles del diccionario de datos (Chávez, 2022). 75

Tabla 12. Tabla rol_usuairo del diccionario de datos (Chávez, 2022). 75
Tabla 13. Tabla Usuarios del diccionario de datos (Chávez, 2022). 75
Tabla 14. Tabla Ordenes del diccionario de datos (Chávez, 2022). 76
Tabla 15. Tabla provincias del diccionario de datos (Chávez, 2022). 76
Tabla 16. Tabla ciudades del diccionario de datos (Chávez, 2022) 76
Tabla 17. Tabla de categorías del diccionario de datos (Chávez, 2022). 76
Tabla 18. Tabla de productos del diccionario de datos (Chávez, 2022). 77
Tabla 19. Tabla de servicios del diccionario de datos (Chávez, 2022). 77
Tabla 20. Tabla de imágenes del diccionario de datos (Chávez, 2022). 77

Capítulo 1: Introducción.

1.1 Caso de estudio

La empresa Saint Founder en la actualidad vende y ofrece servicios de: maquila de cerveza artesanal, alquiler de máquinas dispensadoras de cerveza, estilos personalizados, asesoría y consultoría, diseño de recetas, cursos y tours cerveceros.

Antes de comenzar a realizar actividades comerciales, se contempló durante cinco años la posibilidad de emprender un negocio familiar de elaboración de cerveza artesanal, por lo que previamente fue necesario la implementación de la fábrica de cerveza y la capacitación de sus empleados.

Cuando el emprendimiento comenzó a surgir y establecerse como empresa se procedió a la producción, empaquetado y distribución de la cerveza, donde surgió el reto de lograr vender sus productos a la mayor cantidad de clientes posibles, por lo que optó por promocionar los productos mediante redes sociales, ofreciendo entregas a domicilio con pagos al contado o transferencias bancarias.

"El emprendimiento en Ecuador está mayormente orientado hacia actividades de comercio y a productos y servicios vendidos en el mercado doméstico" (Zamora, 2017), dentro de este contexto las personas tienden a emprender en negocios los cuales ellos tienen conocimientos previos del tema como es el caso de la empresa Saint Founder,

Conforme se fue desarrollando esta empresa comenzó a crecer y por lo tanto a ofrecer una mayor variedad en sus productos y también comenzó a ofrecer servicios relacionados con las actividades que realiza, su clientela ha ido aumentado de a poco, conforme pasa el tiempo gracias a la publicidad por redes sociales y a las personas que los recomiendan.

1.1 Justificación

El desarrollo de las aplicaciones web de e-commerce dentro del Ecuador han venido aumentando con el paso de los años, pues no es ninguna novedad que con el auge de la tecnología los consumidores busquen maneras de optimizar la forma en la que realizan compras de productos o servicios. Según UEES (2017) a pesar de que los ecuatorianos usan el internet para actividades rutinarias como la obtención de información o comunicación, se perfila un futuro prometedor para el comercio electrónico.

En la actualidad con “la llegada del Covid-19 ha traído como consecuencia un incremento en la frecuencia de compra online” (Dakduk & Ricardo, 2020), por lo que se puede decir que en los tiempos actuales el comercio electrónico es más rentable que nunca.

La empresa Saint Founder lleva más de un año desde que inició sus actividades comerciales con la venta en línea de cerveza artesanal, entre otros productos a través de medios digitales de difusión como Facebook e Instagram. Con el presente proyecto surge la oportunidad automatizar los procesos de compra y venta de los productos de que ofrece la empresa, además de informar a los clientes de los servicios que oferta.

El proyecto permitirá a la empresa publicar y administrar sus productos y/o servicios y a su vez los posibles clientes sean capaces de ver las publicaciones, ponerse en contacto y/o realizar pagos por un producto en el cual se encuentren interesados.

1.2 Planteamiento del problema.

La empresa Saint Founder ofrece productos y servicios relacionados a la producción y distribución de cerveza artesanal. Estos productos y servicios se han venido publicitando en redes sociales desde que la empresa se creó, y estas le han servido para promocionarse y crecer, pero conforme comenzó a aumentar su catálogo de productos y servicios, el promocionar solo a través de las redes sociales dejó de ser tan viable ya que se toparon con las dificultades de que: no se puede mostrar todo su inventario de productos y servicios al cliente de una manera dinámica, debido a que este solo puede visualizar la publicaciones desde la más actual hasta la más antigua, inhabilitando la posibilidad de navegar libremente entre publicaciones, además de que en las redes sociales no se puede realizar una venta como tal ya que estas solo sirven para promocionar el producto y ponerse en contacto.

Por lo que se genera un limitante al momento de tratar de acentuarse de manera correcta en el mercado de las ventas en línea, debido a las características que ya fueron mencionadas, por lo que se ha previsto que una aplicación web de tipo e-commerce les ayudará a solventar estas necesidades, de una manera que sea óptima el poder gestionar las compras y ventas entre la empresa Saint Founder y sus posibles clientes.

1.3 Objetivos

1.3.1 Objetivo Principal

Desarrollar una aplicación web de tipo e-commerce que automatice los procesos de ventas y gestión de productos y servicios de la empresa Saint Founder.

1.3.2 Objetivos Específicos

1. Analizar los requerimientos de la empresa para llevar una correcta operación de su tienda en línea, la administración de los productos y servicios que ofrece, además de las posibles soluciones que satisfagan la problemática del proyecto.
2. Desarrollar el modelo de la base de datos, la gestión de roles y permisos, las tareas, los procesos y la parte lógica del proyecto.
3. Diseñar la interfaz de usuario buscando siempre la optimización del código, la fluidez de la aplicación y la experiencia de usuario.
4. Implementar la aplicación web con todas sus funcionalidades comprobadas.

1.4 Antecedentes

1.5 Alcance

Este trabajo tiene como alcance el desarrollo e implementación de una aplicación web tipo e-commerce para la empresa Saint Founder con la finalidad de automatizar los siguientes procesos:

- Creación y registro de usuarios: Se podrá crear usuarios y los posibles nuevos usuarios que entren a la página podrán registrarse para realizar sus compras y contactar con los vendedores.
- Gestión de roles y permisos: Cada usuario que exista en la página tendrá asignado un rol ya sea de administrador o de cliente, dependiendo de los roles tendrán sus respectivas funcionalidades.
- Gestión de los productos y servicios ofrecidos: Los administradores de la página podrán gestionar los productos y servicios que ofrecen, la cantidad y el precio de estos.
- Carrito de compras y pagos: Los clientes serán capaces de agregar los productos que estén interesados en adquirir dentro de un carro de compras para después proceder a realizar los pagos.
- Gestión de ordenes: Los administradores de la página serán capaces de llevar un control de las ordenes generadas por la aplicación.

Capítulo 2 Marco teórico.

2.1 Aplicaciones web

Las aplicaciones web aparecieron con lo auge del internet, aunque en un principio eran estáticas y no muy llamativas visualmente, representaron un antes y un después dentro de industria del software. Se puede decir que una aplicación web no es sino un software alojado dentro de un servidor para el cual es necesario el uso un navegador para poder interactuar con el mismo.

El uso de aplicaciones web se ha ido extendiendo de manera incremental a lo largo de los años gracias a su intrínseca versatilidad al no necesitar de un proceso previo de instalación dentro del dispositivo con el cual se está accediendo.

En la actualidad se pueden dividir las aplicaciones web en estáticas y dinámicas, siendo estas últimas las más demandadas. El desarrollo de aplicaciones web dinámicas es mucho más complejo que las estáticas, ya que requiere de procesos que controlen el dinamismo de la página y administren las bases de datos.

El desarrollo de aplicaciones web dinámicas suele dividirse en dos partes: "*front-end*" y "*back-end*", siendo el "*front-end*" el componente que interactúa con el usuario final de la misma, y el "*back-end*" el componente que gestiona todos los procesos lógicos que existe por detrás de la aplicación, junto con el manejo de las bases de datos.

2.2 E-commerce

"Internet se ha convertido en un insumo intangible de toda estrategia para llegar al cliente" (Barrientos Felipa, 2017) y no es para menos, en la actualidad el internet y las aplicaciones web están presentes en casi todo. Una de las formas más viables que tienen las empresas y negocios de presentarse hacia el mundo es mediante una aplicación web, pero cuando una aplicación web va más allá de los tradicional y busca funcionar de manera similar a una tienda en línea, es cuando se la comienza a considerar como un e-commerce o comercio electrónico.

La demanda de este tipo de aplicaciones ha ido creciendo de manera exponencial en los últimos años. Según Rojas (2017) en la última década se ha generado un importante crecimiento del comercio electrónico sustentado principalmente por el continuo aumento del número de usuarios de internet y el crecimiento constante de usuarios que realizan compras

en línea, por lo que se espera que a futuro el principal canal de ventas de las compañías sea a través de los medios digitales.

Entonces se puede decir que desarrollar e implementar un e-commerce "se ha convertido en una ventaja debido a que las empresas nacionales e internacionales puede ofrecer bienes y servicios a los posibles clientes con mayor facilidad y accesibilidad" (Carrión Gonzáles, 2020).

Según Zambrano, Castellanos y Miranda (2021) la cultura ecuatoriana está acostumbrada a realizar compras de manera presencial debido a que percibe un ambiente de seguridad, pero debido a la pandemia provocada por el covid-19 muchas personas han comenzado a realizar sus compras online, y a partir de esto los clientes a comenzado a tener una perspectiva positiva hacia el comercio digital.

Por lo que para poder desarrollar este tipo de aplicación web se han creado varios tipos de comercios electrónicos los cuales se ajustan a los distintos modelos de negocio y requerimientos de las empresas o negocios, siendo estos:

Business to Busines(B2B): Traducido como de negocio a negocio, este tipo de comercio electrónico está pensado para que empresas puedan ofrecer sus productos o servicios a otras empresas, quien gestiona esta aplicación web puede ser la empresa que ofrece los productos o servicios, u otra empresa que quiera generar ganancias ofreciendo su plataforma y actuando de manera de intermediario.

Business to Customer(B2C): Traducido como de empresa a consumidor, este tipo de comercio electrónico busca que las empresas puedan ofrecer sus productos o servicios a sus consumidores, aparte de que desea encontrar nuevos posibles clientes que se sientan atraídos por la empresa.

Consumer to Consumer(C2C): Traducido como de consumidor a consumidor, este es un tipo de comercio electrónico donde la empresa que lo gestiona tiene como intención el de realizar un papel intermediario entre un vendedor de productos y servicios, con su posible comprador, generando ganancias por los servicios ofrecidos como intermediario.

Consumer to Business(C2B): Traducido como de consumidor a negocio, este funciona de manera inversa del modelo de tipo "*Business to consumer*", siendo en este caso los consumidores quienes pueden ofrecer sus servicios o productos a los negocios.

Business to Administration(B2A): Traducido como de negocio a administración, este tipo de comercio electrónico busca cubrir las relaciones entre las empresas y la administración que es necesaria dependiendo de la zona geográfica, como pueden ser pago de tasas, cálculos de impuestos entre otros.

Consumer to Administration(C2A): Traducido como de consumidor a administración, funciona de manera similar al comercio electrónico de tipo "*Business to administration*" con la diferencia de que está enfocado al consumidor y no a empresas.

2.3 ¿Cómo se puede crear una aplicación web de tipo e-commerce?

Crear una aplicación web de tipo e-commerce es bastante similar a la creación de aplicaciones web tradicionales, pero es necesario tener en cuenta ciertas consideraciones que por lo general se suelen presentar en este tipo de aplicaciones.

Por lo que antes de comenzar a desarrollar es necesario primero realizar un análisis de requerimientos, este análisis se debe realizar en conjunto con la lógica del negocio al cual se le va a implementar el comercio electrónico.

Una vez realizado lo anterior es necesario realizar una selección adecuada de las herramientas que se van a usar en el desarrollo de proyecto, como son: los lenguajes de programación, los Frameworks, la base de datos, los servidores web, pasarelas de pago, entre otros.

Se debe escoger también una metodología de desarrollo para que el proyecto tenga estándares donde seguir y este se desarrolle con el tiempo, algunas metodologías suelen tener su propio enfoque de desarrollo, es decir cada metodología tiene su propia forma de realizar la cosas. El equipo de desarrollo puede escoger la metodología que mejor se adapte a su estilo de trabajo y experiencia.

2.4 Tecnología seleccionada.

2.4.1 Lenguajes de programación.

Desarrollar una aplicación web se puede conseguir de muchas maneras en la actualidad gracias a la existencia de un sin número de lenguajes de programación que son capaces de crear aplicaciones web.

A pesar de que la mayoría de los lenguajes de programación son capaces de crear aplicaciones web, solo algunos están orientados o son recomendados para crear aplicaciones web, como es el caso de PHP y JavaScript.

PHP: Tiene su nombre de "*Hypertext Preprocessor*" que traducido seria "*Preprocesado de Hipertexto*", es un lenguaje interpretado que nació en el año 1995, está orientado a la programación web permitiendo crear paginas dinámicas con ayuda de scripts que se ejecutan del lado del servidor.

Es de código abierto, tiene una gran agilidad para trabajar con grandes cantidades de datos, permite combinar HTML con código de PHP renderizando desde el lado del servidor las vistas para el usuario final, tiene tanto ventajas como desventajas de las cuales vale la pena mencionar:

Ventajas:

- Dispone de sintaxis bastante limpia y es fácil de aprender.
- Trabaja bastante bien con diferentes bases de datos.
- Es un lenguaje altamente escalable.

Desventajas:

- Necesita de un servidor para su ejecución.
- Se necesitan de conocimientos avanzados para configurar las brechas de seguridad que puedan existir.

JavaScript: Este lenguaje de programación vio la luz en 1995 a los inicios de la web, teniendo como objetivo agregarles dinamismo a aquellas aplicaciones web que para ese tiempo eran estáticas en su mayoría. En un comienzo solo estaba enfocado en los navegadores y cada navegador tenía su variante de este lenguaje, por lo que crearon estándares para que se pueda trabajar con un mismo código en varios navegadores, a estos estándares se les denominó ECMAScript, teniendo varias versiones siendo la última ECMAScript6.

A pesar de que JavaScript estaba originalmente pensado para ser ejecutado dentro de navegadores, es decir del lado de cliente, en la actualidad también es posible ejecutar este lenguaje del lado del servidor con la ayuda del entorno de ejecución node.js.

Según el portal web stackoverflow (2021) JavaScript se ha posicionado como el lenguaje más usado del mundo por los últimos 9 años. Debido a que permite realizar un gran número de cosas, entre las ventajas y desventajas de este lenguaje cabe destacar:

Ventajas:

- Es fácil de aprender.
- Tienen una sintaxis sencilla.
- Es compatible con la mayoría de los navegadores.
- Tiene una gran versatilidad ya que puede ser usado para desarrollar tanto front-end como back-end.
- Permite el desarrollo multiplataforma, ya que puede crear aplicaciones que se ejecuten en varios sistemas operativos.

Desventajas:

- No tiene muchas opciones para gráficos en 3D.
- Puede llegar a ser inseguro si no se usan lineamientos correctos al momento del desarrollo front-end.
- La falta de tipado estricto puede convertirse en un problema al momento de desarrollar aplicaciones demasiado grandes.

Este proyecto será desarrollado principalmente con el lenguaje PHP, pero se usará también JavaScript para darle un mayor dinamismo a la aplicación.

2.4.2 Frameworks de PHP

Un “framework” se puede traducir como “marco de trabajo”, es un entorno de desarrollo el cual puede incluir bibliotecas que añaden todo tipo de funciones necesarias para el desarrollo al cual este enfocado el framework, permitiendo ahorra tiempo de desarrollo al programador que de lo contrario hubiese tenido que desarrollar todas estas funcionalidades desde cero.

El lenguaje PHP tiene un gran número de Frameworks disponibles, de los cuales los desarrolladores pueden escoger el que mejor se adapte a sus necesidades, a continuación, se mencionan algunos de los Frameworks más usados de PHP.

CodeIgniter: Es un Framework de arquitectura Modelo Vista Controlador el cual utiliza componentes para su desarrollo, a pesar de ser de código abierto es mantenido por la empresa EllisLab.

Symfony: Es un Frameworks bastante flexible el cual busca simplificar lo máximo posible el desarrollo de aplicaciones web, utiliza la arquitectura Modelo Vista Controlador, es de código abierto y es mantenido por la comunidad.

CakePHP: Fue de los primeros Frameworks de arquitectura Modelo Vista Controlador que aparecieron para PHP en el año 2005, es de código abierto, fue creado y es mantenido por la empresa Cake Software Foundation.

Laravel: Es un Framework de código abierto mantenido mayormente por la comunidad, el cual asegura ofrecer un espacio elegante de trabajo, usa la arquitectura Modelo Vista Controlador, está construido sobre un gran número de dependencias.

De manera general se pueden ver las características de los Frameworks mencionados.

CodeIgniter: Es sencillo de aprender. Es ligero y tiene un buen rendimiento. Tiene una buena seguridad.	**Symfony:** Componentes modulares flexibles. Excelente depuración. Es compatible con una gran cantidad de bases de datos. Incluye Doctrine como su ORM (mapeo de objeto-relacional). por defecto. Es bastante popular.
CakePHP: Son necesaria pocas configuraciones. Tiene un ORM propio. Incluye características de seguridad. Tiene una gran cantidad de recursos y plugins disponibles.	**Laravel:** Tiene una sintaxis elegante y expresiva. Es altamente escalable. Su ecosistema es bastante amplio. Incluye métodos autenticación y seguridad. Tiene un sistema de plantillas propio fácil de usar. Permite automatizar diversas tareas a través de comandos. Incluye Eloquent como su ORM.

Tabla 1. Comparativa entre Frameworks de Php (Chávez, 2022)

Se ha seleccionado “Laravel” como el “*framework”* para el desarrollo de esta aplicación por los motivos de ser altamente escalable, tiene un sistema propio de plantillas, es fácil de aprender y automatiza distintas tareas por medio de comandos.

2.4.3 Bases de datos:

“Toda aplicación informática que se crea hoy en día debe operar sobre una base de datos” (Piñeiro Gomez, 2014). En la actualidad existen varios tipos de bases de datos, pero dentro de las que más se usan en la actualidad están las relacionales también conocidas como bases de datos SQL.

Para administrar las bases de datos se requiere de un sistema de gestión de base de datos abreviado como “DBMS”, esto permite acceder a las mismas y realizar acciones como: visualización de los datos, crear copias de seguridad, revisar el rendimiento, entre otros.

MariaDB: Se encuentra en la actualidad en la versión 10.8, es de código abierto y de licencia libre, es el motor de gestos de base de datos seleccionado para este proyecto debido a que:

- Es ligero y fácil de implementar.
- El tipo de licencia se adapta a las necesidades de la aplicación.
- Interactúa bien con PHP y el “*framework*” de desarrollo “Laravel”.

2.4.4 Herramientas de desarrollo

Dentro de las herramientas a desarrollo que se van a usar están:

Xamp: Es un entorno de desarrollo el cual incluye todas las características que se necesitaran durante el desarrollo del proyecto, como el lenguaje PHP, el servidor integrado “Apache” y la base de datos “MariaDB”.

Visual Studio Code: Es un editor de texto de uso libre el cual es altamente configurable permitiendo trabajar con cualquier lenguaje de programación con la ayuda de extensiones.

Git: Es un software para la administración de control de versión de código abierto y de uso libre.

GitHub: Es un software el cual permite la colaboración con “*git*” y a su vez poder crear repositorios en la nube para de esta manera respaldar el trabajo realizado.

Trello: Es un software en línea el cual se especializa en la gestión de proyecto mediante el uso de tableros.

2.4.5 Herramientas de producción

Las herramientas de producción serán utilizadas al momento de subir el proyecto a la web para que pueda ser accedido desde cualquier lugar.

Laravel Forge: Es un servicio el cual permite la implementación rápida y eficiente de proyectos "Laravel" en distintos proveedores de servicios en la nube, facilitando la administración de estos con la ayuda de herramientas visuales.

Digital Ocean: Es un proveedor de servicios en la nube el cual se enfoca en facilitar diferentes operaciones a los desarrolladores. Se ha escogido para este proyecto debido a que es compatible con "Laravel Forge" y tiene diferentes planes y precios los cuales permite escalar según las necesidades de los sistemas o proyectos.

2.4.6 Pasarelas de pago.

"Las pasarelas de pagos son sistemas de pago electrónico que permiten la realización de pagos y transferencias entre tiendas electrónicas y entidades bancarias de manera segura". (Solis Fonseca, Roque Pérez, & Morilla Faurés, 2013).

Se integrará una pasarela de pagos en este proyecto para que se puedan realizar y recibir pagos, se ha seleccionado "PayPal" para ser el medio de pago el cual se va a integrar gracias a sus características las cuales son:

- Permite transacciones seguras.
- Es fácil de implementar en aplicaciones web.
- Ofrece un entorno de prueba
- Es una de plataformas de pagos más conocidas de la actualidad.

2.4.7 Resumen de la tecnología seleccionada

Tecnología seleccionada	Motivos
Lenguajes de programación	
PhP	Lenguaje principal de la aplicación
	Sintaxis limpia y fácil de aprender
	Trabaja bien con distintas bases de datos
JavaScript	Servirá para darle dinamismo y reactividad a la aplicación
Framework	
Laravel	Es altamente escalable
	Sistema propio de plantillas
	Fácil de aprender

	Tareas automatizadas a través de comandos
Bases de datos	
MariaDB	Fácil de implementar
	Licencia que se adapta a la aplicación
	Interactúa bien con PHP y Laravel
Herramientas de desarrollo	
Xamp	Incluye el lenguaje PHP, servidor apache y MariaDB como base de datos
Visual Studio Code	Editor de texto altamente configurable
Git	Software para control de versiones
GitHub	Permite crear repositorios en la nube en conjunto con GitHub
Trello	Software en línea que permite la gestión de proyecto mediante tableros
Herramientas de producción	
Laravel Forge	Permite desplegar aplicaciones Laravel de manera fácil
Digital Ocean	Proveedor de servicios en la nube con distintos planes y precios.
Pasarelas de pago	
PayPal	Transacciones seguras
	Fácil de implementar
	Entorno de prueba

Tabla 2. Resumen de la tecnología seleccionada (Chávez, 2022)

2.5 Selección de metodología:

"Es primordial que los desarrolladores realicen software por medio de metodologías que permitan guiar su trabajo y acoplar el producto final con aspectos como la funcionalidad, seguridad, consistencia y fiabilidad" (Molina Ríos, Zea Ordoñez, Contento Segarra, & García Zerda, 2018).

En la actualidad cuando se trata de implementar metodologías a un proyecto de software se puede elegir entre metodologías tradicionales y agiles, la elección de esta dependerá de la compatibilidad del proyecto con la misma.

2.5.1 Metodologías Tradicionales.

Son las primeras metodologías que surgieron de ahí su nombre, tiene un enfoque secuencial, por lo que se dividen en fases de desarrollo, una vez terminada una fase se procede a la

siguiente de manera ordenada, son bastante predictivas con respecto al producto final de desarrollo siendo el más popular de estos:

Modelo en cascada:

Según Ortega y Camacho (2019) este modelo es considerado como el básico por su sencillez en compresión y antigüedad, siendo a su vez la base para el desarrollo de nuevas metodologías mejorando su estructura, ejecución y distribución de tareas.

Este modelo funciona de manera notable cuando se tiene un entendimiento claro de cuál va a ser el resultado final del proyecto, ya que tiene un progreso secuencial del mismo.

2.5.2 Metodologías Agiles

Las metodologías agiles aparecieron con el "manifiesto Agile" en el 2001 que fue propuesto por 17 firmantes los cuales manifestaron ideas de que el desarrollo de software debería ser cambiante y adaptarse siempre a las necesidades del usuario. Dentro de las metodologías agiles más destacables se tiene:

Scrum:

"Scrum es un marco de referencia para crear software complejo y entregarlo a tiempo de una forma mucho más sencilla" (Dimes, 2015). El cual está enfocado al desarrollo por equipos asignando roles a cada uno de los miembros de este, como lo son: "*Scrum Máster*", "*Product Owner"* y el equipo de desarrollo.

El proceso de este consiste en crear historias de usuarios en las cuales se evalúan todos los requerimientos de los usuarios, para ser establecidos como tareas a realizar, esto siendo parte del trabajo del "*Product Owner*".

Una vez se establecen las tareas a realizar estas se establecen como metas dentro de un tiempo determinado el cual se conoce como "*sprints*". Conforme se desarrollan estas tareas el equipo de desarrollo suele tener reuniones diarias de no más de 15 minutos que ayudan al equipo a tener una mejor comunicación entre ellos y una mejor percepción del desarrollo del proyecto.

Una vez se finaliza cada sprint se realiza un análisis de como fue el desarrollo de este, de las dificultades que se presentaron, y como se podría mejorar el proceso de desarrollo para futuros "*sprints*".

Extreme Programming:

Es de las primeras metodologías agiles que aparecieron, "concebida para proyectos de pequeño y mediano tamaño, prevé participación y frecuente con el cliente. Describe detalladamente las prácticas de desarrollo que se van a utilizar." (Laínez Fuentes, 2015).

Extreme programming aparte de buscar la colaboración entre el equipo de desarrollo y el ciente, también propone preferir la comunicación verbal antes que la escrita, a su vez también sugiere que los desarrolladores se centren en diseñar solo las partes que van a ser programadas en ese momento sin enfocarse demasiado a futuro, dividiendo todo el trabajo en iteraciones.

Cada iteración consta de 4 fases las cuales se repiten hasta que se finalice el proyecto las cuales son; planeación, diseño, codificación y prueba.

Kanban:

Kanban no es en sí una metodología de desarrollo, pero se ha venido aplicando en el desarrollo ágil en la comunidad gracias a que se adapta a los principios del "manifiesto agile". "El motivo para crear el método fue principalmente el encontrar una mejor manera de gestionar los negocios de servicios profesionales, así como una manera de proveer un método humano para el cambio" (Kanban University, 2011).

Se diferencia de las otras metodologías agiles de desarrollo en el sentido de que no permite tener una referencia del avance del proyecto, pero permite tener un excelente control de las tareas que se están realizando, las que fueron probadas y aquellas que ya fueron terminadas.

Por el motivo de que el equipo de desarrollo está compuesto por una sola persona y no hay necesidad de dividir el trabajo en intervalos de tiempo se ha optado por el uso de Kanban como metodología de desarrollo ágil.

Capítulo 3: Desarrollo de la aplicación web.

3.1 Establecer requerimientos

Como fue mencionado en el capítulo anterior el desarrollo de la aplicación web será utilizando el método Kanban.

Kanban funciona con lo que se conoce como "tablero Kanban", el cual no es más que un tablero en el cual está compuesto por columnas en las cuales se colocan encima las cartas de los procesos que se van a desarrollar. El número de columnas por tablero depende de las necesidades del proyecto.

Se va a usar el software en línea "trello" para realizar el tablero Kanban de esta aplicación web. Para este caso se creó un tablero el cual consta de 5 columnas las cuales son: Requerimientos, Diseño, Desarrollo, Pruebas y Realizado:

En la parte de requerimientos de tablero Kanban se agregarán los requerimientos funcionales para el desarrollo de esta aplicación web

Administrar Usuarios y Roles:

La aplicación necesita que sea posible el registro de nuevos usuarios, que los usuarios puedan mantener sesiones, que los usuarios dispongan de roles los cuales serán cliente y administrador, que los usuarios con el rol administrador sean capaces de administrar a otros usuarios, y que tengan acceso a una parte administrativa de la aplicación.

Administrar Productos:

Es necesario que los usuarios administradores puedan crear, editar y eliminar los productos que se ofrezcan dentro de la aplicación.

Administrar Categorías:

Es necesario que los productos pertenezcan a una categoría en específico, por lo cual los administradores de la aplicación podrán crear, editar y eliminar categorías.

Implementar carrito de compras:

Al ser una aplicación web de tipo e-commerce es necesario que se implemente un carrito de compras, en el cual se puedan almacenar de manera temporal los productos en los cuales los usuarios que visiten la aplicación puedan tener interés.

Administrar direcciones de envío:

Sera necesario que los usuarios administradores puedan crear, editar y eliminar, direcciones de envió disponibles para los usuarios que deseen realizar una compra.

Administrar Órdenes:

Una vez el usuario seleccione los productos en los cuales tiene intención de comprar podrá generar una orden de compra, los administradores podrán cambiar el estado de las órdenes los cuales serán: pagado, enviado, entregado o anulado.

Implementar pasarela de pago:

Para que los usuarios paguen las ordenes que generen se implementara una pasarela de pago que se adapte a las necesidades del negocio.

Administrar Servicios:

Esta aplicación web también permitirá que los administradores puedan crear, editar y eliminar los servicios que ofrece el negocio, a diferencia de los productos los usuarios de la página podrán informarse mas no interactuar con los mismos.

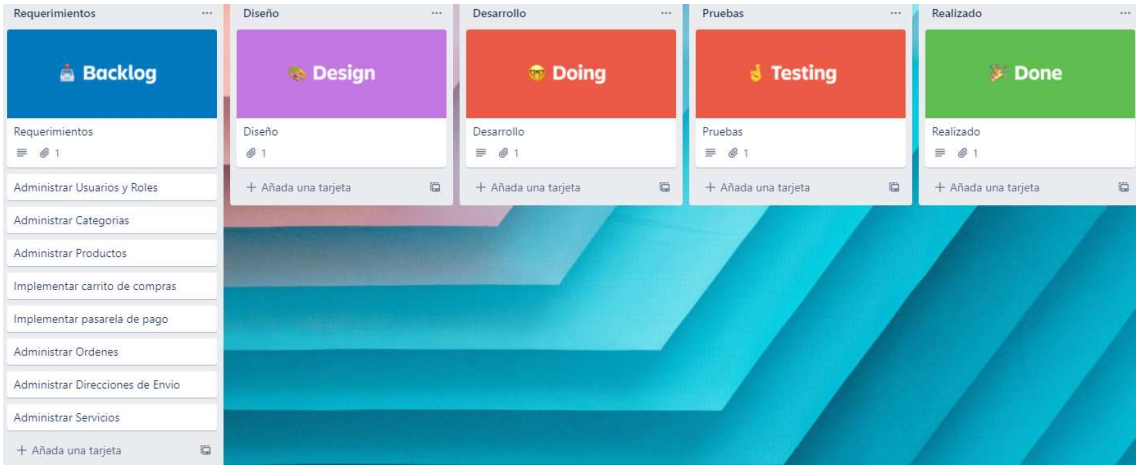


Figura 1. Inicialización del tablero Kanban (Chávez, 2022)

3.2 Diseño de la base de datos

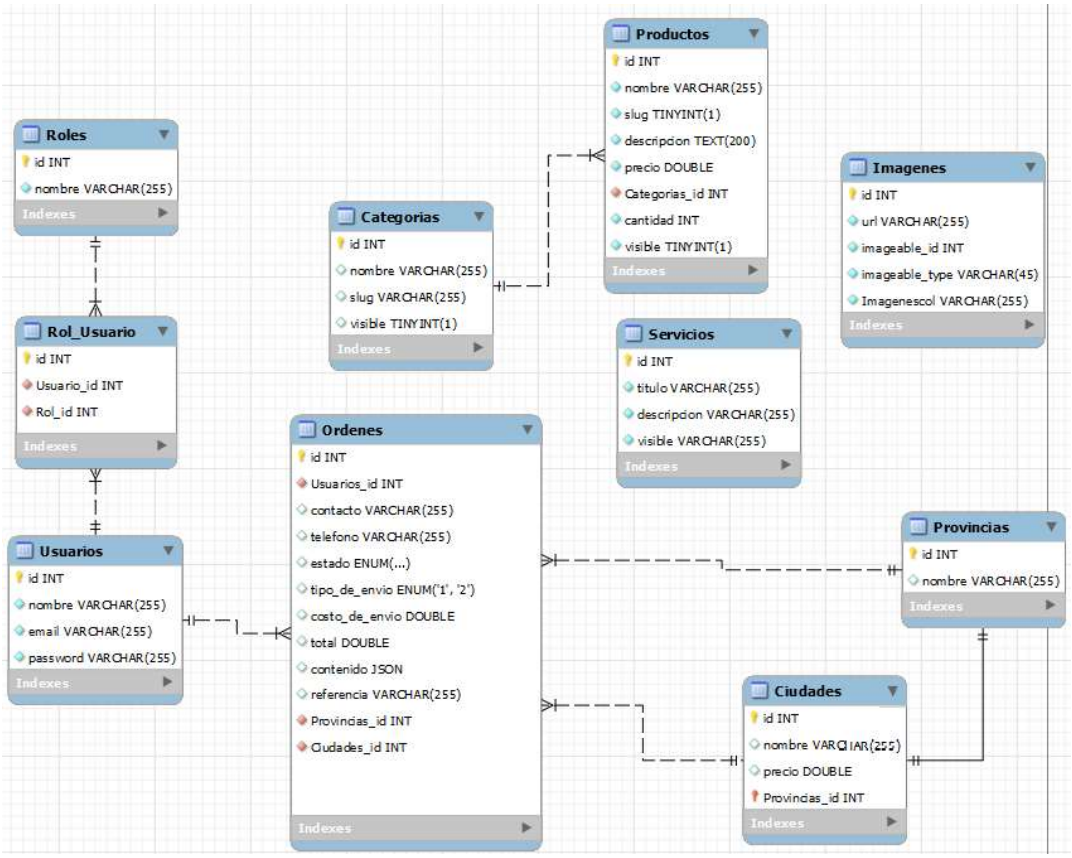


Figura 2. Diseño de la base de datos (Chávez, 2022)

Para visualizar el diccionario de datos véase el Anexo 1.

3.2 Configuración de entorno e inicialización de proyecto:

El proyecto va a ser desarrollado con “Laravel” por lo que es necesario crearlo dentro de la carpeta “htdocs” del directorio “xamp” que es donde se encuentra instalado el lenguaje “Php” de esta herramienta de desarrollo. Una vez allí es necesario ejecuta el comando “laravel new” en conjunto con el nombre que se le va a asignar al proyecto y automáticamente comenzará la creación de este.

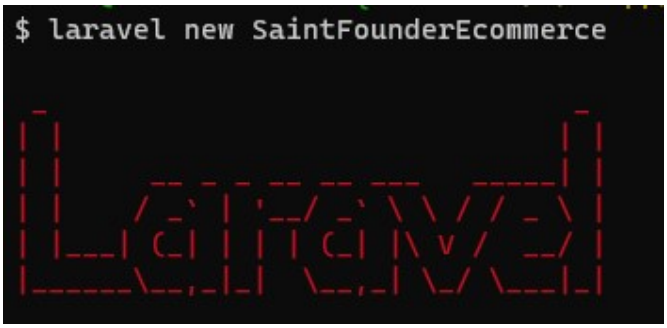


Figura 3. Creación del proyecto (Chávez, 2022)

3.2. Administrar Usuarios y roles

Esta es la primera tarea para realizarse de la columna de requerimientos, se agregará la capacidad de la aplicación web de permitir la creación de usuarios los cuales puedan mantener sesiones y a su vez que los usuarios puedan tener roles los cuales serán usuario normal y administrador.

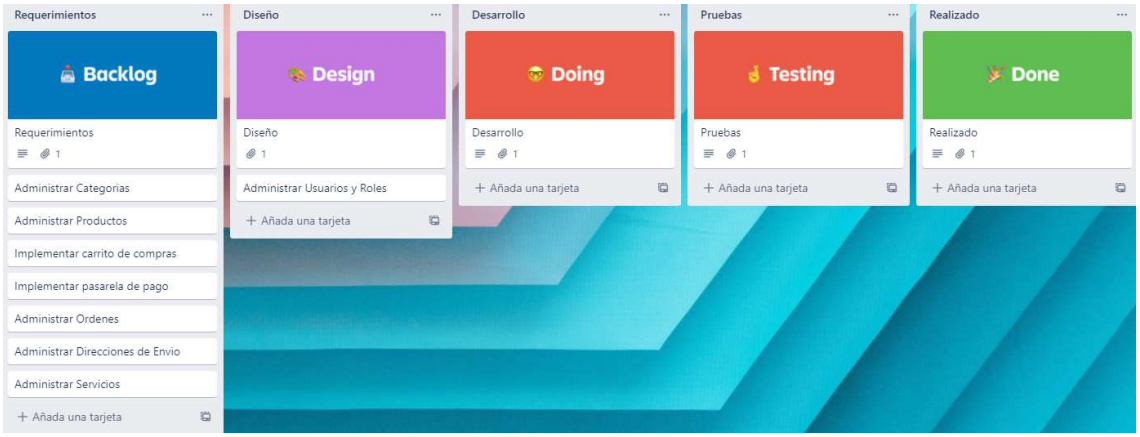


Figura 4. Tablero Kanban-Diseño de administrar usuarios y roles (Chávez, 2022)

Diseño:

Se va a realizar un diagrama de casos de uso que muestre de manera visual el funcionamiento de esta parte del sistema.

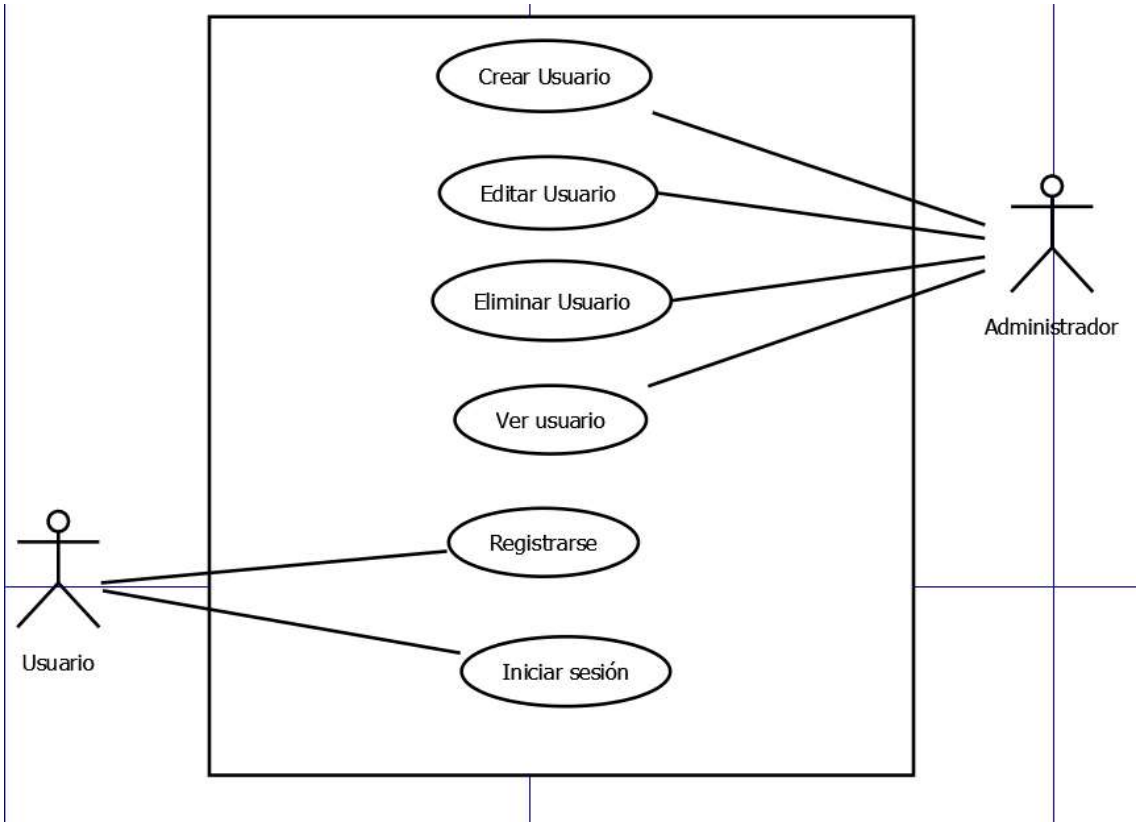


Figura 5. Diagrama de casos de uso de administración de usuarios y roles (Chávez, 2022)

Desarrollo:

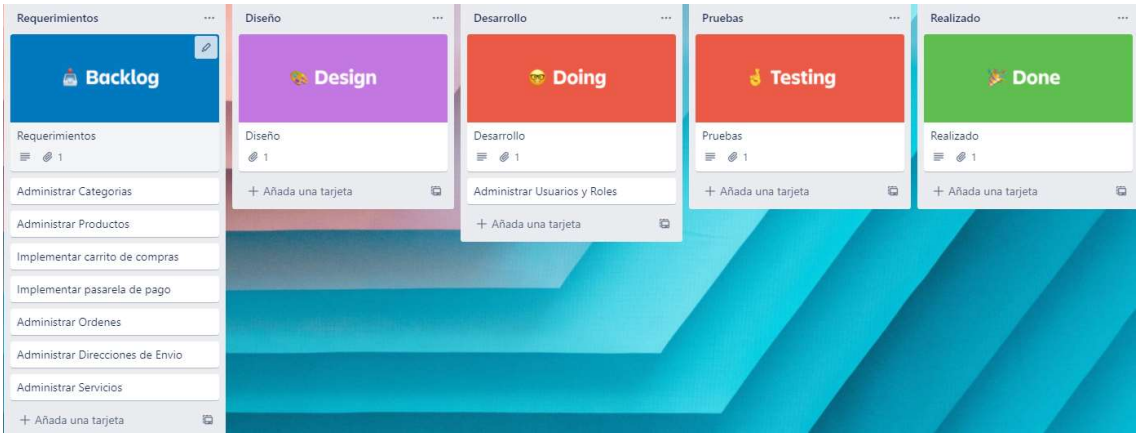


Figura 6. Tablero Kanban – Desarrollo administración de usuarios y roles (Chávez, 2022)

Se debe instalar el paquete propio de "Laravel" denominado "*Jetstream*" el cual incluye diversos estilos, y un sistema de autenticación que nos sirve para autenticar usuarios y mantener sesiones. Como es necesario que existan usuarios con el rol de administrador se instalara el paquete "*laravel permission*", con el cual creamos el rol administrador.

Cuando ha sido realizado lo anterior cabe destacar que Laravel tiene la capacidad de generar modelos y migraciones, los cuales no permiten mediante código configurar la base de datos, por lo que se nos genera automáticamente las tablas y los campos necesarios en la base de datos.

Una vez creado el rol administrador se procede a proteger las rutas, creando un archivo llamado admin.php, en el cual se encontrarán todas las rutas protegidas en las cuales solo pueden entrar los administradores de la página. Dentro de este se crearon todas las rutas necesario para administración de usuarios los cuales serían el poder crear, editar y eliminar usuarios.

Para que se pueda administrar se ha creado una tabla en la cual se muestran todos los usuarios, también se ha puesto un campo de entrada en el cual se pueden realizar búsquedas de usuarios y se ha agregado un botón el cual permite crear un nuevo usuario.

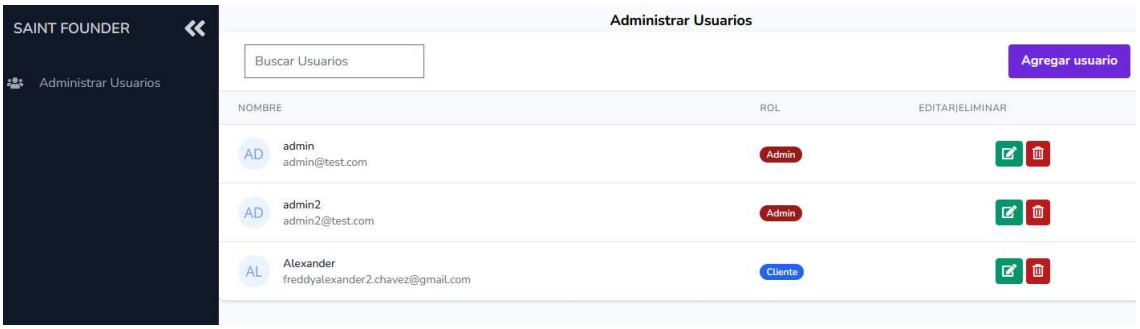


Figura 7. Tabla para administrar usuarios (Chávez, 2022)

Una vez se da clic el botón de “agregar usuario”, se redirecciona una sección para crear un usuario nuevo en donde se nos pide que se llenen los apartados de nombre, correo electrónico, contraseña y confirmación de contraseña.

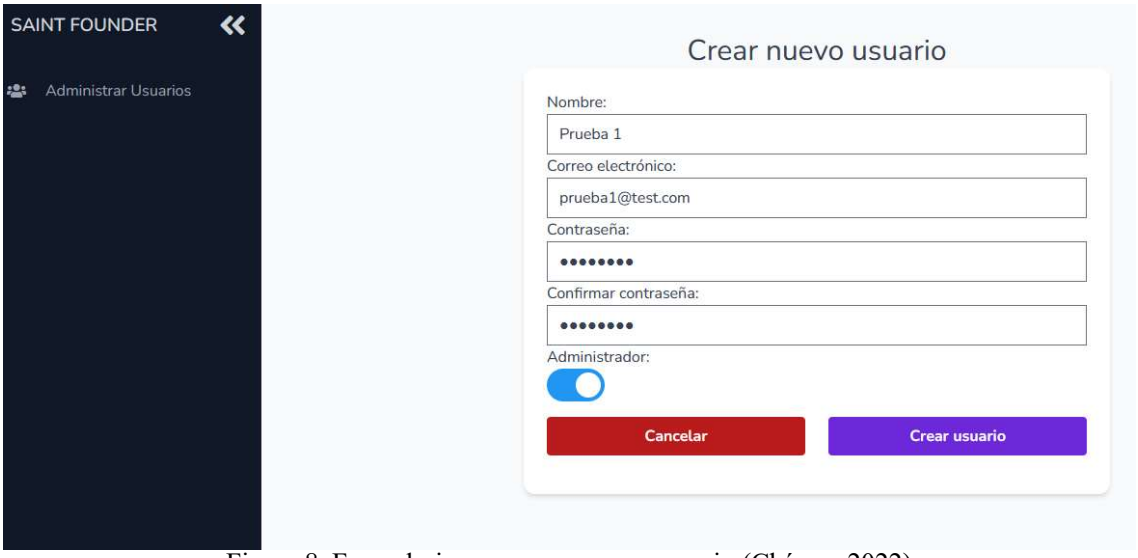


Figura 8. Formulario para crear nuevo usuario (Chávez, 2022)

Si se selección el botón verde en la tabla de usuarios se redireccionara hacia la vista para editar un usuario, el cual funcionara de manera similar a la vista de crear nuevo usuario.



Figura 9. Formulario para editar un usuario (Chávez, 2022)

En caso de que se seleccione el botón de color rojo se desplegara en pantalla una confirmación de necesaria en caso de que de verdad se desee eliminar el usuario.

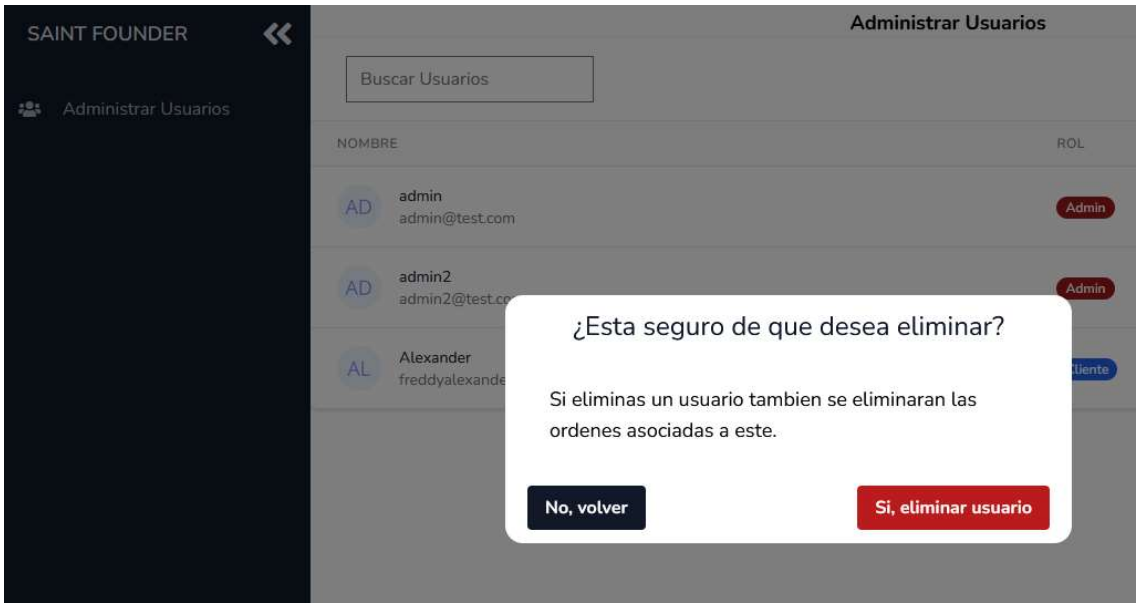


Figura 10. Confirmación para editar un usuario (Chávez, 2022)

Para la parte de los usuarios, en caso de ser un usuario nuevo tiene existe la posibilidad de registrarse. Para los usuarios que tengan una cuenta registrada pueden iniciar sesión con sus credenciales que han sido previamente registrados.

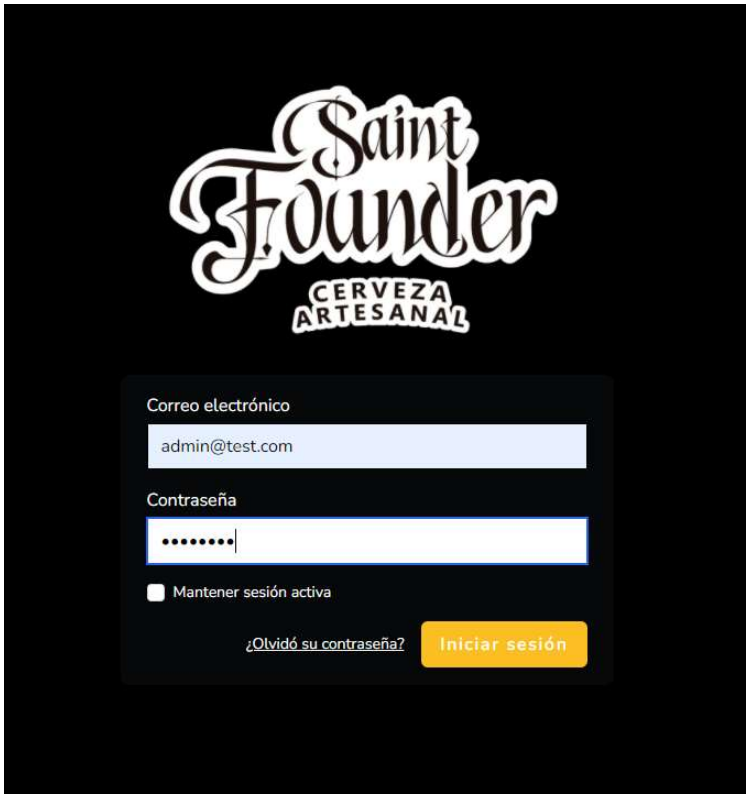


Figura 11. Formulario para iniciar sesión de usuario (Chávez, 2022)

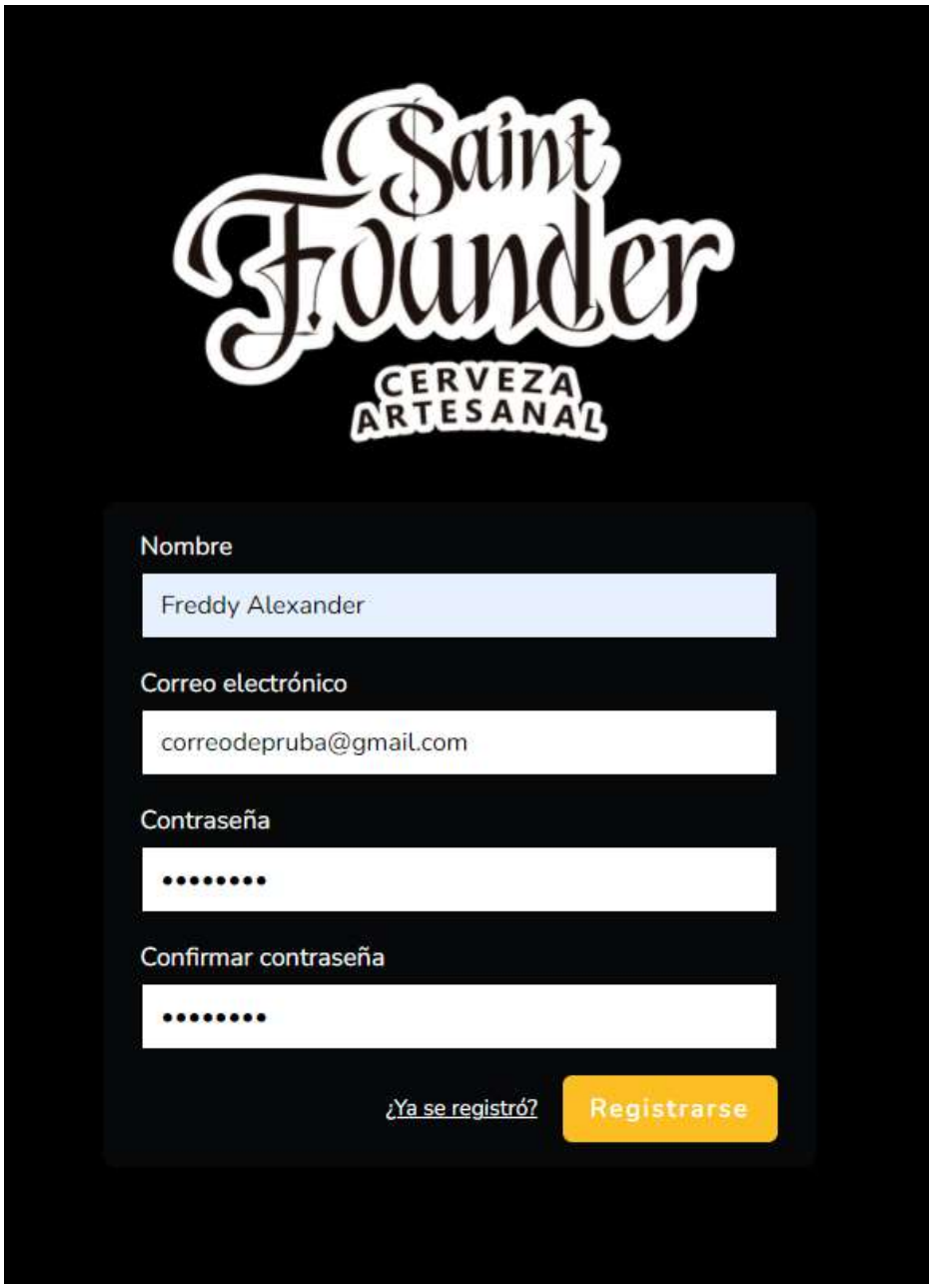


Figura 12. Formulario para registrar un nuevo usuario (Chávez, 2022)

Pruebas:

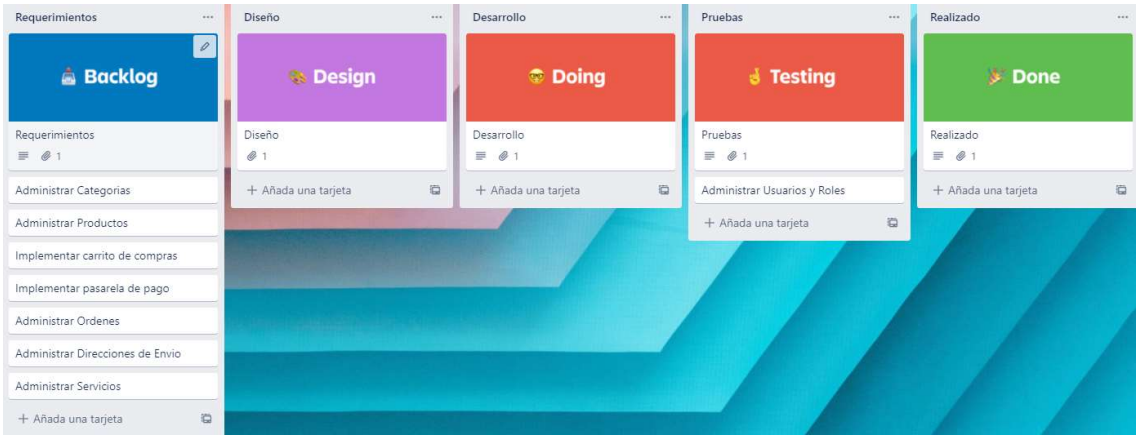


Figura 13. Tablero Kanban –Pruebas de administración de usuarios y roles (Chávez, 2022)

Se han generado criterios de aceptación para considerar que esta parte está bien realizada los cuales son:

Escenarios	**Casos de prueba**	**Estado**
Crear Usuario	Se comprueba que el correo esté disponible	Funcional
	Se comprueba que la contraseña tenga más de 8 dígitos	Funcional
	Se verifica que las contraseñas coincidan	Funcional
	Se puede asignar rol administrador	Funcional
Editar Usuario	Se comprueba que el correo esté disponible	Funcional
	Se comprueba que la contraseña tenga más de 8 dígitos	Funcional
	Se verifica que las contraseñas coincidan	Funcional
Eliminar Usuario	Se presenta una confirmación en pantalla	Funcional
	Se elimina el usuario de la base de datos	Funcional
Registrar Usuario	Se comprueba que el correo esté disponible	Funcional
	Se comprueba que la contraseña tenga más de 8 dígitos	Funcional
	Se verifica que las contraseñas coincidan	Funcional
Iniciar Sesión	Se verifica que exista el correo electrónico	Funcional
	Se verifica que la contraseña y el correo electrónico exista	Funcional

Tabla 3. Casos de prueba de administración de usuarios (Chávez, 2022)

Una vez que todos estos criterios de aceptación han sido aprobados se procede a continuar con el siguiente requerimiento de la "tabla Kanban".

3.3 Administrar Categorías

Diseño:

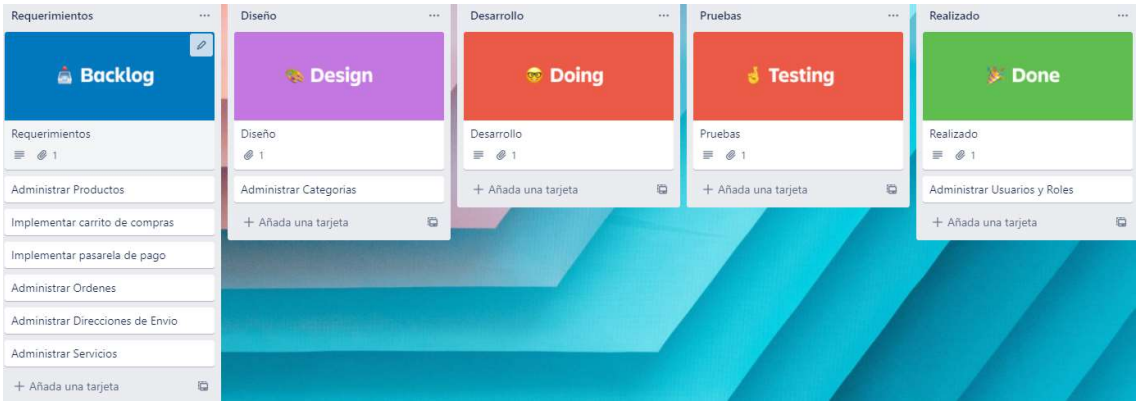


Figura 14. Tablero Kanban -Diseño administrar categorías (Chávez, 2022)

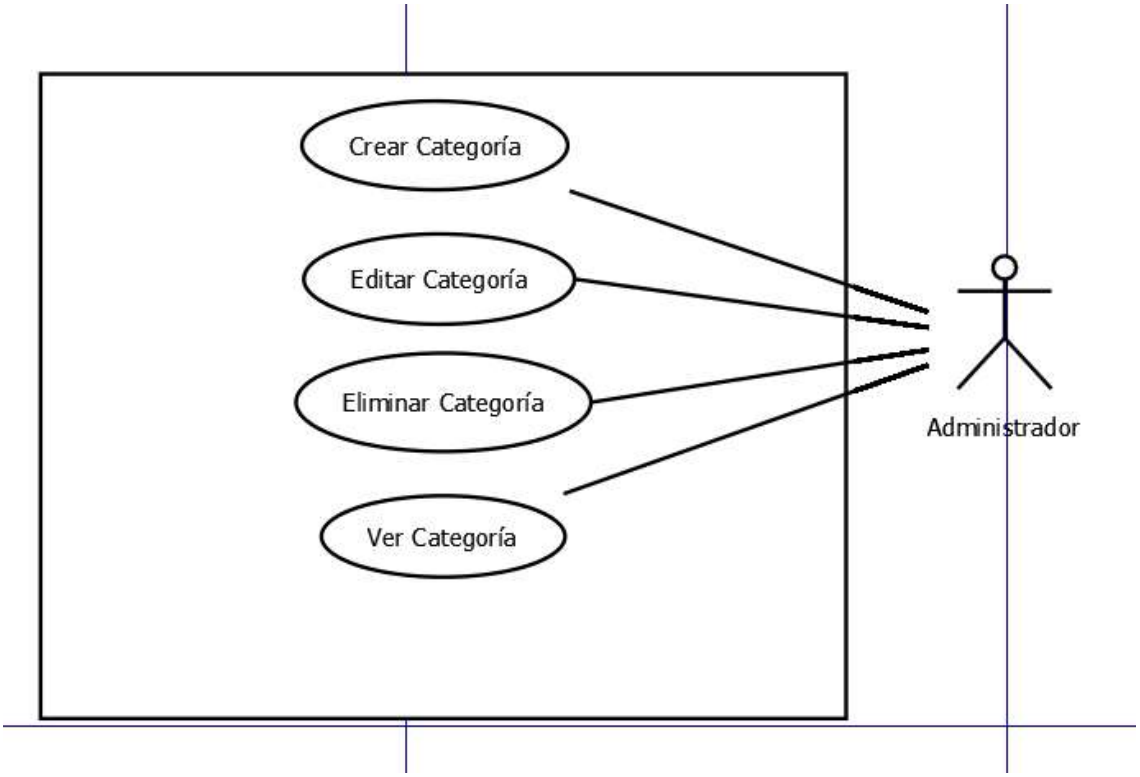


Figura 15. Diseño administrar categorías (Chávez, 2022)

Desarrollo:

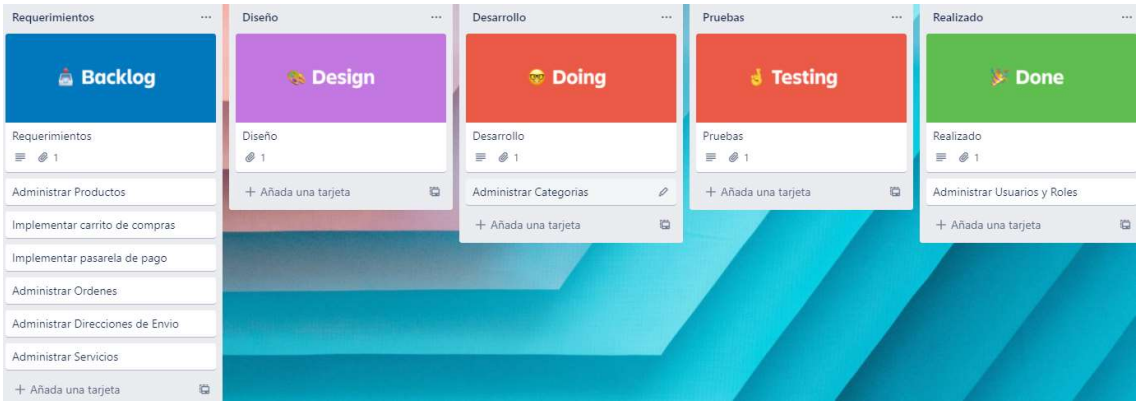


Figura 16. Tablero Kanban -Desarrollo administrar categorías (Chávez, 2022)

Cada producto deberá pertenecer a una categoría por lo cual se ha de desarrollar primero el apartado de administrar categorías antes que el de productos. Es necesario crear un modelo y migración de categoría el cual tiene la capacidad de albergar campos como: el nombre, la "*url*" amigable denominada como "*slug*" y el estado el cual puede ser cómo visible u oculto.

Para que se puedan administrar de manera correcta se ha creado una tabla en la cual se pueden visualizar las categorías de la aplicación, arriba de esta existe un apartado para poder realizar búsqueda en las categorías y un botón para crear una nueva categoría.

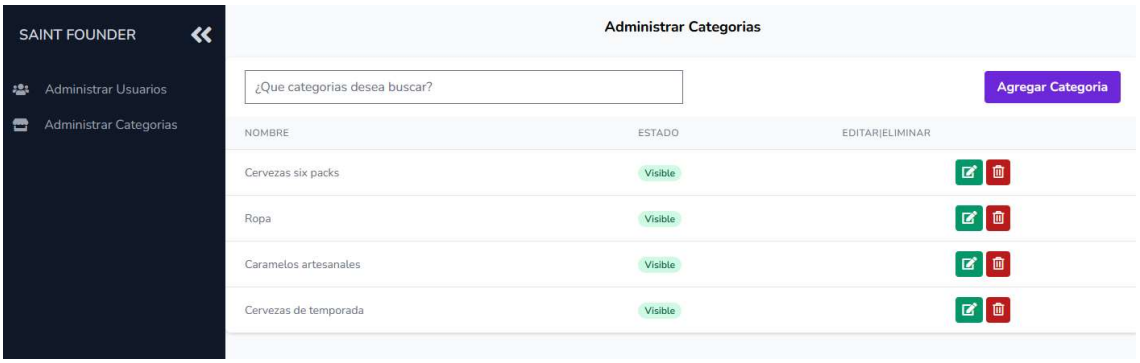


Figura 17. Tabla para administrar categorías (Chávez, 2022)

Cuando se selecciona la opción de agregar una nueva categoría se abre un apartado para agregar una categoría nueva, en los cuales se puede poner el nombre de la categoría, el "*slug*" que es lo que se conoce como "*url*" amigable y la visibilidad de modo que se pueda mostrar u ocultar sin necesidad de tener que eliminar y crear de nuevo.

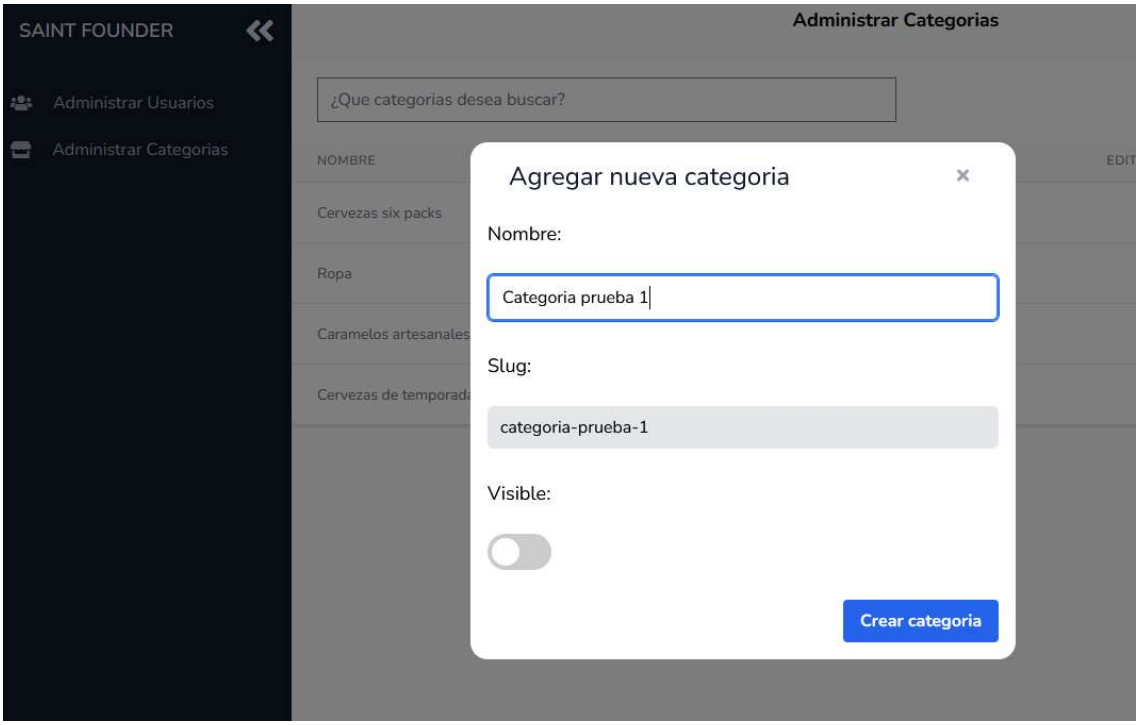


Figura 18. Formulario para crear una categoría (Chávez, 2022)

Si se selecciona el botón verde en la tabla este nos mostrará un aparatado en el cual se podrá editar la categoría seleccionada.

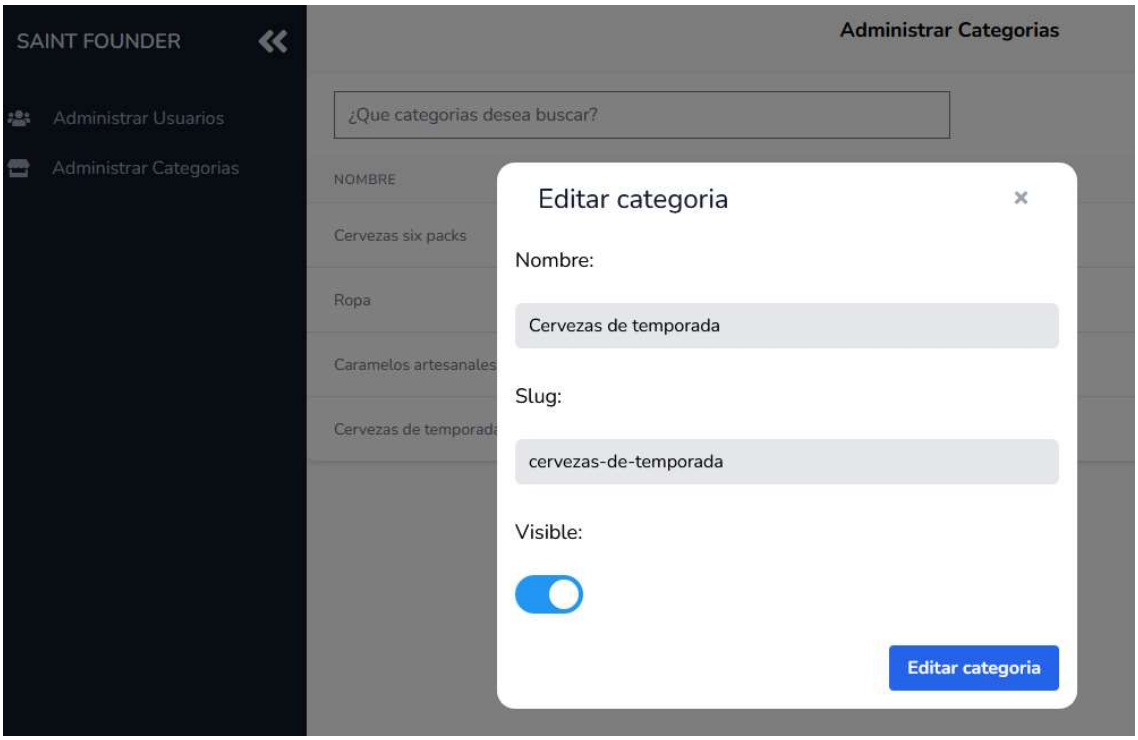


Figura 19. Formulario para editar una categoría (Chávez, 2022)

En caso de que sea necesario eliminar una categoría es necesario darle clic sobre el botón rojo de la tabla, en donde se va a mostrar una confirmación en pantalla para asegurar que de verdad se requiere eliminar la categoría seleccionada.

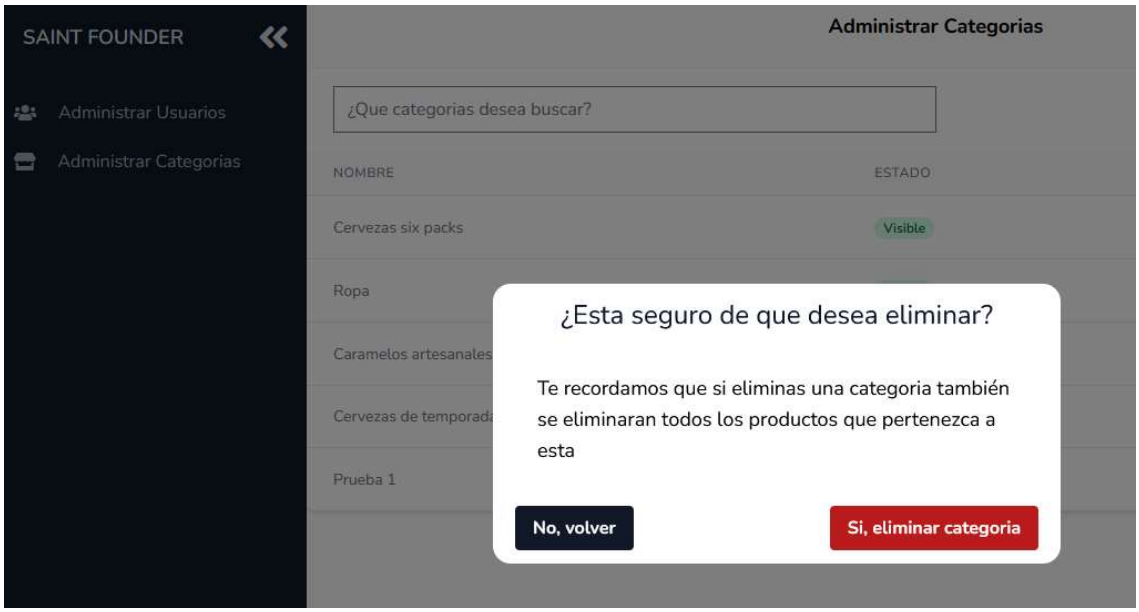


Figura 20. Confirmación para eliminar una categoría (Chávez, 2022)

Pruebas:

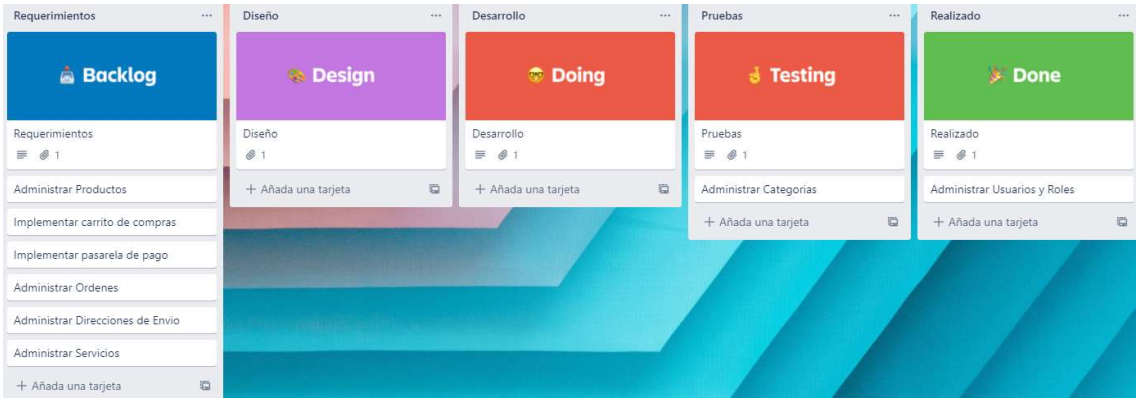


Figura 21. Tablero Kanban – Pruebas de administración de categorías (Chávez, 2022)

Escenarios	Casos de prueba	Estado
Crea Categoría	Se puede crear una categoría	Funcional
	El slug de la categoría coincide con el nombre	Funcional
	Se verifica que el slug sea único	Funcional
	Se puede establecer el estado de la categoría	Funcional
Editar Categoría	Se puede cambiar el nombre de la categoría	Funcional
	El slug de la categoría coincide con el nombre	Funcional
	El slug de la categoría se puede cambiar	Funcional
Eliminar Categoría	Se presenta una confirmación en pantalla	Funcional
	Se elimina la categoría de la base de datos	Funcional

Tabla 4. Casos de prueba de administración de categorías (Chávez, 2022)

3.4 Administrar Productos

Diseño:

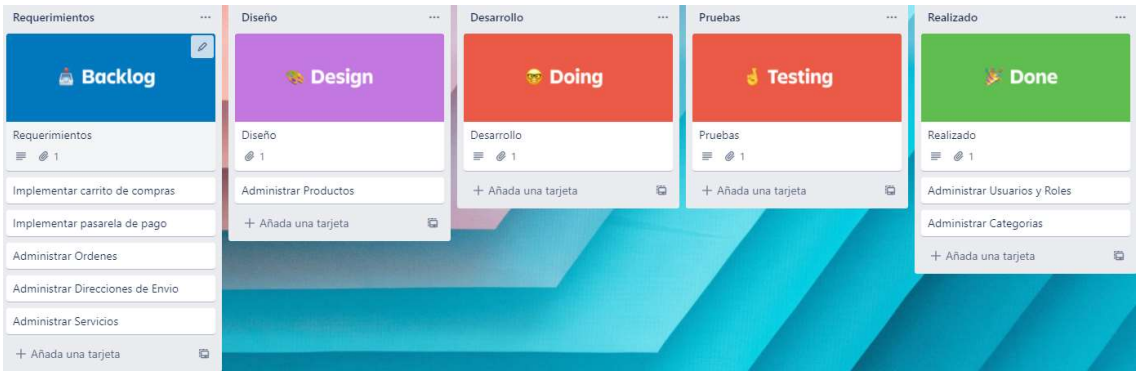


Figura 22. Tablero Kanban – Diseño de administrar productos (Chávez, 2022)

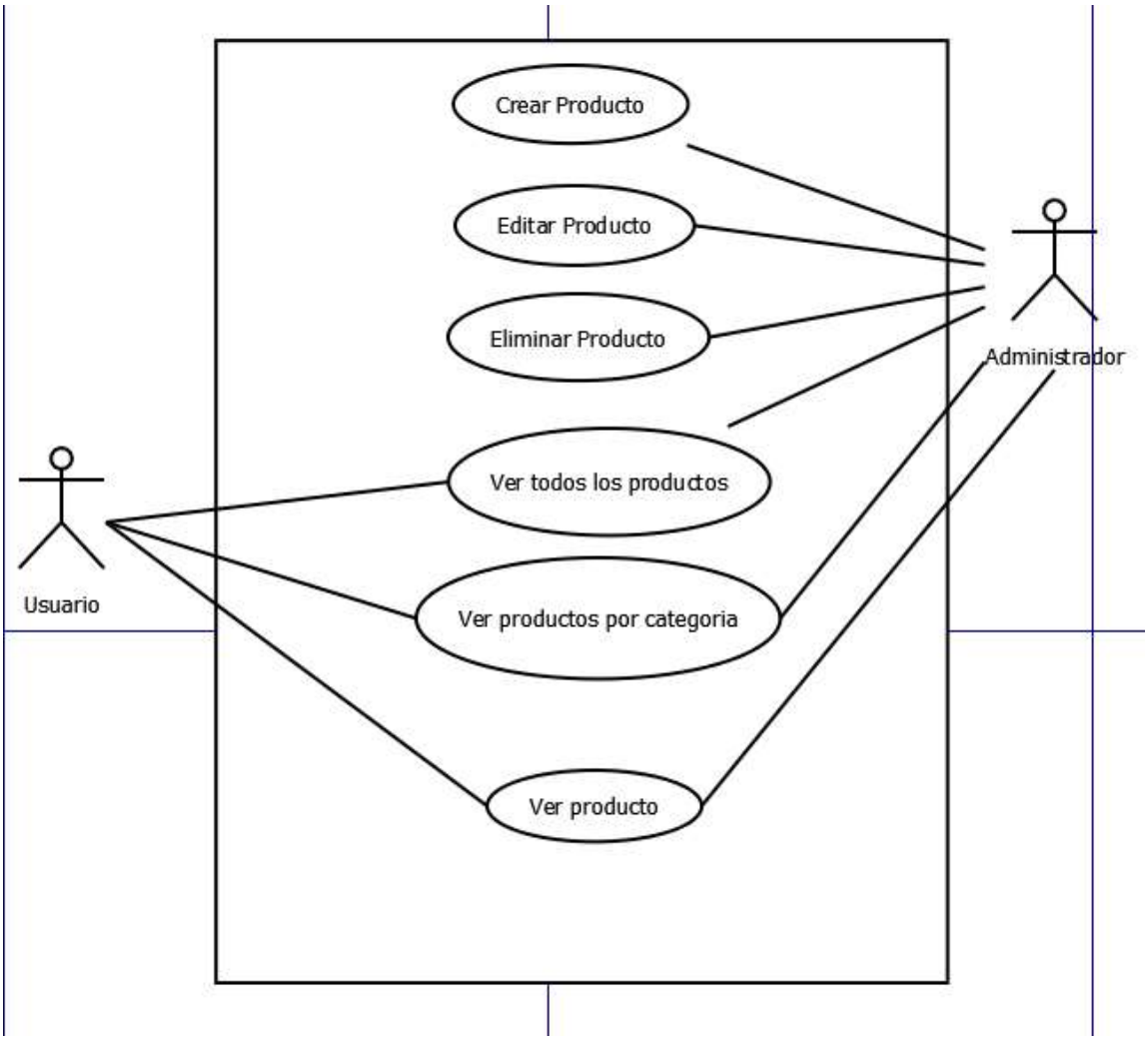


Figura 23. Diagrama de casos de uso de administrar productos (Chávez, 2022)

Desarrollo:

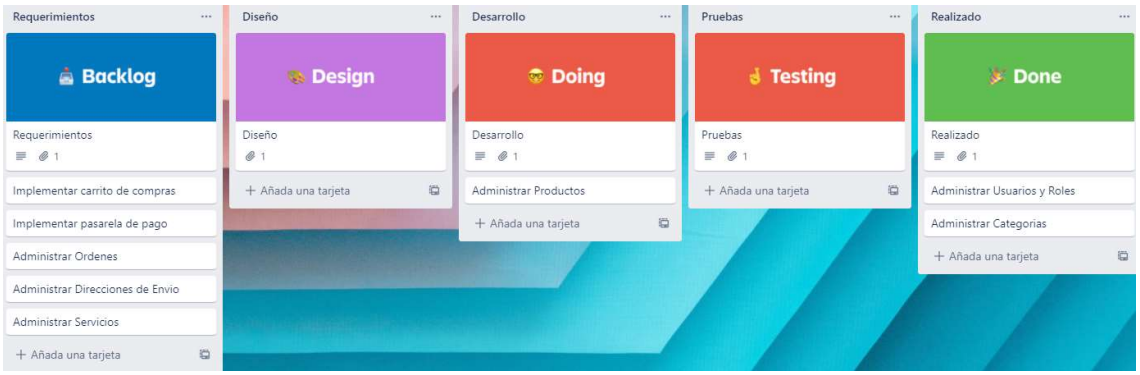


Figura 24. Tablero Kanban – Desarrollo de administrar productos (Chávez, 2022)

Una vez se ha realizado el apartado de administrar categorías, ya es posible crear productos y asociar estos a una categoría, por lo que se va a desarrollar primero la parte administrativa y luego la parte visible para todos los usuarios.

Se debe crear un modelo y migración de producto en el cual se incluyen los campos: a que categoría pertenece, cuál es su nombre, su "*url*" amigable con el nombre de "*slug*", la descripción del producto, el precio, la cantidad e imágenes asociadas a este.

Se ha llenado la base de datos con datos e imágenes de prueba para que sea pueda visualizar como funcionaria la aplicación con un gran número de datos. Se va a crear una tabla en la cual se puede visualizar los productos, arriba de esta existe un apartado para poder realizar búsquedas en los productos y un botón para crear un nuevo producto.

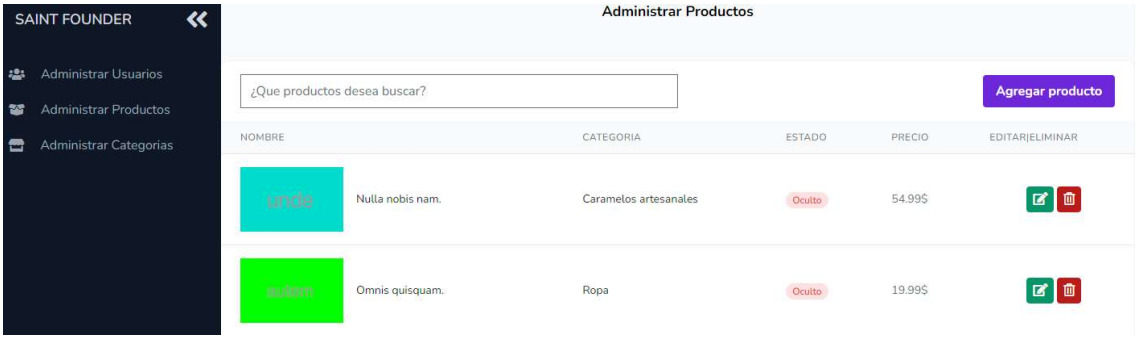


Figura 25. Tabla de administración de productos (Chávez, 2022)

Al momento de dar clic en el botón de agregar producto se redireccionará a un apartado en el cual será posible agregar un producto nuevo, en el cual se deberá seleccionar la categoría a la cual pertenece, el estado si es visible u oculto, el nombre del producto, la "*url*" amigable bajo el nombre de "*slug*", la descripción, el precio y la cantidad.

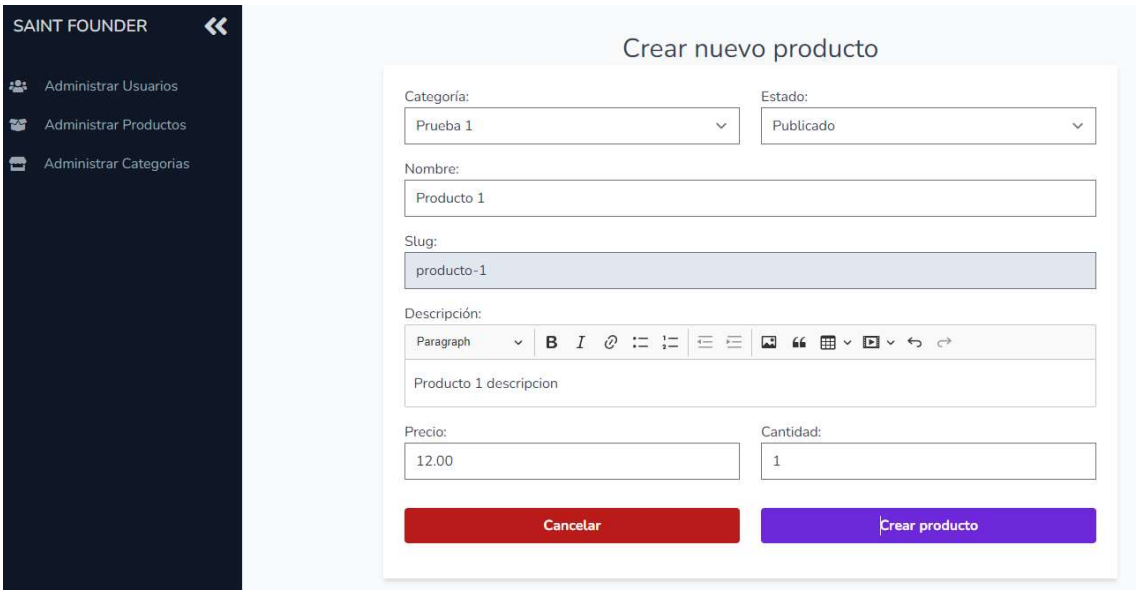


Figura 26. Formulario de creación de un nuevo producto (Chávez, 2022)

Cuando se dé clic en el botón verde de la tabla se podrá editar el producto seleccionado, en el apartado de editar se ha agregado la capacidad de enlazar imágenes al producto.

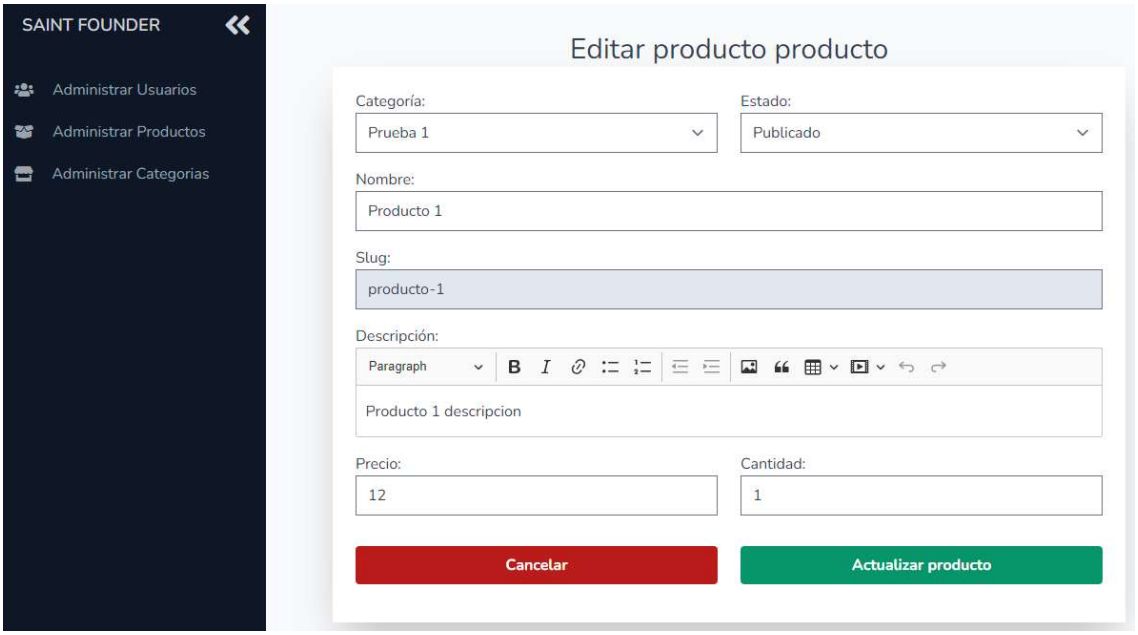


Figura 27. Formulario para editar un producto (Chávez, 2022)

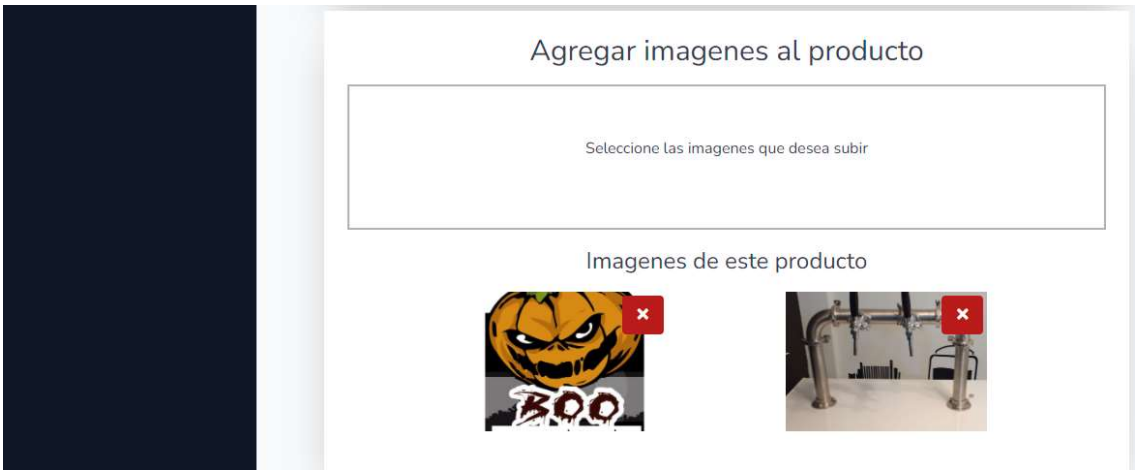


Figura 28. Formulario para agregar imágenes a un producto (Chávez, 2022)

En caso de que se necesite eliminar un producto por cualquier motivo, se va a mostrar una confirmación en pantalla para asegurarse que de verdad se desee eliminar el producto seleccionado.

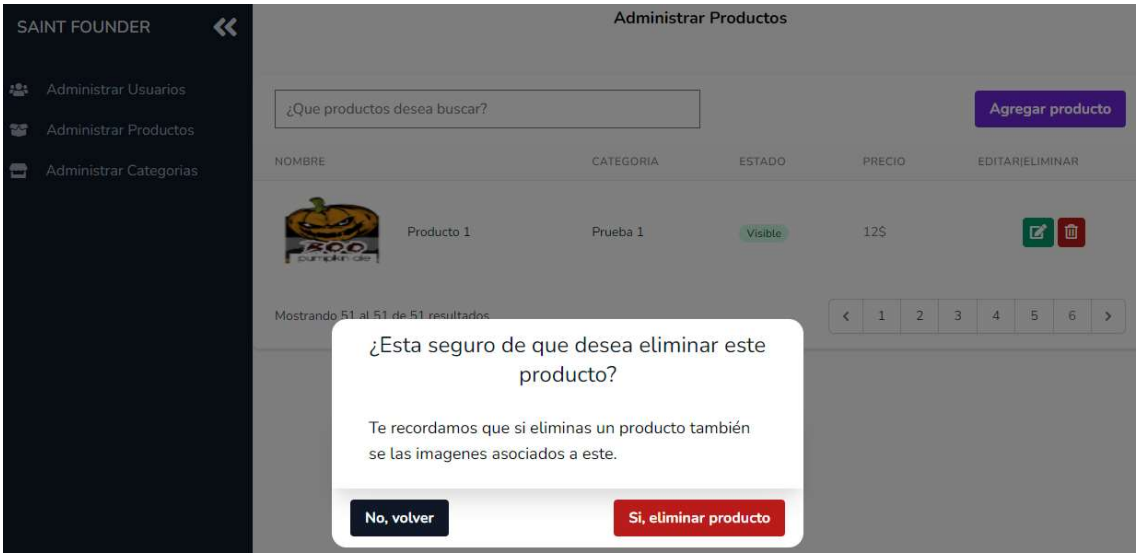


Figura 29. Confirmación para eliminar un producto (Chávez, 2022)

Para la vista de los usuarios clientes que entren a la página estos se encontrarán con una lista con las categorías y dentro de estas saldrán una muestra de los productos disponibles, con la opción de ver más.

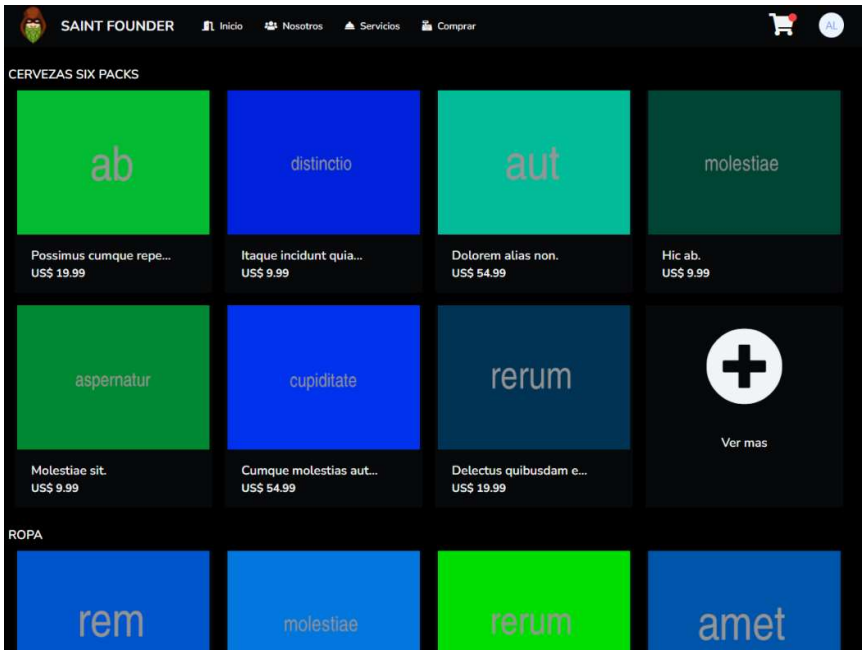


Figura 30. Vista de las categorías y productos (Chávez, 2022)

Si se selecciona la opción de ver más se abrirá redireccionará a una vista en donde se podrá navegar entre las categorías de una manera más sencilla además de que se permitirá realizar búsquedas.

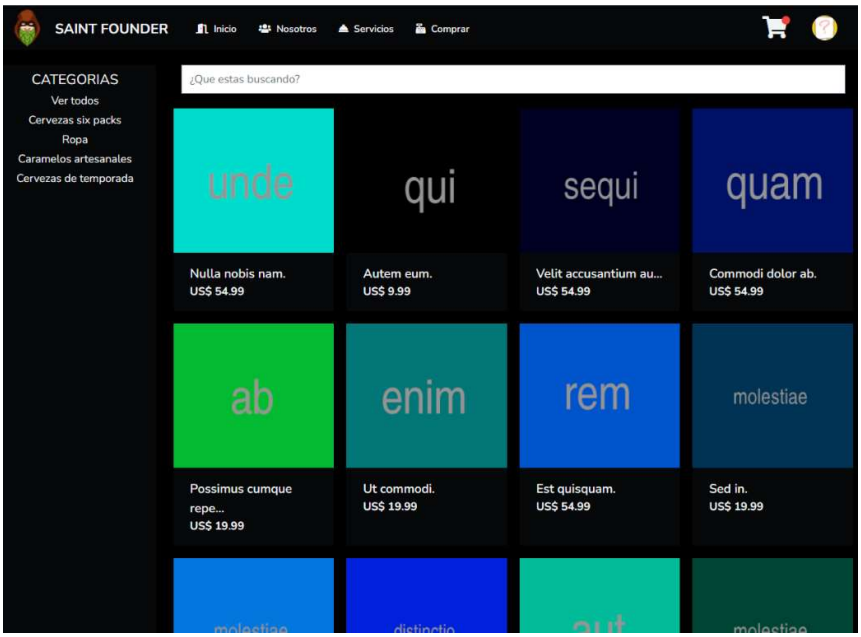


Figura 31. Vista de productos filtrado por categoría (Chávez, 2022)

Cuando se dé clic sobre el nombre de algún producto se redireccionará a una página donde se mostrará el producto, en conjunto con una descripción de este, el precio, la cantidad disponible y las imágenes asociadas a este producto.

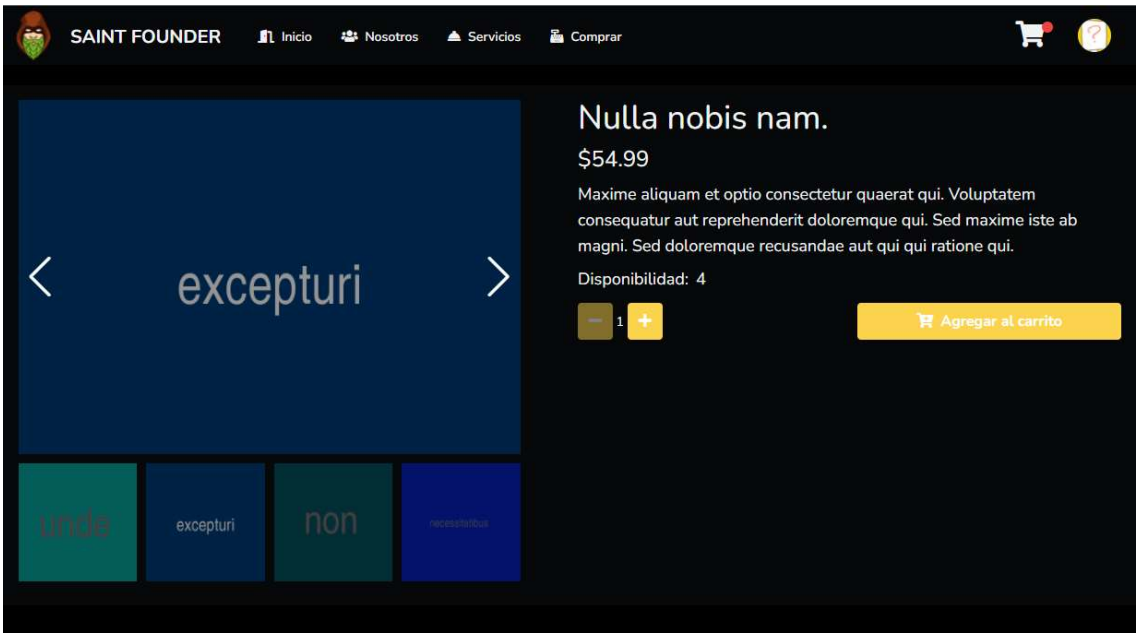


Figura 32. Vista de un producto con sus características (Chávez, 2022)

Pruebas:

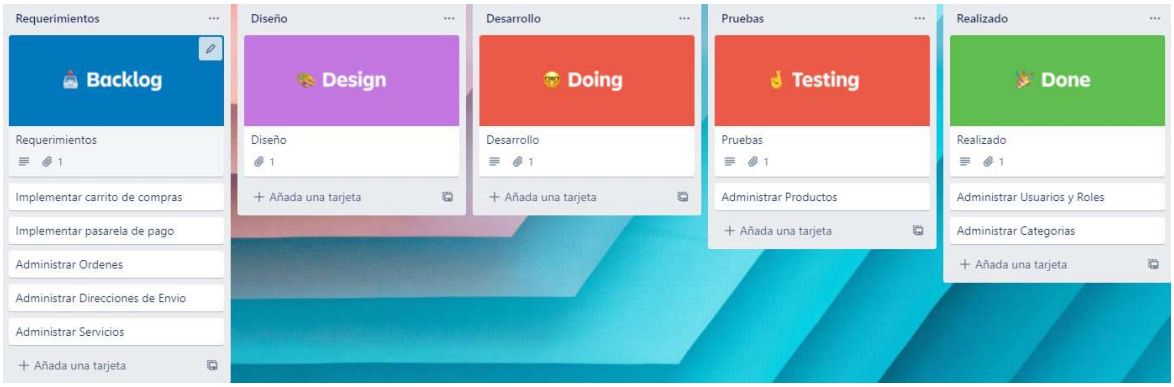


Figura 33. Tablero Kanban – Pruebas de administración de productos (Chávez, 2022)

Escenarios	Casos de prueba	Estado
Crea Producto	Se puede crear un producto	Funcional
	El slug de producto coincide con el nombre del producto	Funcional
	Se verifica que el slug sea único	Funcional
	Se pueden establecer los parámetros del producto	Funcional
Editar Producto	Se puede cambiar el nombre del producto	Funcional

	El slug del producto coincide con el nombre	Funcional
	El slug del producto se puede cambiar	Funcional
	Se pueden agregar imágenes al producto	Funcional
	Se pueden eliminar las imágenes del producto	Funcional
Eliminar Producto	Se presenta una confirmación en pantalla	Funcional
	Se elimina el producto de la base de datos	Funcional
	Se eliminan las imágenes del producto de la base de datos	Funcional
Ver todos los productos	Se muestran todos los productos en pantalla	Funcional
Ver productos por categoría	Se muestran los productos de una determinada categoría	Funcional
Ver producto	Se muestra el producto	Funcional
	Se muestran las imágenes del producto	Funcional
	Se muestran las características del producto	Funcional

Tabla 5. Casos de prueba de administrar productos (Chávez, 2022).

3.5 Implementar carrito de compras

Diseño:

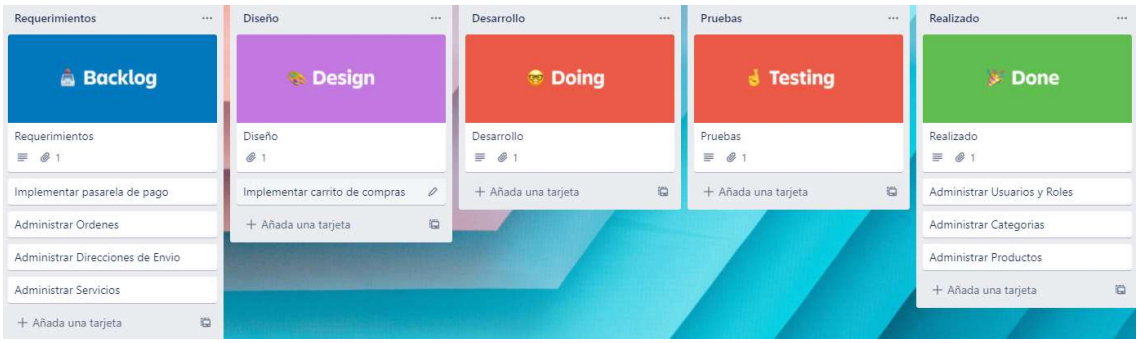


Figura 34. Tablero Kanban – Diseño de implementar carrito de compra (Chávez, 2022)

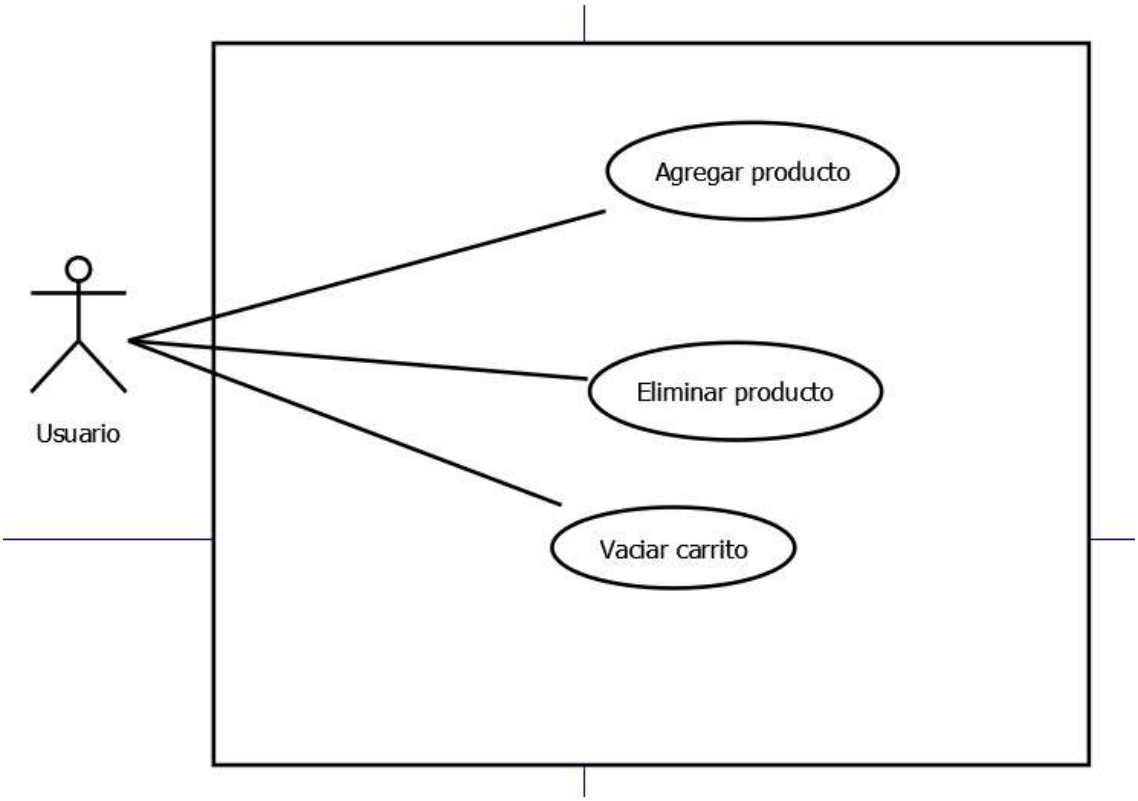


Figura 35. Diagrama de casos de uso de implementar carrito de compras (Chávez, 2022)

Desarrollo:

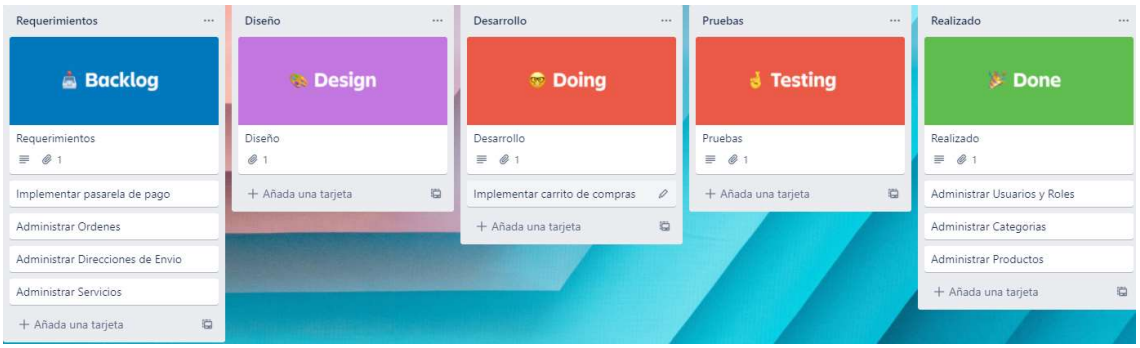


Figura 36. Tablero Kanban – Desarrollo de implementar carrito de compras (Chávez, 2022)

Para implementar el carrito de compras se va a instalar el paquete "*shoppingcart*", una vez se ha instalado el paquete se pueden agregar productos dentro del carrito. Al darle clic en el icono superior con forma de carrito se mostrarán los productos seleccionados y datos adicionales como la cantidad, el precio individual y el total.

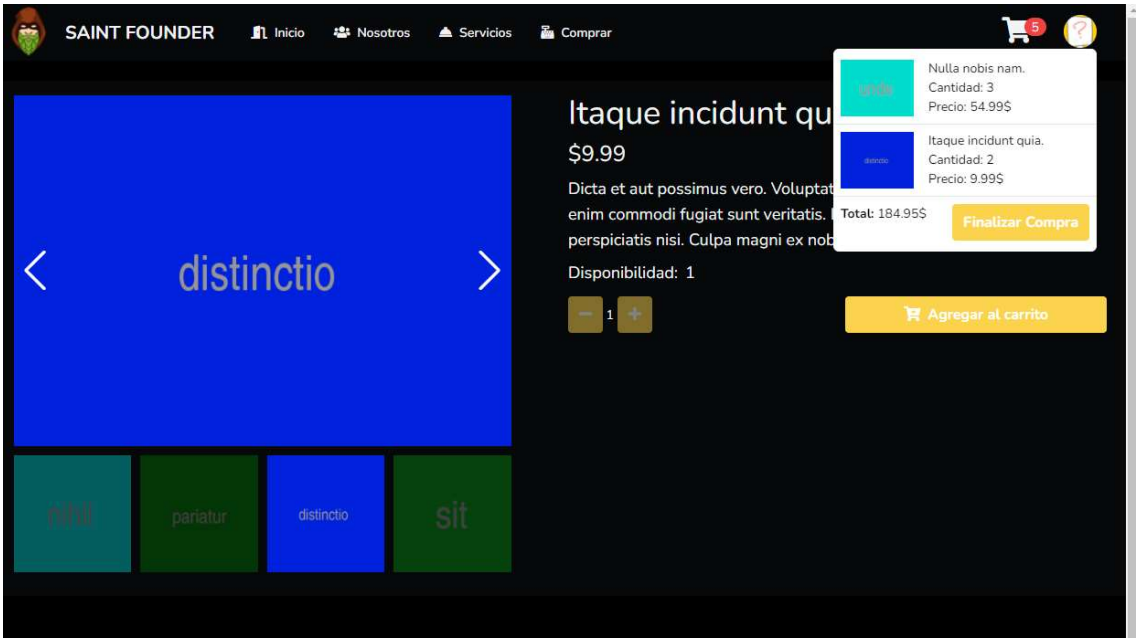


Figura 37. Carrito de compras pequeño con productos dentro de este (Chávez, 2022)

Si se le da clic en el botón de finalizar compra nos redirecciona a otra vista en el cual se ve el carrito de una manera más ordenada, además de que nos permite eliminar algún producto, cambiar la cantidad de los productos o de por si vaciar el carrito de compras.



Figura 38. Carrito de compras con productos dentro de este (Chávez, 2022)

En caso de que se seleccione el continuar con la compra redireccionara a una sección donde se puede ingresar los datos que serán necesarios para generar una orden de compra.

Pruebas:

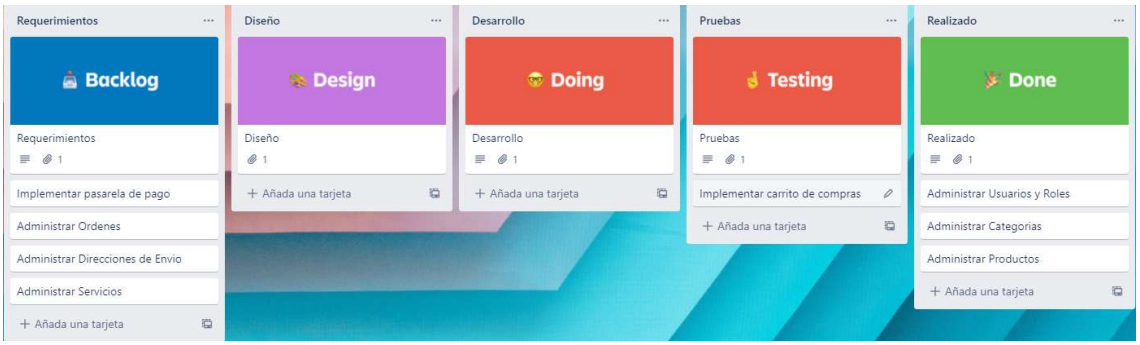


Figura 39. Tablero Kanban – Pruebas de implementar carrito de compras (Chávez, 2022)

Escenarios	Casos de prueba	Estado
Agregar producto	Se puede agregar un producto	Funcional
	Se puede agregar más de un producto	Funcional
Eliminar producto	Se puede eliminar un producto al carrito	Funcional
Vaciar carrito	Se puede eliminar todos los productos del carrito	Funcional

Tabla 6. Casos de prueba de implementar carrito de compras (Chávez, 2022)

3.6 Administrar Ordenes

Diseño:

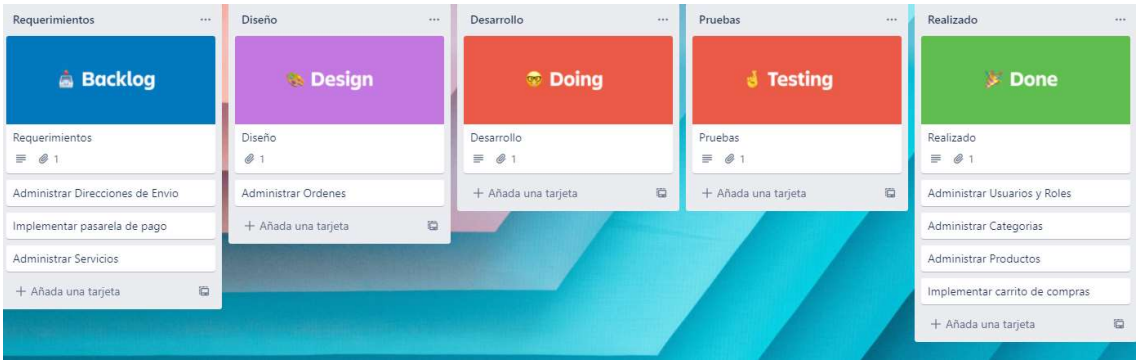


Figura 40. Tablero Kanban – Diseño de administrar órdenes (Chávez, 2022)

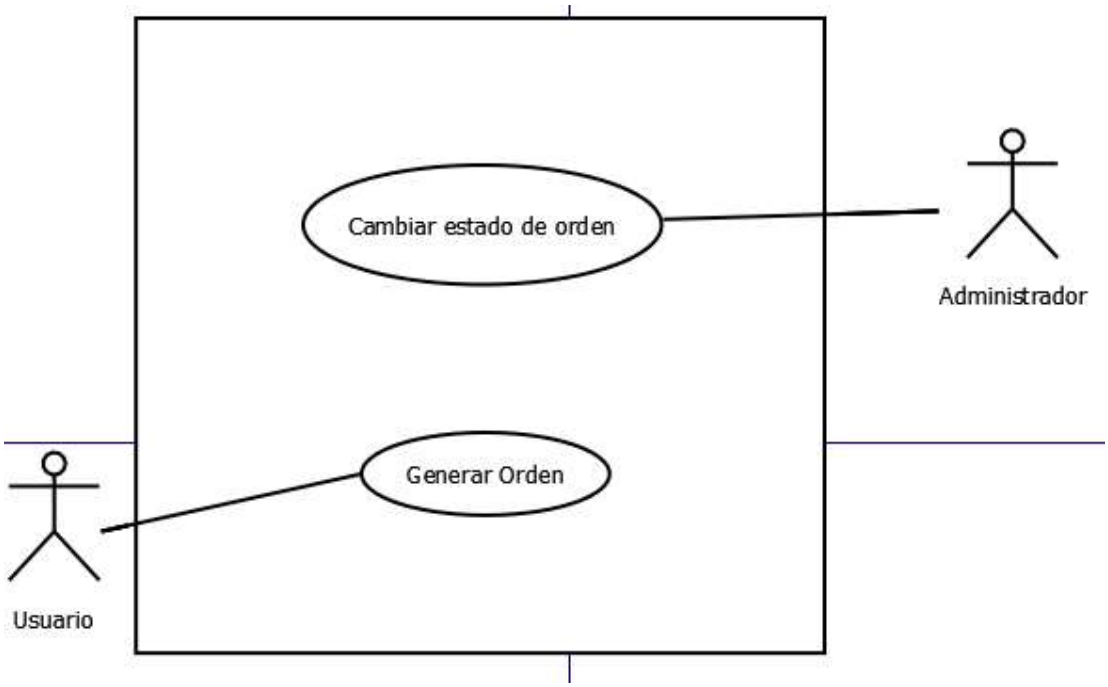


Figura 41. Diagrama de casos de uso de administrar órdenes (Chávez, 2022)

Desarrollo:

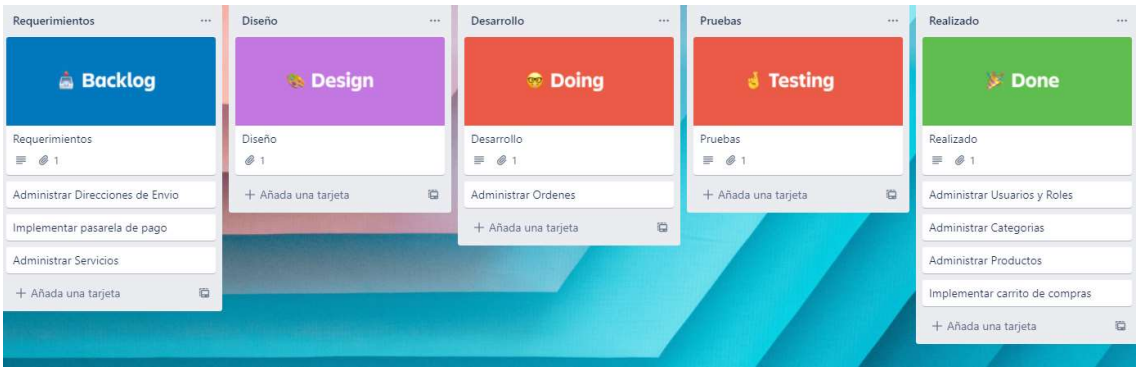


Figura 42. Tablero Kanban – Desarrollo de administrar órdenes (Chávez, 2022)

Se crea un modelo y una migración para las ordenes la cual tiene como campos, el nombre de contacto, el teléfono, los productos asociados a esta, el tipo de envío el cual puede ser retiro en tienda o envío a domicilio, el estado los cuales pueden ser pendiente, pagado, enviado, entregado o anulado, el total a pagar y de manera opcional en caso de seleccionar envío a domicilio se agregan los campos de provincia, ciudad, dirección de envío y referencia.

Para la administración de órdenes existen dos apartados los cuales son una vista para los administradores y una vista para los usuarios normales.

Cuando un usuario desea crear una nueva orden, primero debe poner sus datos de contacto los cuales son el nombre del contacto y teléfono de contacto. Después de eso puede seleccionar si desea acercarse personalmente a retirar el producto en la tienda o que este le sea enviado al sitio de su preferencia.

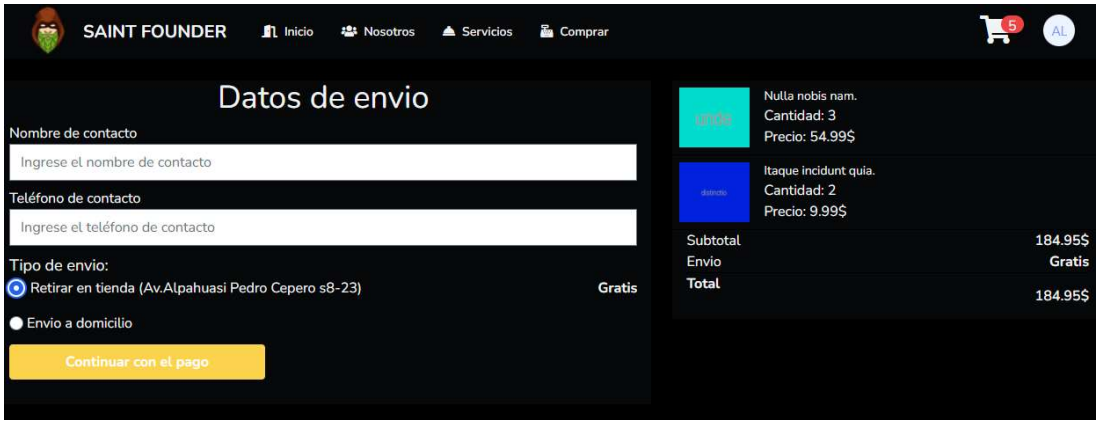


Figura 43 Generar Ordenes con retiro en tienda (Chávez, 2022)

En caso de seleccionar envío a domicilio, la aplicación despliega un menú para que se puedan dar los datos necesarios para realizar un envío como son la provincia, la ciudad, la dirección de envío y las referencias. Al valor del total a pagar se le agrega el valor del envío.

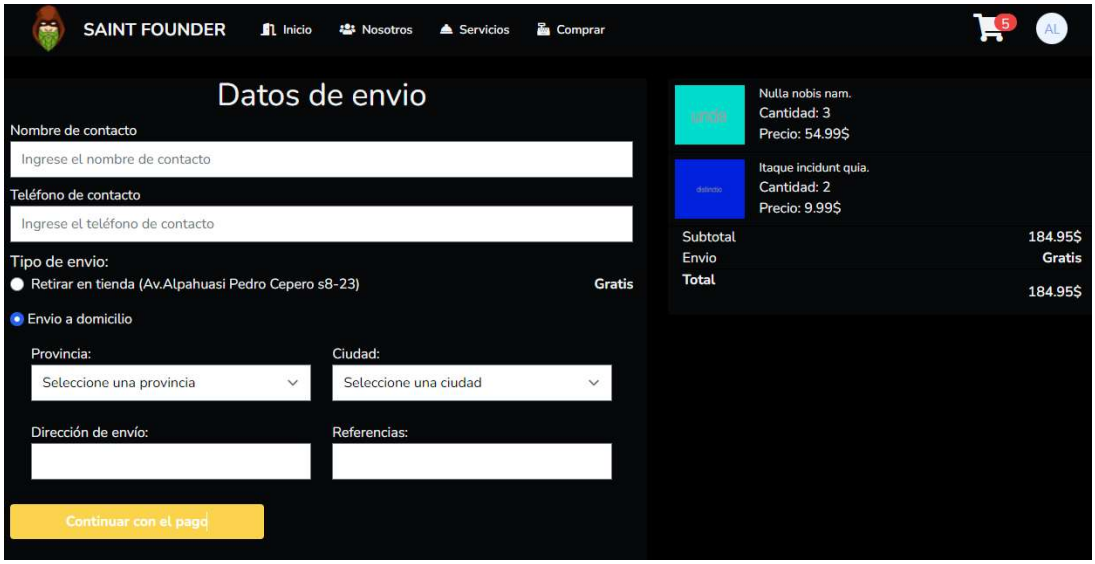


Figura 44. Generar Ordenes con envío a domicilio (Chávez, 2022)

Una vez se crea una orden esta tiene como estado pendiente, se mostrarán los datos de envió, y el detalle de los productos comprados. En caso de que sea actualizado el estado de la orden se reflejara de manera colorida para el usuario.

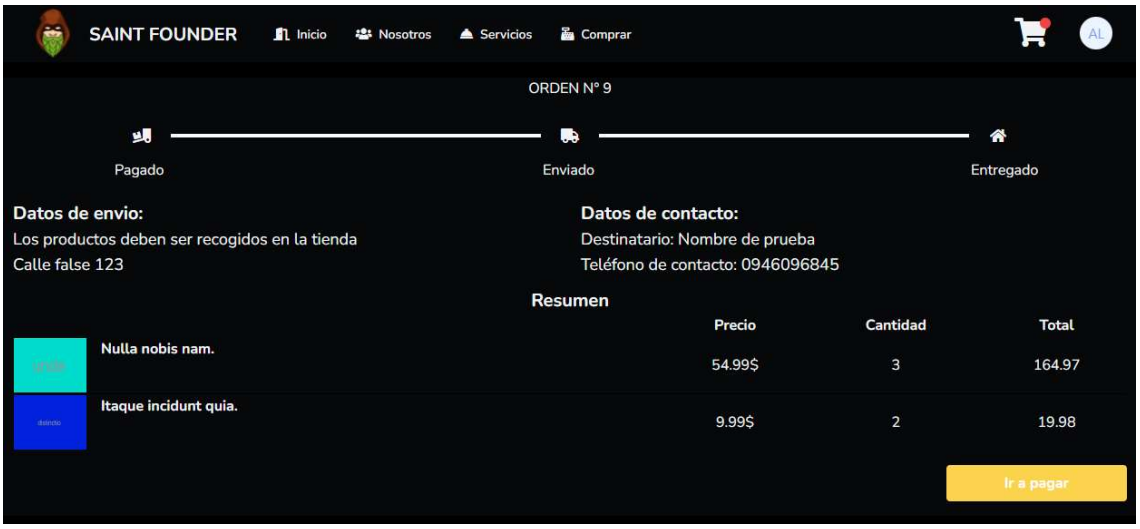


Figura 45. Visualización de orden generada (Chávez, 2022)

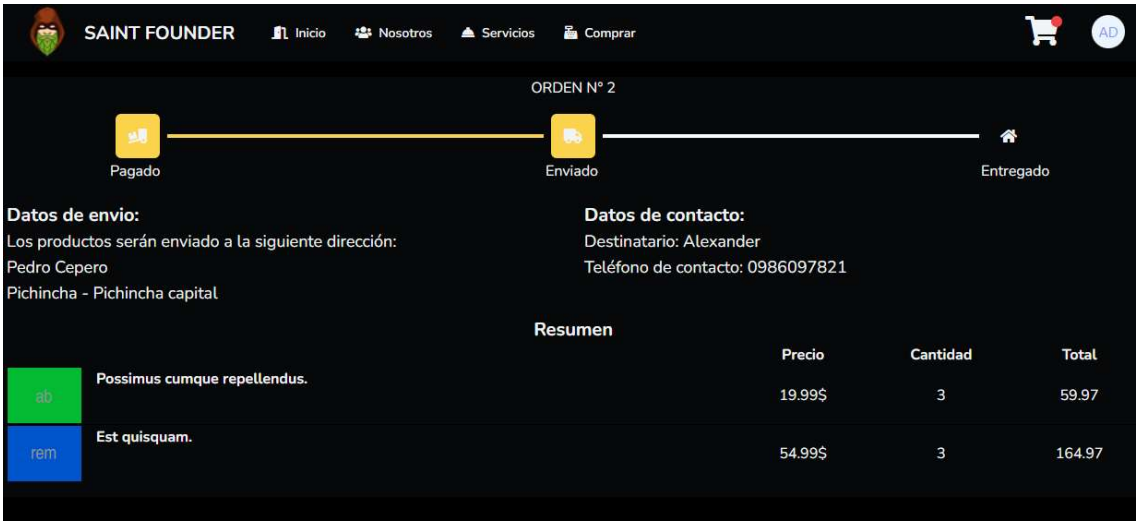


Figura 46. Visualización de orden generada con estado cambiado (Chávez, 2022)

Cuando una orden se encuentre como estado pendiente y no ha sido pagada, se dará un recordatorio en la parte superior de la aplicación. Cuando el usuario seleccione “Ir a pagar” se redireccionará a una página en la cual se pueden visualizar todas la ordenes generadas por el usuario.

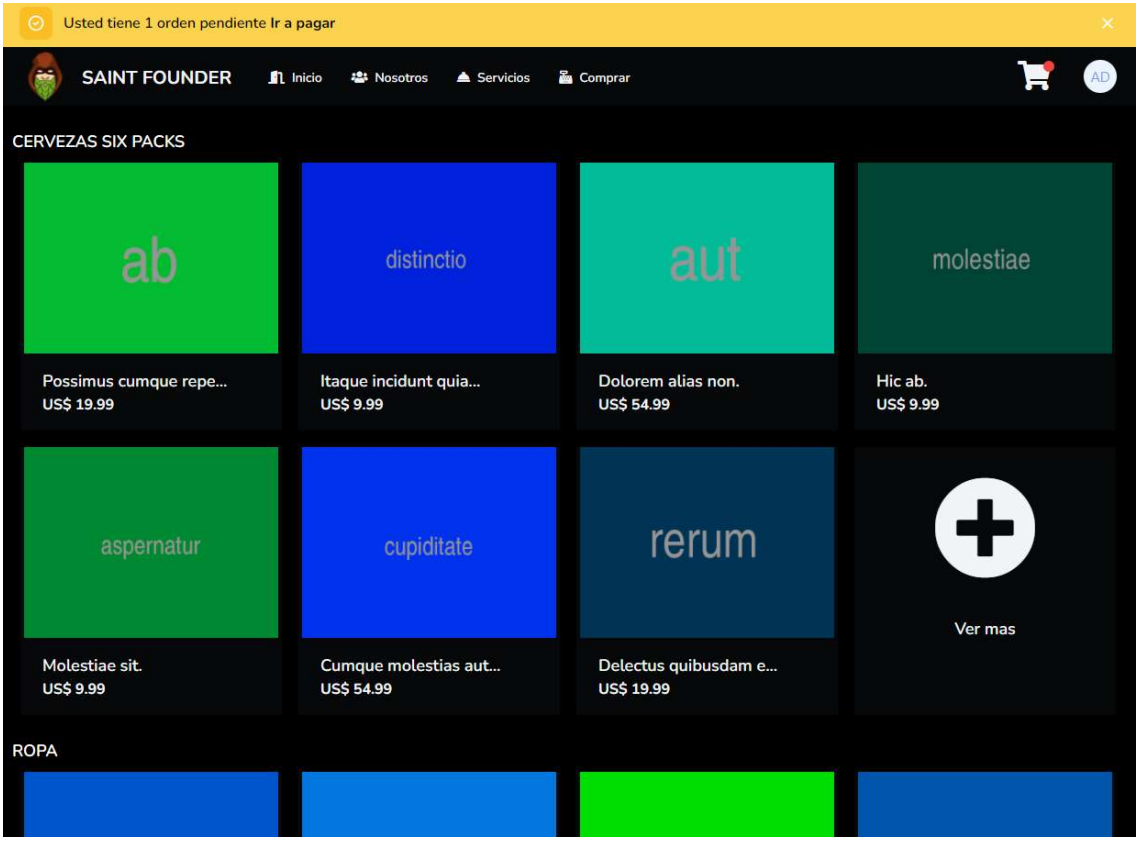


Figura 47. Recordatorio de parte de la aplicación de que una orden no ha sido pagada (Chávez, 2022)



Figura 48. Visualización de la ordenes que ha generado el usuario (Chávez, 2022)

Para la parte administrativa de la aplicación se ha creado una lista similar con la diferencia que los administradores pueden ver todas las ordenes de todos los usuarios.



Figura 49. Visualización de las órdenes que se han generado en la aplicación (Chávez, 2022)

En caso de seleccionarse en el icono amarillo de cada orden se redireccionará a una página en la cual se puede visualizar la orden, con sus datos respectivos, y con la posibilidad de poder cambiar el estado de la orden según conveniencia.

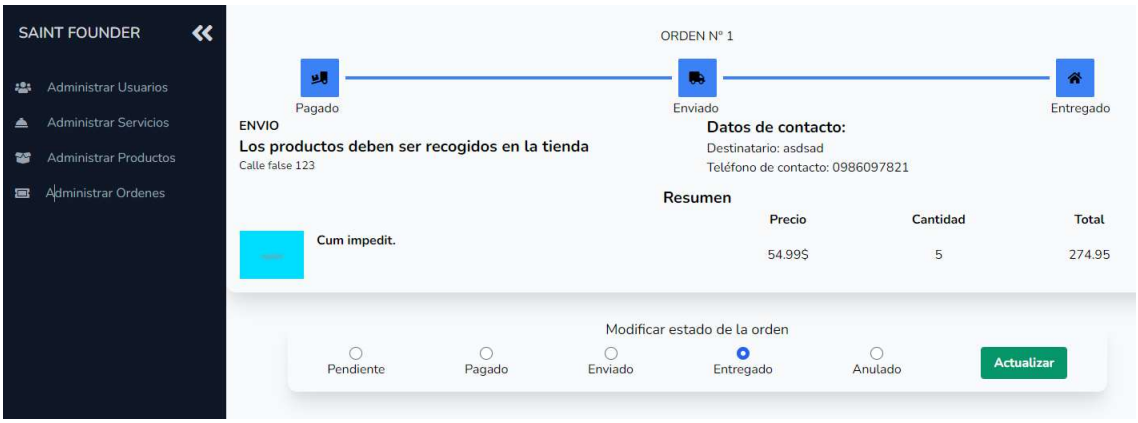


Figura 50. Visualización de una orden y formulario para cambiar el estado de la orden (Chávez, 2022)

Pruebas:

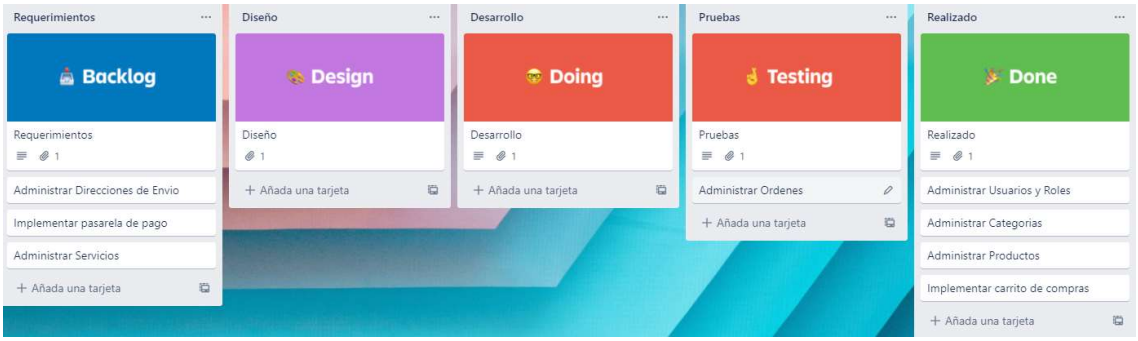


Figura 51. Tablero Kanban – Pruebas administrar órdenes (Chávez, 2022)

Escenarios	Casos de prueba	Estado
Cambiar estado de la orden	Se puede cambiar el estado de la orden	Funcional
Generar Orden	Se puede generar ordenes	Funcional

Tabla 7. Casos de prueba de administrar órdenes (Chávez, 2022)

3.7 Implementar pasarela de pago

Diseño:

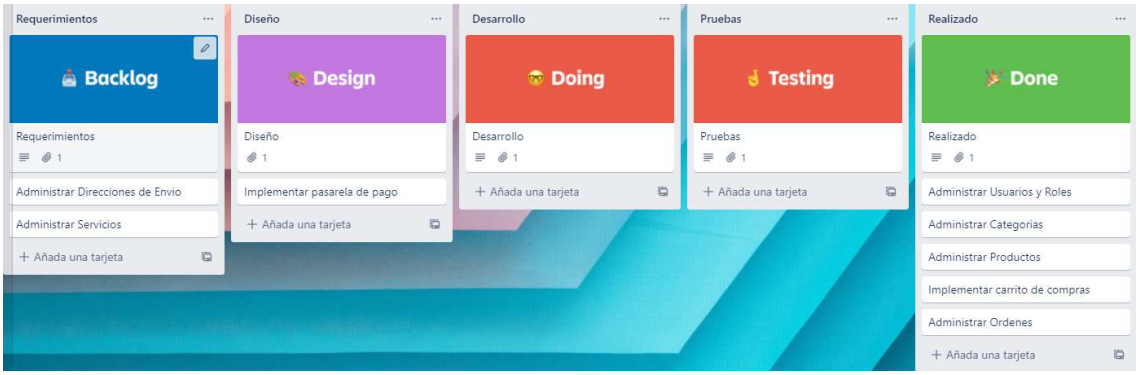


Figura 52. Tablero Kanban – Diseño implementar pasarela de pago (Chávez, 2022)

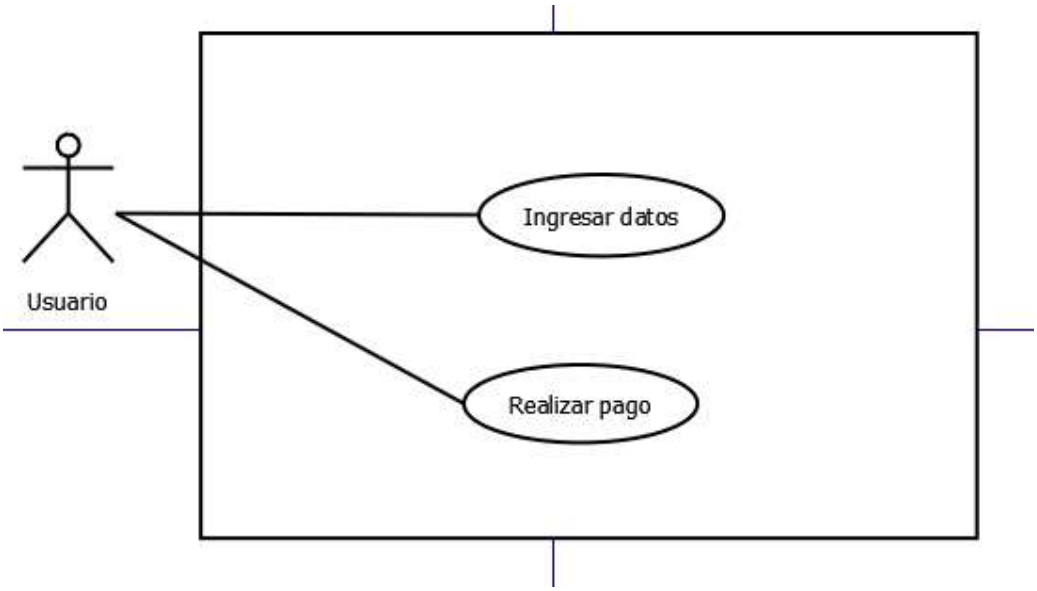


Figura 53. Diagrama de casos de uso de implementar pasarela de pago (Chávez, 2022)

Desarrollo:

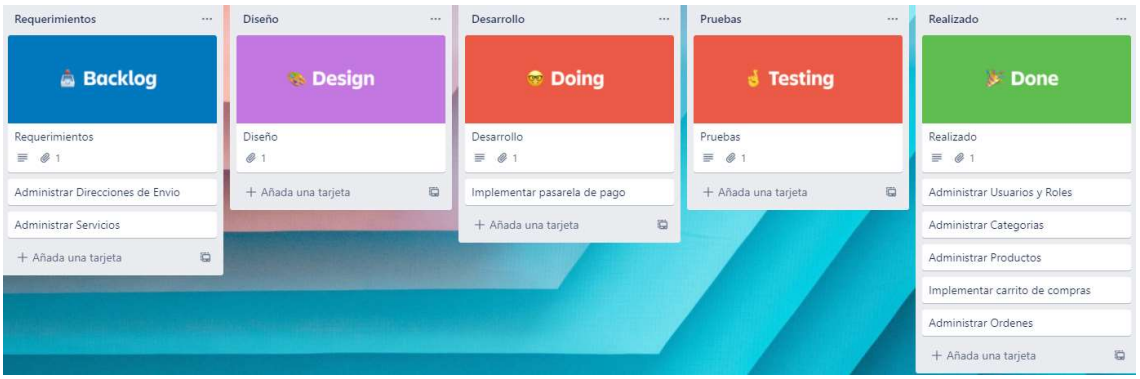


Figura 54. Tablero Kanban – Desarrollo implementar pasarela de pago (Chávez, 2022)

Para implementar la pasarela de pago “Paypal” es necesario primero tener una cuenta registrada y a través del apartado de desarrolladores obtener las credenciales que permitan conectar la aplicación web con “Paypal”. Para desarrollar y realizar pruebas “Paypal” ofrece un entorno de desarrollo denominado como “*sandbox*” en el cual se puede simular pagos.

PayPal también provee de un botón de pago el cual recibe datos desde la aplicación como el precio total a pagar, procesa este pago, y en caso de no existir algún problema envía una notificación a la aplicación informando que efectivamente se ha realizado el pago.

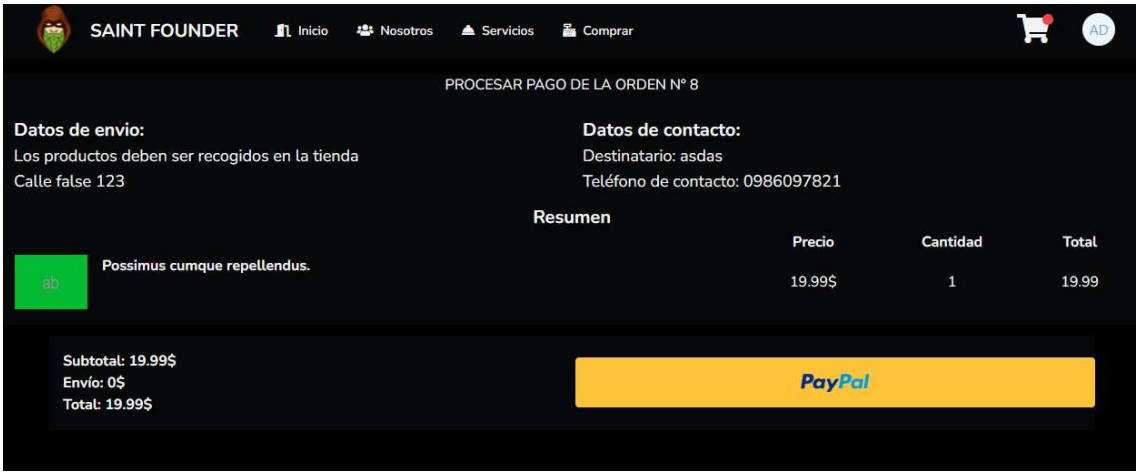


Figura 55. Implementación del botón de PayPal (Chávez, 2022)

Cuando se da clic en el botón de PayPal se despliega un menú en la pantalla donde será necesario introducir credenciales validas.

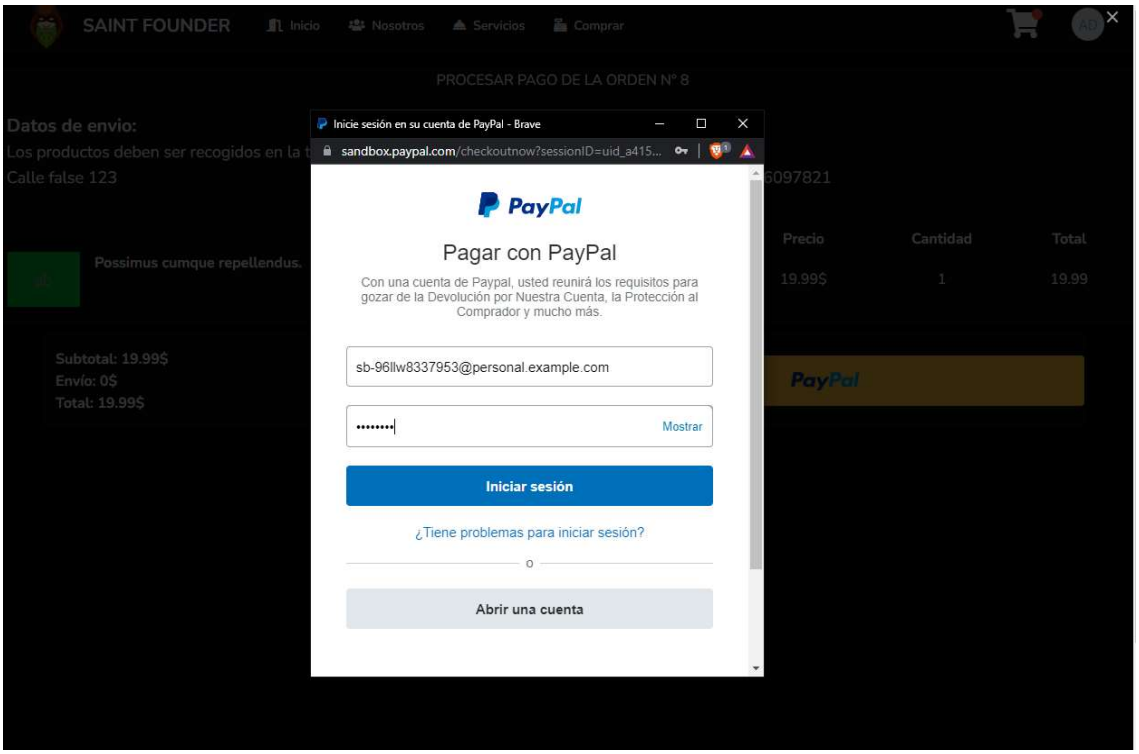


Figura 56. Funcionamiento del botón de PayPal (Chávez, 2022)

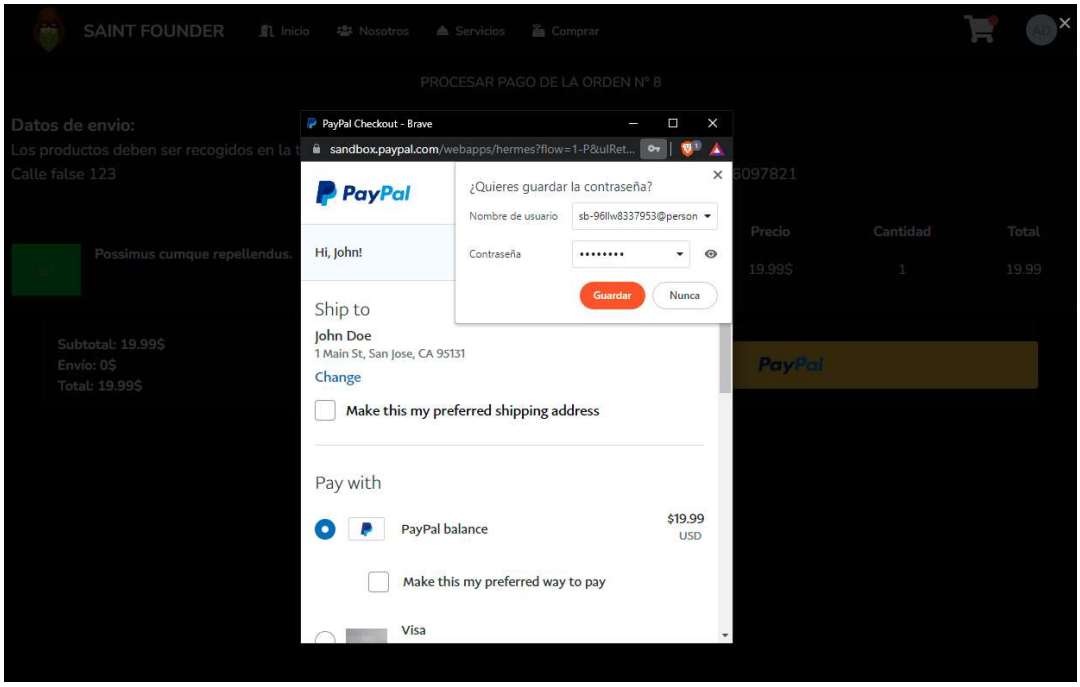


Figura 57. Continuación del funcionamiento del botón de PayPal (Chávez, 2022)

Una vez se realiza el pago, la aplicación recibe una confirmación de la transacción a través de paypal y redirecciona hacia la orden cambiando el estado de la orden de pendiente a pagado.

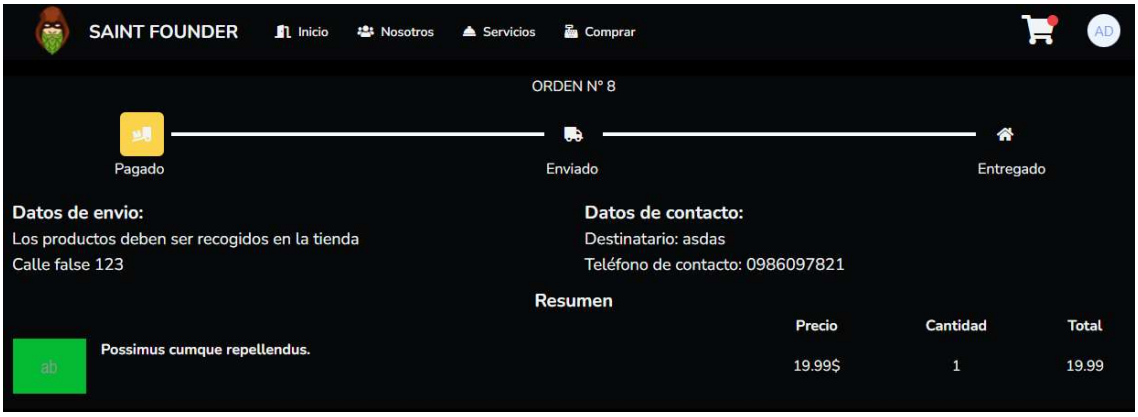


Figura 58. Vista de orden tras confirmación del botón de PayPal (Chávez, 2022)

Pruebas:

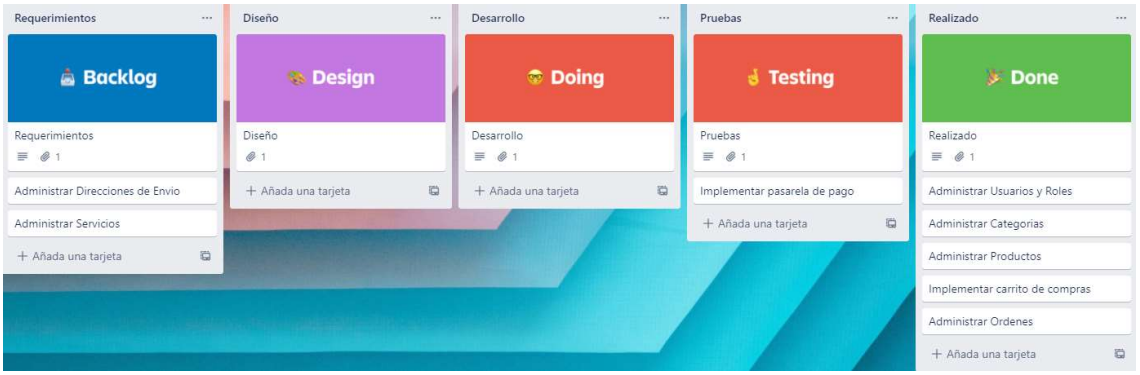


Figura 59. Tablero Kanban – Pruebas implementar pasarela de pago (Chávez, 2022)

Escenarios	Casos de prueba	Estado
Ingresar datos	Se pueden ingresar los datos del usuario	Funcional
	Se validan los datos del usuario	Funcional
Realizar pago	El usuario puede realizar pagos	Funcional

Tabla 8. Casos de prueba implementar pasarela de pago (Chávez, 2022)

3.8 Administrar direcciones de envió.

Diseño:

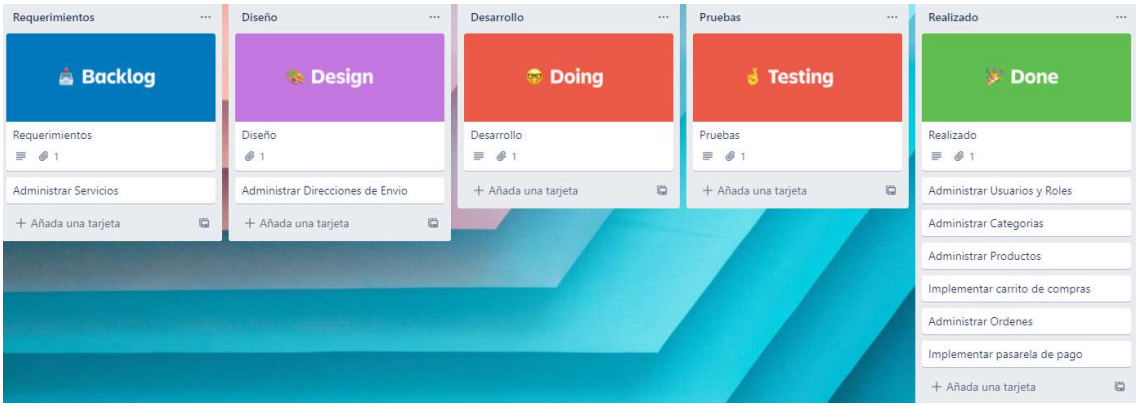


Figura 60. Tablero Kanban – Diseño administrar ordenes de envío (Chávez, 2022)

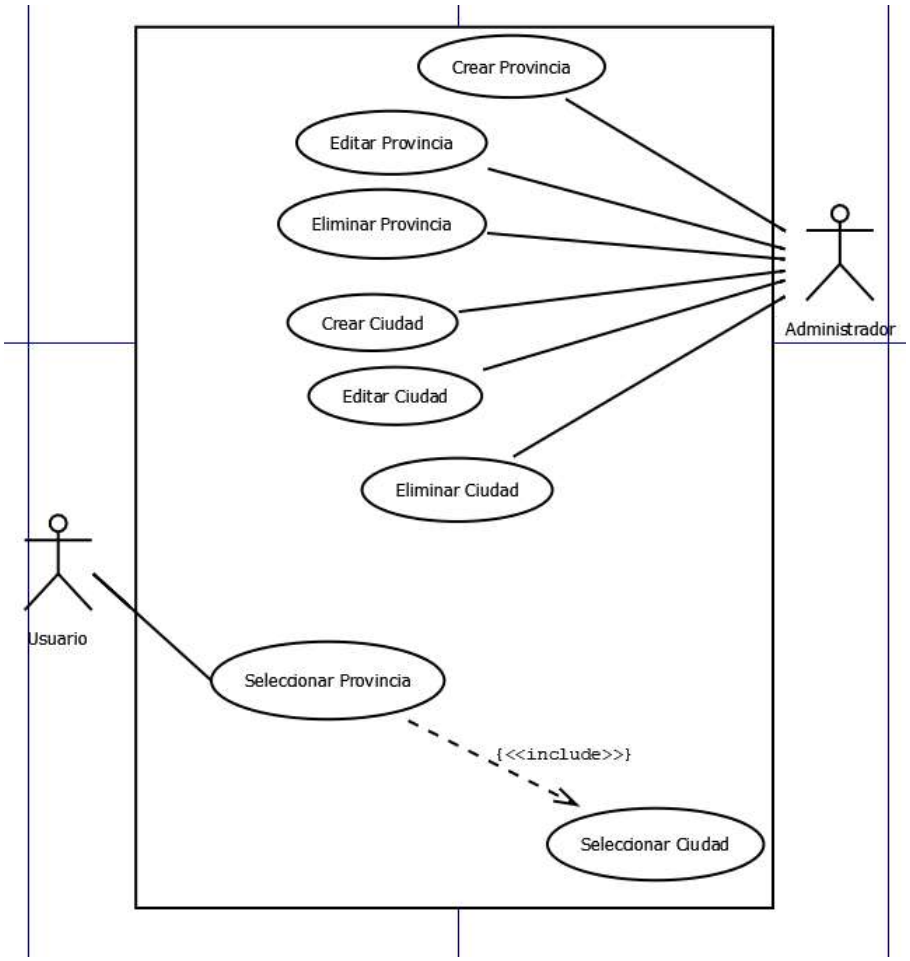


Figura 61. Diagrama de casos de uso de administrar direcciones de envío (Chávez, 2022)

Desarrollo:

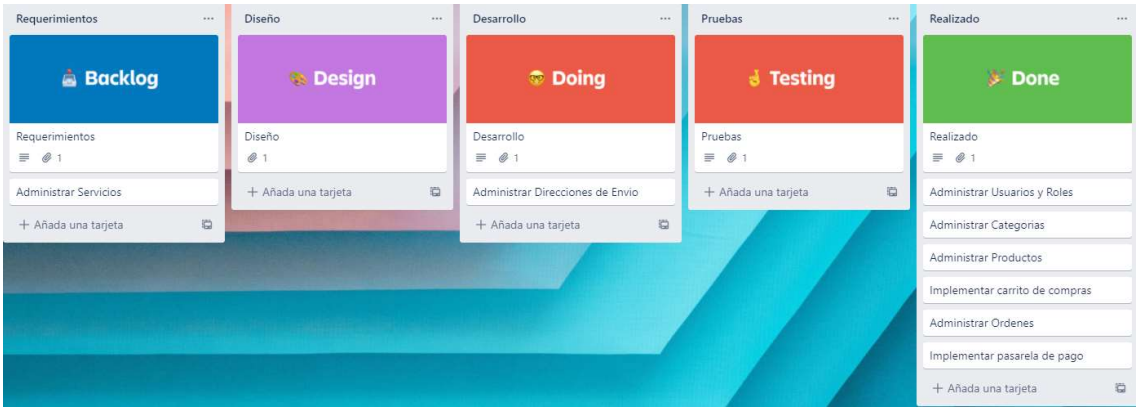


Figura 62 Tablero Kanban – Desarrollo administrar direcciones de envío (Chávez, 2022)

Las ordenes necesitan de una dirección de envió en caso de que el usuario así lo necesite, hasta ahora se han realizado las pruebas de las ordenes con una única dirección de envío que fue previamente cargada en la base de datos, pero es necesario que los administradores de la página puedan agregar más direcciones de envío y que cada una de estas tengan un precio el cual se sume a la creación de las órdenes.

Las direcciones de envió se componen de provincias, cuidades y precio, para que se pueda administrar de una manera más cómoda se ha creado una vista en la cual se puede agregar una provincia, y abajo una tabla con una lista de todas las provincias que existen en la aplicación.



Figura 63. Vista de administración de provincias (Chávez, 2022)

Si se da clic sobre el botón rojo sale un mensaje de confirmación para verificar que enserio se desea eliminar esa provincia, en caso de eliminar una provincia también se eliminan todas las ciudades asociadas a esta.



Figura 64. Confirmación de eliminar provincia (Chávez, 2022)

Si se da clic sobre el botón verde la tabla nos redirecciona a una sección en la cual se puede editar el nombre de la provincia y al mismo tiempo se ve una tabla con todas las ciudades asociadas a esta provincia.



Figura 65. Vista para edición de provincias (Chávez, 2022)

Si se da clic sobre el botón de agregar ciudad se despliega en pantalla un formulario en el cual se pueden llenar los campos de nombre y precio.

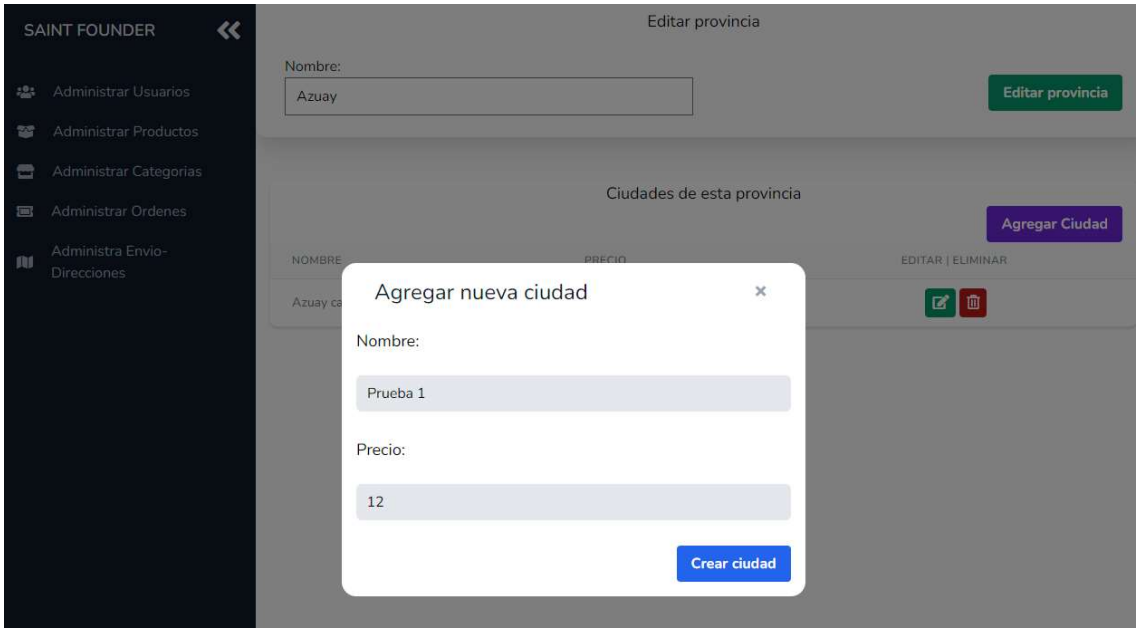


Figura 66. Vista para agregar nueva ciudad (Chávez, 2022)

En caso de que se desee editar la ciudad es necesario dar clic sobre el botón verde, y se desplegara un formulario en el cual se pueden modificar los campos.

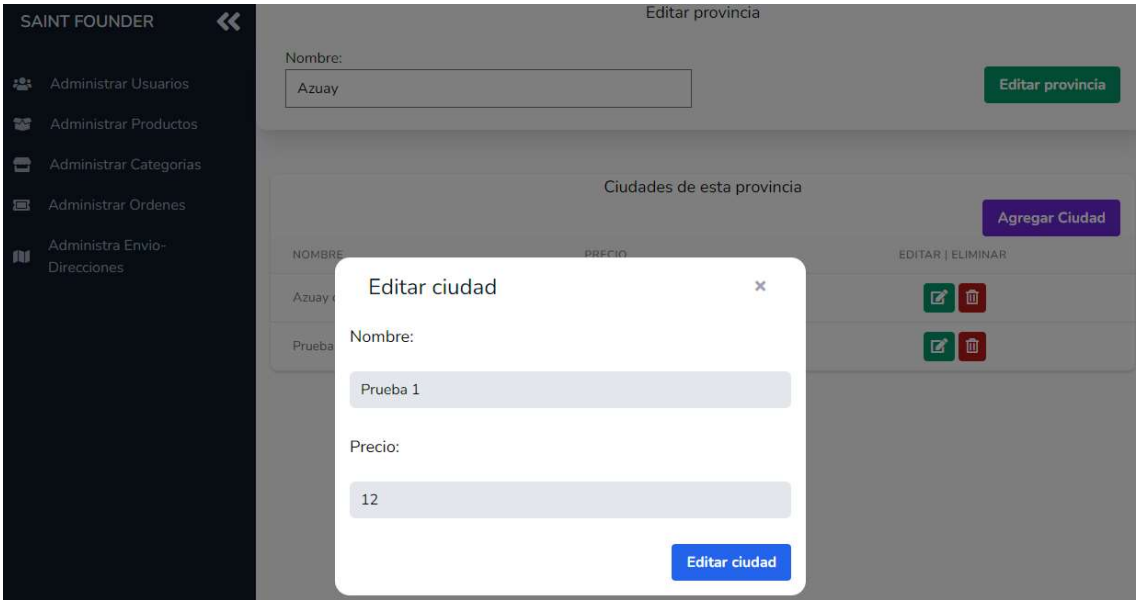


Figura 67. Vista para editar ciudad (Chávez, 2022)

Cuando sea necesario eliminar una ciudad se debe dar clic sobre el botón rojo sobre la tabla y se abrirá una confirmación preguntando si en verdad se desea eliminar aquella ciudad.



Figura 68. Confirmación para eliminar una ciudad (Chávez, 2022)

Pruebas:

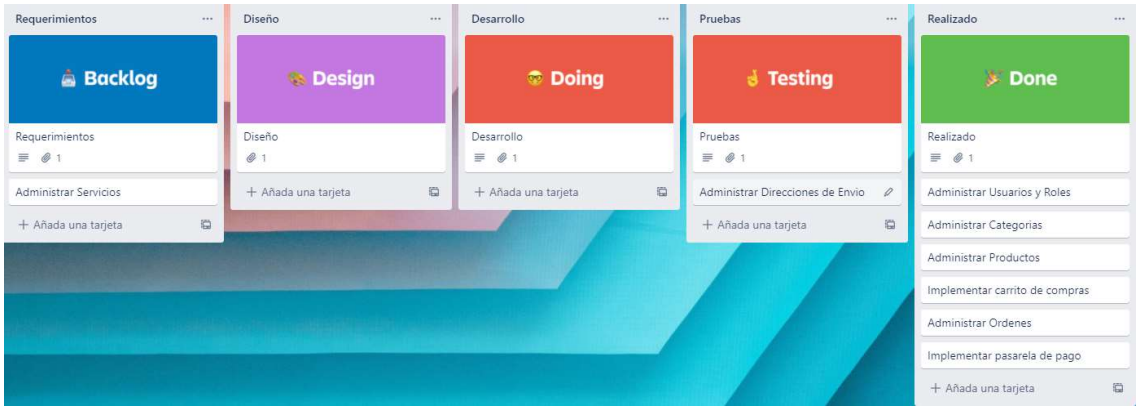


Figura 69. Pruebas Administrar Direcciones de envío (Chávez, 2022)

Escenarios	Casos de prueba	Estado
Crear Provincia	Se puede crear una provincia	Funcional
Editar Provincia	Se puede editar una provincia	Funcional
Eliminar Provincia	Se presenta una confirmación en pantalla	Funcional
	Se elimina la provincia y las ciudades asociadas a esta	Funcional
Crear Ciudad	Se puede crear una ciudad	Funcional
Editar Ciudad	Se puede editar una ciudad	Funcional
Eliminar Ciudad	Se presenta una confirmación en pantalla	Funcional
	Se elimina la ciudad	Funcional
Seleccionar Provincia	El usuario puede seleccionar una provincia	Funcional
Seleccionar Ciudad	El usuario puede seleccionar una ciudad correspondiente a la provincia seleccionada	Funcional

Tabla 9. Casos de prueba de administrar direcciones de envío (Chávez, 2022)

3.9 Administrar servicios.

Diseño:

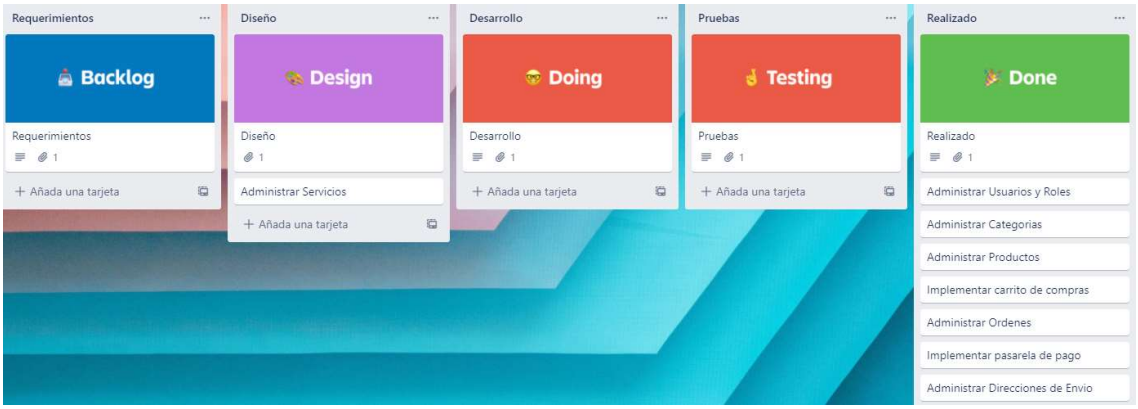


Figura 70. Tablero Kanban – Diseño administrar servicios (Chávez, 2022)

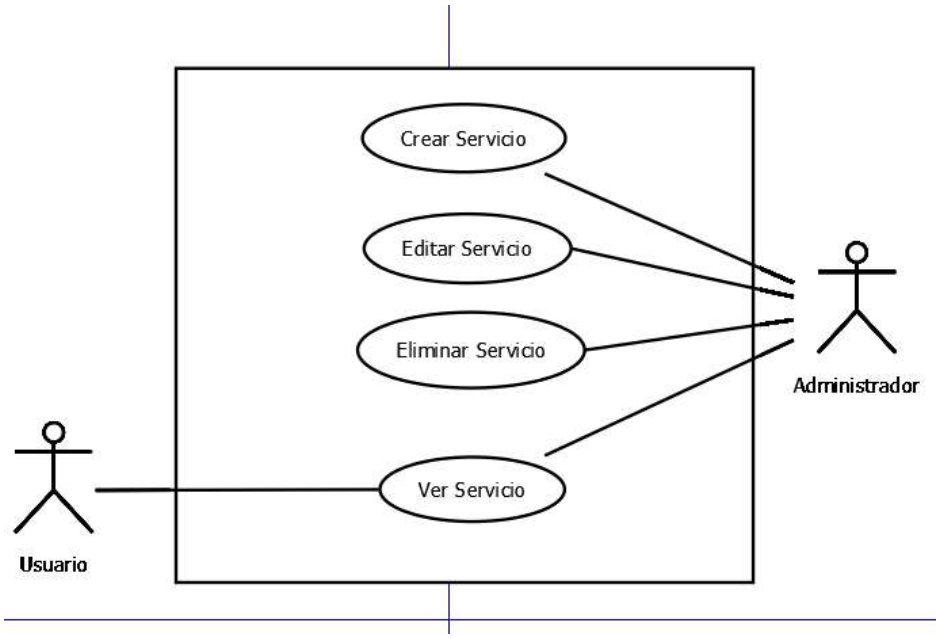


Figura 71. Diagrama de casos de uso de administrar servicios (Chávez, 2022)

Desarrollo:

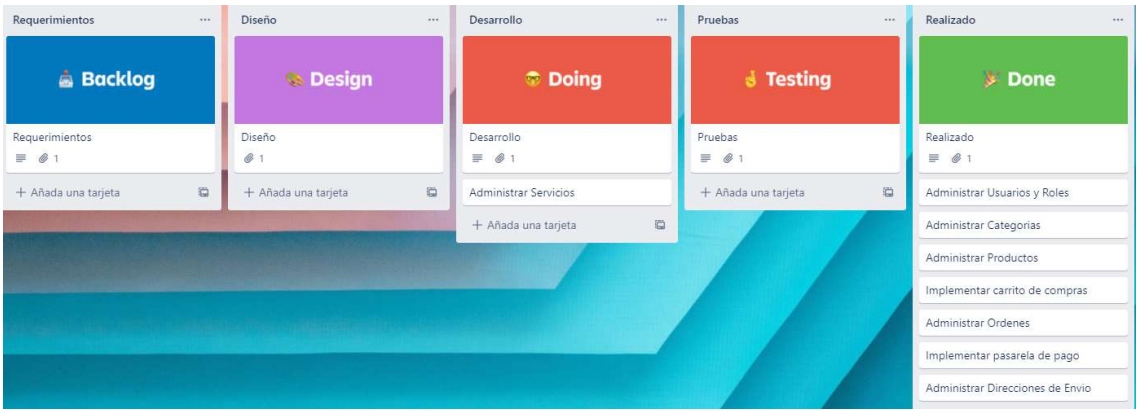


Figura 72. Tablero Kanban – Desarrollo administrar servicios (Chávez, 2022)

La administración de servicios a diferencia de los productos se pretende que sea de carácter informativo sin que el usuario tenga algún tipo de interacción con ellos.

Por lo que para una buena administración se ha realizado una tabla para cada uno de los apartados de servicios, en donde se puede realizar una búsqueda y existe un botón para agregar un nuevo servicio.

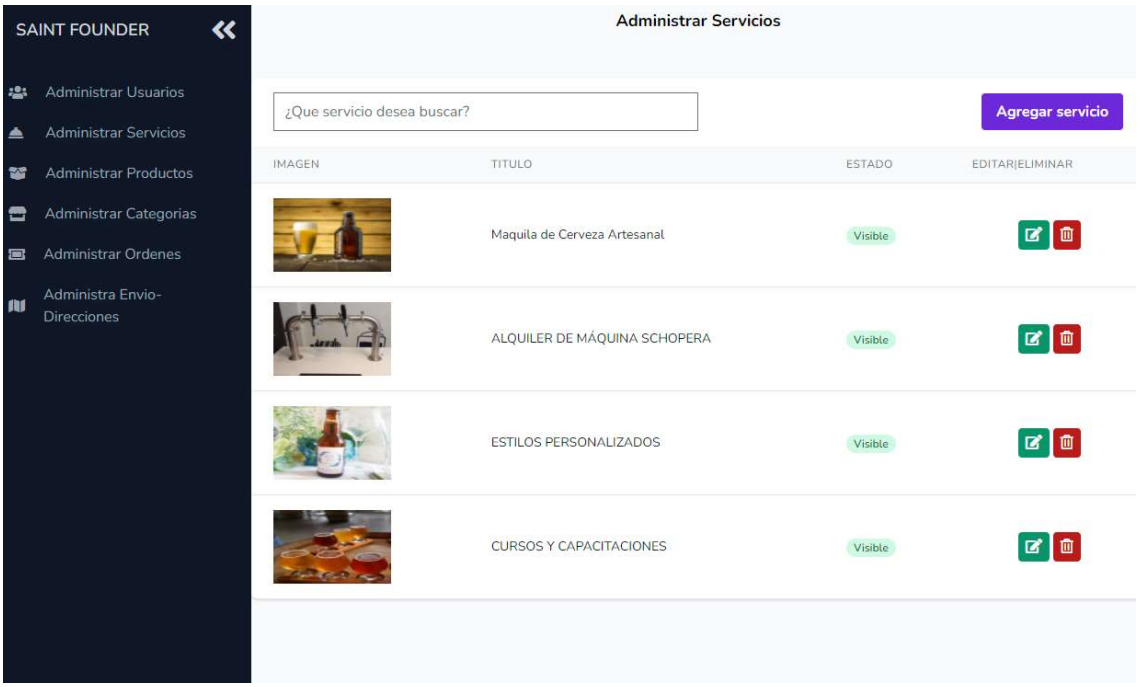


Figura 73. Tabla de administrar servicios (Chávez, 2022)

Si se presiona el botón de agregar un nuevo servicio se muestra en pantalla una vista en la cual se puede ingresar el título, el estado si es visible o está oculto y la descripción.

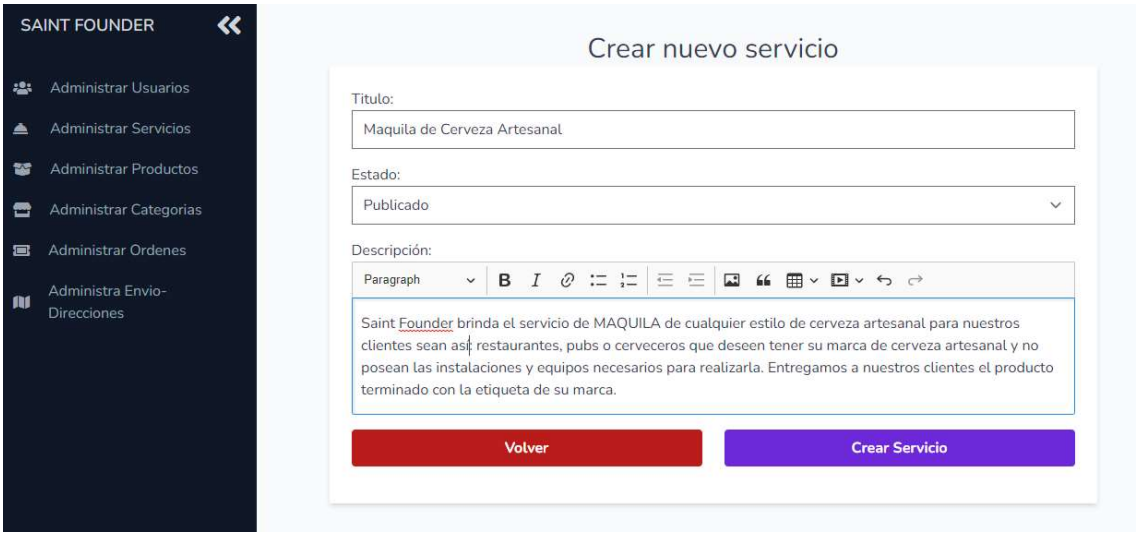


Figura 74. Formulario para crear un servicio nuevo (Chávez, 2022)

En caso de presionar el botón verde en cualquiera de las dos tablas se redirecciona a la parte para editar el servicio/estilo y también se pueden agregar imágenes.

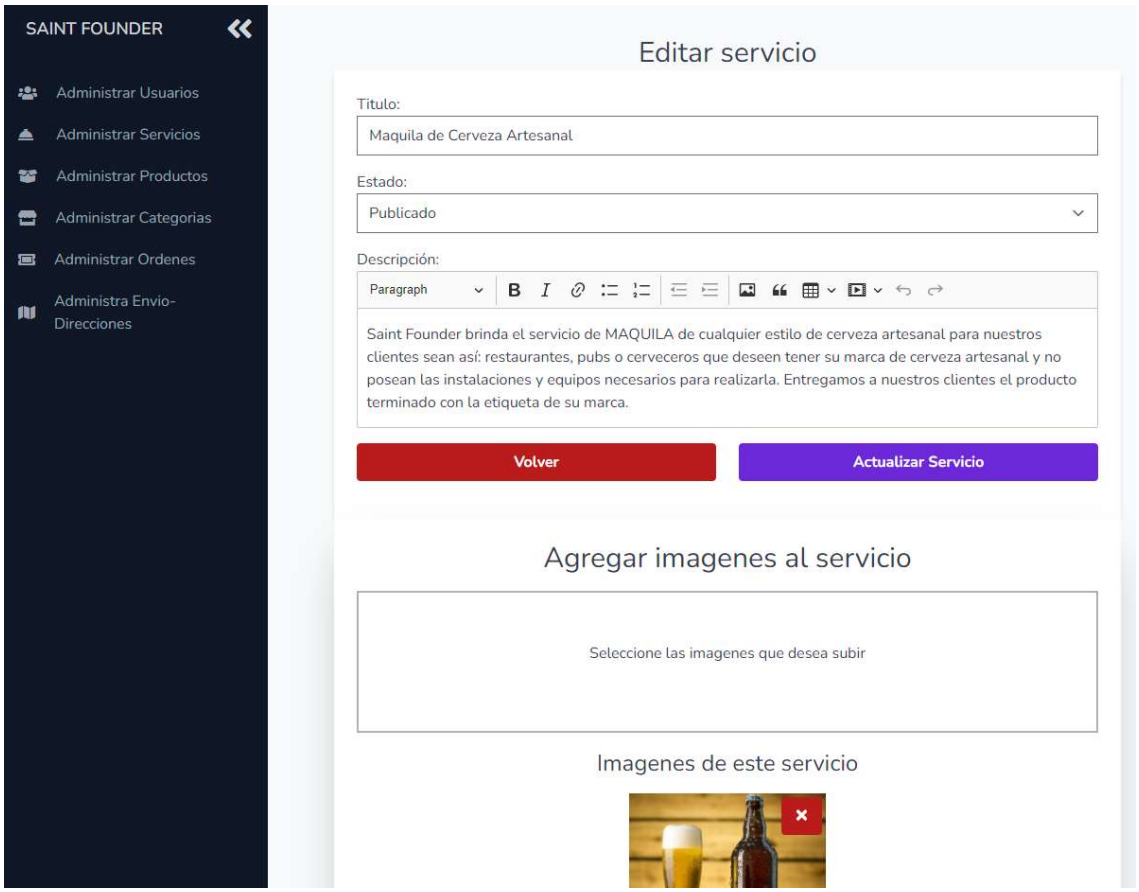


Figura 75. Formulario para editar servicio y agregar imágenes (Chávez, 2022)

En caso de tener la necesidad de eliminar algún servicio esto es posible con solo presionar el botón rojo en cualquiera de las dos tablas. Una vez se presione el botón mostrara en pantalla una confirmación para asegurarse que el administrador de verdad desea eliminar el mencionado servicio.

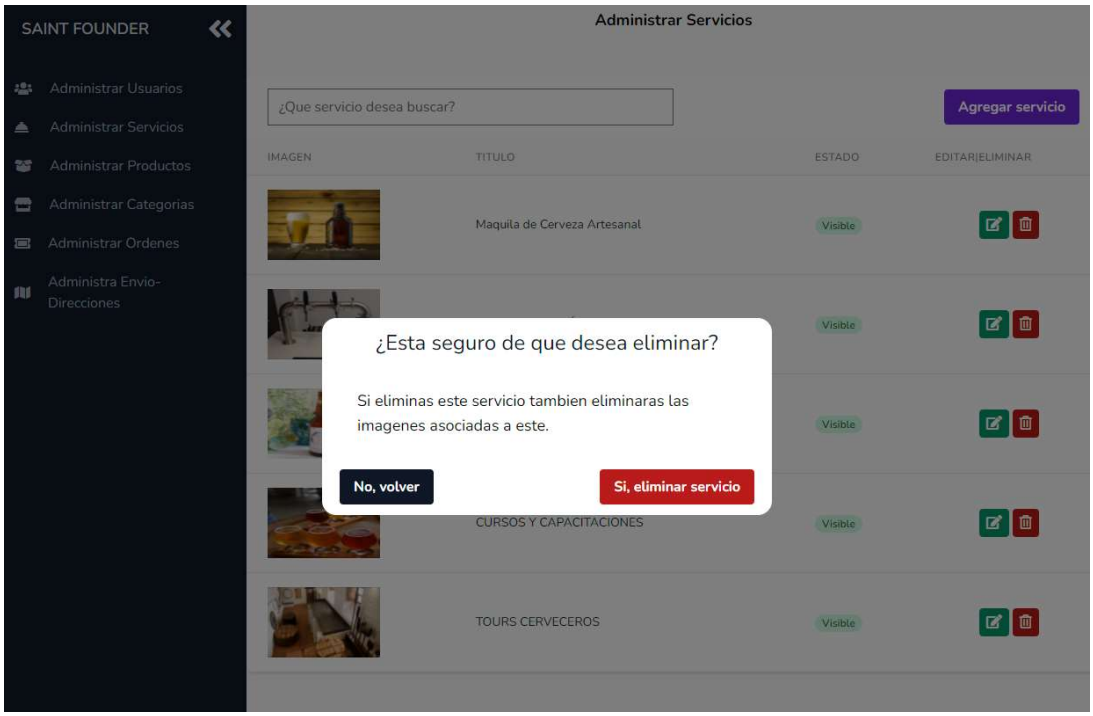


Figura 76. Confirmación para eliminar un servicio (Chávez, 2022)

En cuando a los usuarios que entren en la página web, si estos seleccionan servicios se mostraran todos los servicios.

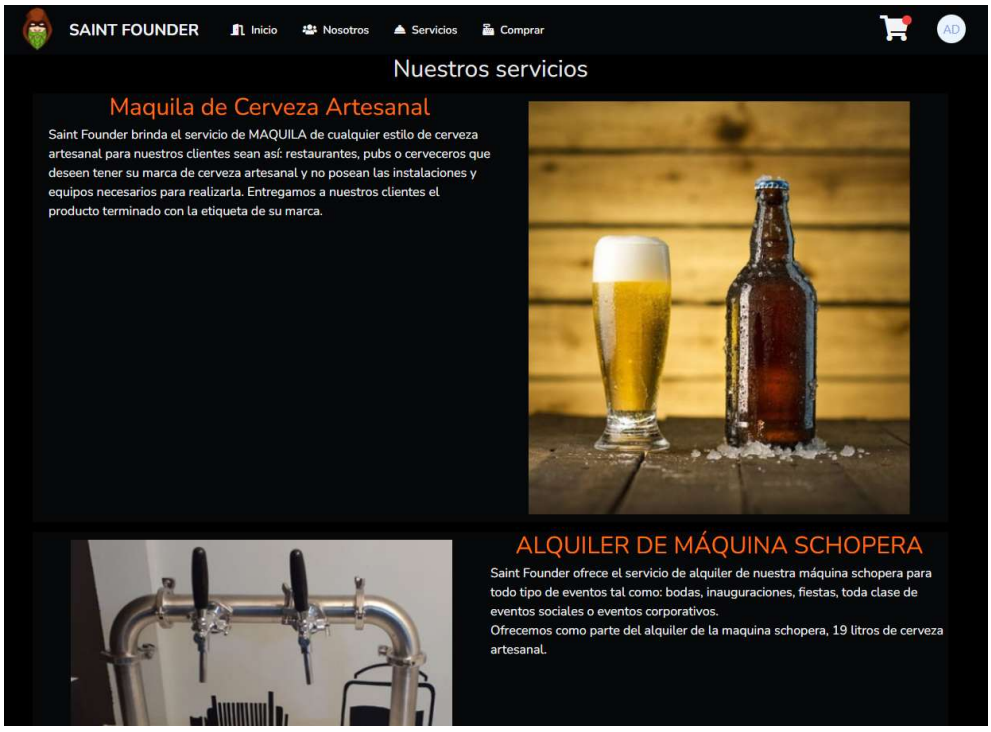


Figura 77- Vista de usuario para servicios (Chávez, 2022)

Pruebas:

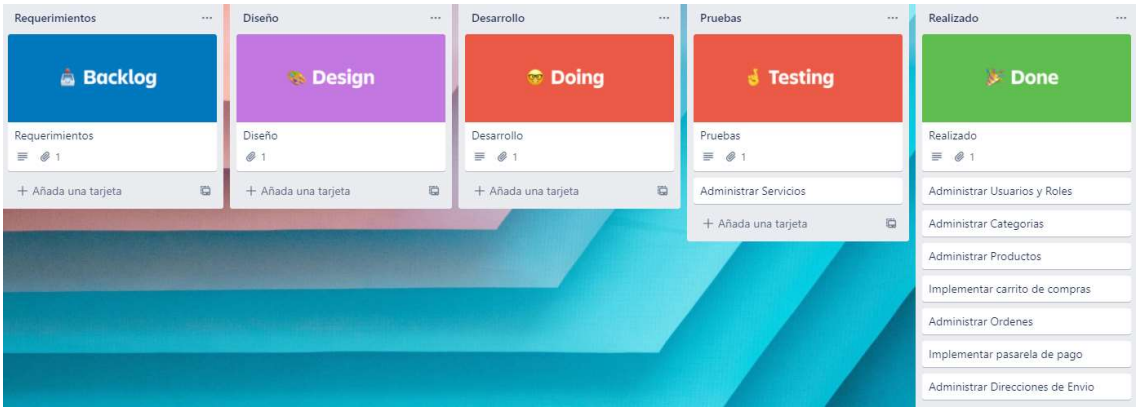


Figura 78. Tablero Kanban – Pruebas administrar servicios (Chávez, 2022)

Escenarios	Casos de prueba	Estado
Crea Servicio	Se puede crear un servicio	Funcional
	Se pueden establecer los parámetros del servicio	Funcional
Editar Servicio	Se puede cambiar el nombre del servicio	Funcional
	Se pueden cambiar los parámetros del servicio	Funcional
Eliminar Servicio	Se presenta una confirmación en pantalla	Funcional
	Se elimina el servicio de la base de datos	Funcional
Ver Servicio	Los usuarios pueden ver los servicios	Funcional

Tabla 10. Casos de prueba de administrar servicios (Chávez, 2022)

Capítulo 4: Subir a producción.

Se va a usar el servicio web de “Laravel Forge” el cual facilita todas las operaciones necesarias para se pueda desplegar una aplicación de Laravel en la web, en este caso dentro de un servidor de “Digital Ocean”.

Cuando se crea un servidor nuevo se deben seleccionar las características que se consideren necesarias para el mismo, como es en este caso se ha ingresado el nombre de este y se ha seleccionado la capacidad del servidor.

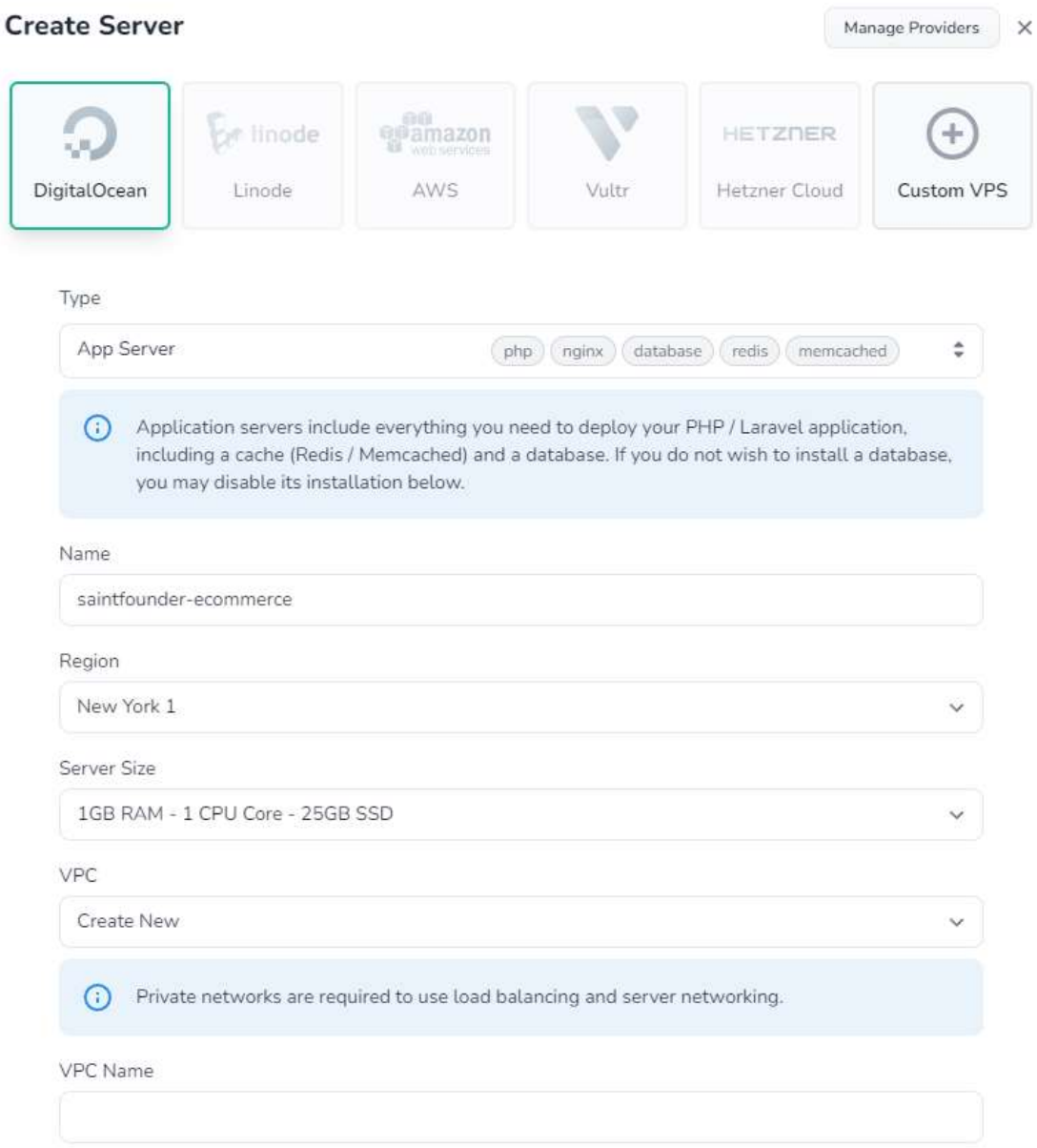


Figura 79. Configuración de servidor de Laravel Forge (Chávez, 2022)

El servicio comenzara a ejecutar las acciones necesarias dentro del servidor para que se puedan ejecutar aplicaciones de Laravel dentro del mismo, una vez finalizado se mostrara el servidor con sus características.

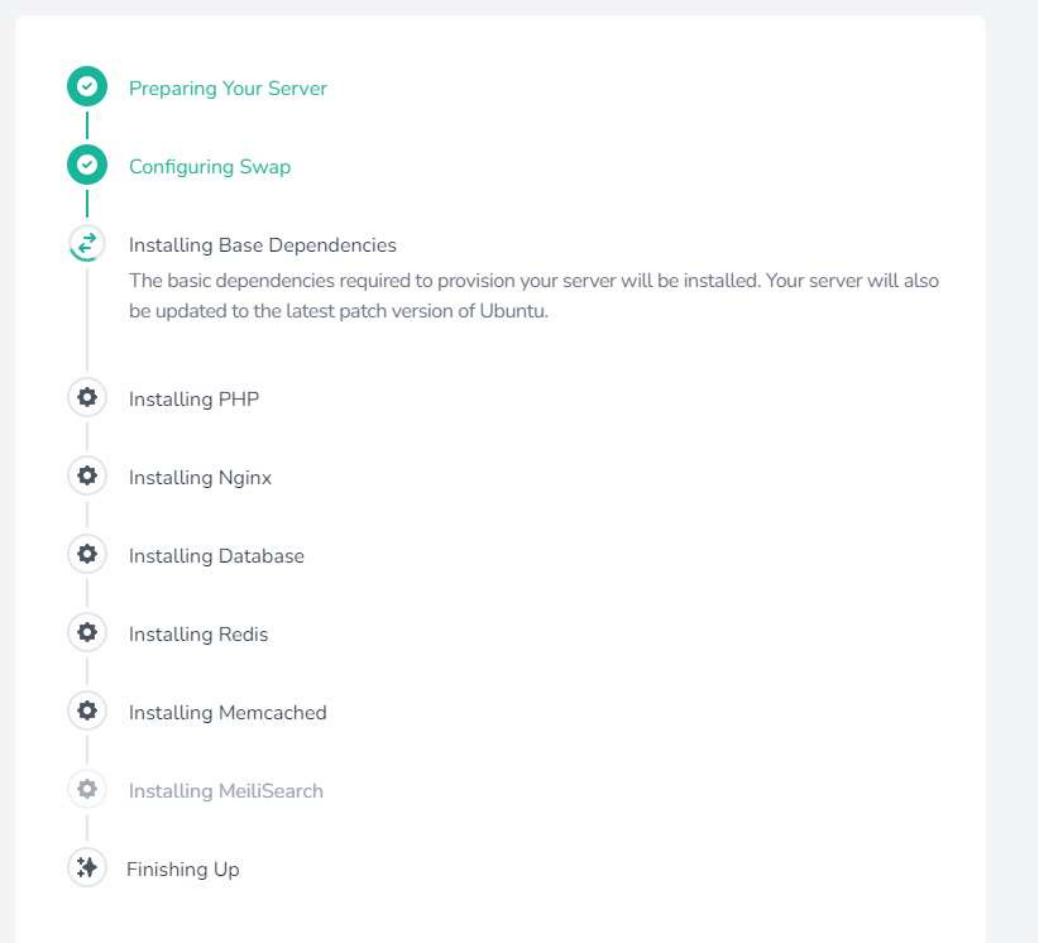


Figura 80. Proceso de creación de un servidor con Laravel Forge (Chávez, 2022)

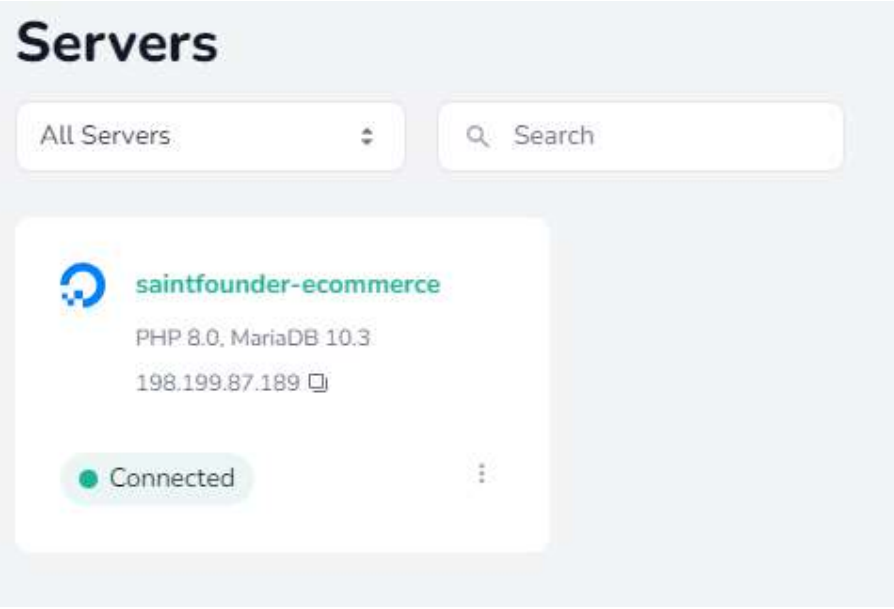


Figura 81. Servidor creando con Laravel Forge (Chávez, 2022)

Si se selecciona el servidor creado se pueden visualizar todas las opciones que ofrece "Laravel Forge", de la cuales para el caso de estudio solo se usaran las de "Sites" para crear una aplicación web dentro del servidor el cual se conectara a la base de datos que se crea por defecto.

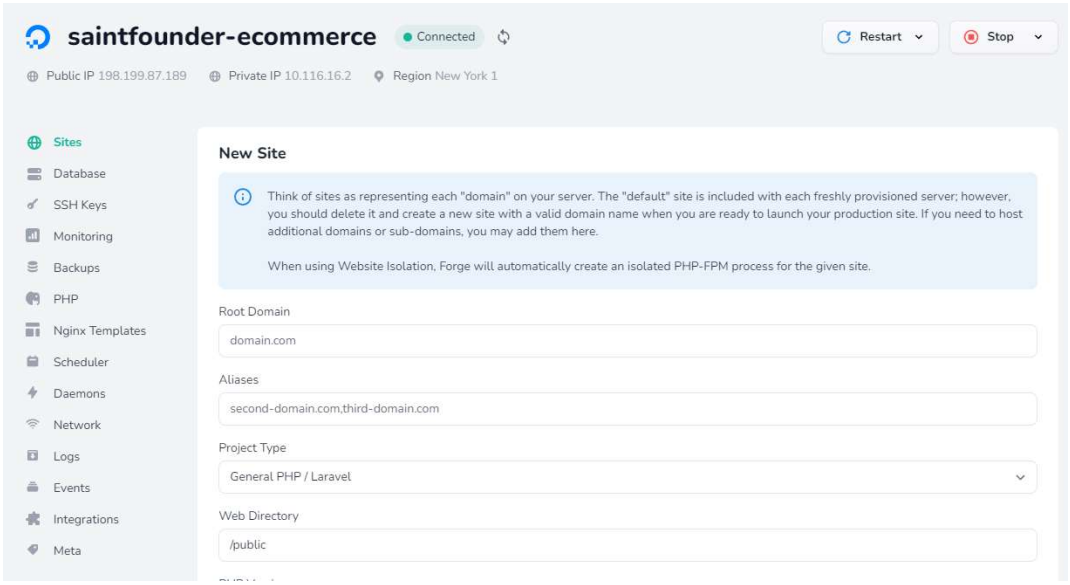


Figura 82. Opciones de servidor (Chávez, 2022)

Si seleccionamos en crear un sitio nuevo, para poder realizar le despliegue de este dentro del servidor es necesario que se enlace con el repositorio donde se encuentra la aplicación web, y si se selecciona el botón de "Install Repository" comienza la instalación.

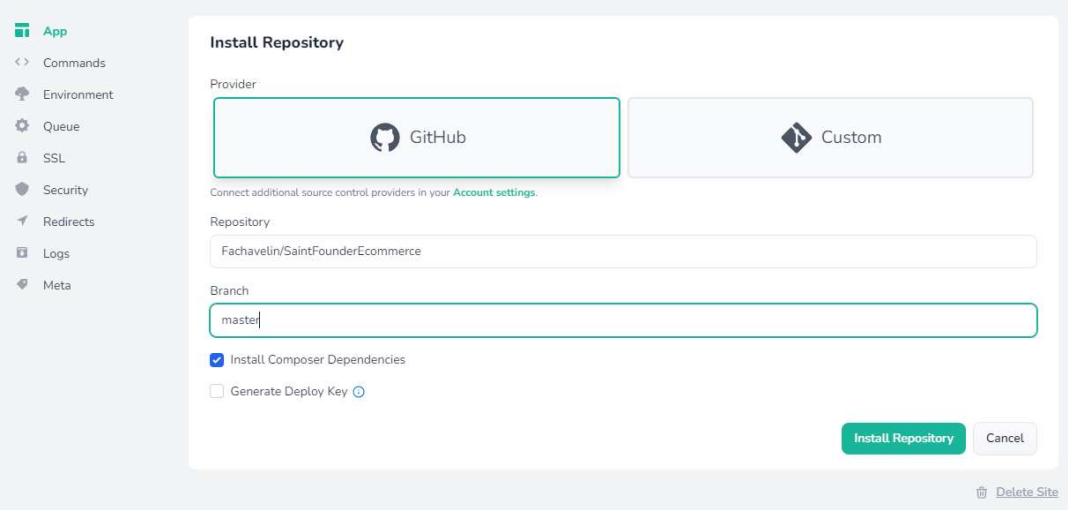


Figura 83. Instalación de un proyecto desde Github (Chávez, 2022)

Una vez se haya instalado el sitio dentro del servidor este será accesible, desde cualquier parte con conexión a internet, sin necesidad de realizar muchas configuraciones gracias a todas las facilidades brindadas para “Laravel Forge”.

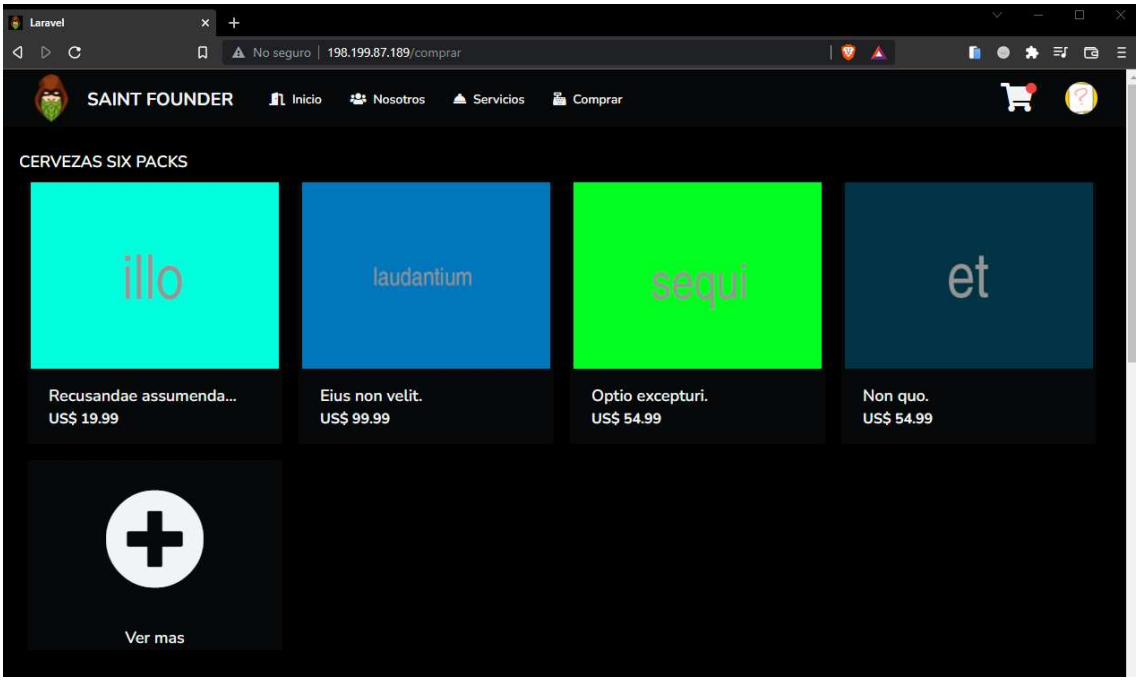


Figura 84. Sitio desplegado en un servidor (Chávez, 2022)

Capítulo 5: Conclusiones y Recomendaciones

Conclusiones.

- La implementación de aplicaciones web de tipo e-commerce. permitirá a las empresas ofrecer los productos y servicios a sus consumidores de forma rápida, fácil y accesible, manteniendo la información siempre actualizada.
- La seguridad de la aplicación se logra con esquemas de roles, el cual otorga permisos de escritura y lectura de los datos, a los administradores permitiendo la capacidad de controlar e ingresar la información sensible al sistema.
- La metodología Kanban permite ajustarse al ciclo de vida necesario para el desarrollo del proyecto, otra ventaja con respecto a otras metodologías agiles es que no necesita definir roles para su desenvolvimiento. Por ejemplo, la metodología Scrum necesita establecer los roles de “*Scrum master*”, “*Product owner*” y el equipo de desarrollo.
- La metodología Kanban permite el avance progresivo del proyecto mediante la descomposición de cada requerimiento funcional en tareas.

- La aplicación de herramientas de desarrollo, mejoran la productividad en el correcto desenvolvimiento del proyecto, destacando en especial el Framework Laravel
- Laravel Forge, facilita el proceso de subir a producción la aplicación web, reduciendo los numerosos procesos que son necesarios realizar de manera manual en un servidor para ejecutar aplicaciones Laravel.
- La construcción de un modelo relacional de Base de Datos es de gran importancia para el desarrollo de un sistema, su desarrollo establece el modelado de datos que será utilizado en la aplicación.

Recomendaciones.

- Se recomienda estudiar previamente las herramientas de desarrollo, que evitarán algún tipo de inconvenientes relacionados con el desarrollo de proyectos informáticos.
- Para dar seguridad a la información susceptible de los sistemas, se recomienda que exista al menos una persona encargada de administrar la aplicación bajo el rol de administrador, y sea la que administre la información actualizada de los productos y servicios que ofrece la empresa Saint Founder.
- Se recomienda realizar un análisis previo de los requerimientos de la empresa antes de comenzar con la planificación y desarrollo de la aplicación, para tener una idea base de como solventar las necesidades de la empresa.
- Se recomienda utilizar Kanban como metodología de desarrollo, su ayuda visual permite establecer las distintas tareas de las cuales se compone el ciclo de vida del proyecto, de una manera sencilla.

Bibliografía

Barrientos Felipa, P. (2017). Marketing + internet = e-commerce: oportunidades y desafíos. *Finanzas y Politica Económica*, 17.

Carrión Gonzáles, J. (2020). *El imacto del e-commerce en las pymes de las provincia de el oro.* Obtenido de https://rus.ucf.edu.cu/: https://rus.ucf.edu.cu/index.php/rus/article/view/1547

Dakduk, S., & Ricardo, D. (2020). *Transacciones electrónicas en Ecuador durante el Covid-19.*

Dimes, T. (2015). *Conceptos básicos de Scrum: Desarrollo de software Agile y manejo de proyectos Agile.* Babelcube, Inc.

Kanban University. (21 de Feb de 2011). *La guía oficial del método Kanban.* Obtenido de https://resources.kanban.university/wp-content/uploads/2021/08/The-Official-Kanban-Guide_Spanish_A4.pdf

Laínez Fuentes, J. R. (2015). *Desarrollo de Software ÁGIL: Extreme Programming y Scrum.* IT Campus Academy.

Molina Ríos, J. R., Zea Ordoñez, M. P., Contento Segarra, M. J., & García Zerda, F. G. (2018). *COMPARACIÓN DE METODOLOGÍAS EN APLICACIONES WEB.* Obtenido de https://www.3ciencias.com/articulos/articulo/comparacion-metodologias-aplicaciones-web/

Ortega, M. A., & Camacho, E. D. (2019). *Uso de los modelos tradicionales y las metodologías ágiles aplicadas en la industria de software colombiano.* Obtenido de https://repository.usc.edu.co/bitstream/handle/20.500.12421/1730/USO%20DE%20MODELOS.pdf?sequence=1&isAllowed=y

Piñeiro Gomez, J. M. (2014). *Diseño de bases de datos relacionales.* Ediciones Paraninfo.

Rojas, J. (2017). *Estrategias de Marketing Digital para empresas de E-commerce.* Obtenido de http://bibliotecadigital.econ.uba.ar/.

Solis Fonseca, D., Roque Pérez, W., & Morilla Faurés, M. L. (2013). Pasarela de pagos para la seguridad de transacciones bancarias en línea. *3C Empresa. Investigación Y Pensamiento crítico*.

stackoverflow. (2021). Obtenido de https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language-prof

UEES. (2017). *ANTECEDENTES Y SITUACIÓN DEL E-COMMERCE EN ECUADOR.*

Zambrano Velasco, B., Castellanos Espinoza, E., & Miranda Guatumillo, M. (2021). EL E-COMMERCE EN LAS EMPRESAS ECUATORIANAS. UN ANÁLISIS DE LOS INFORMES DE LA CÁMARA ECUATORIANA DE COMERCIO ELECTRÓNICO (CECE) EN EL MARCO DE LA PANDEMIA COVID-19. *Publicando*, 14-20.

Zamora, C. (2017). La importancia del emprendimiento enla economía: el caso de Ecuador. *Revista Espacios*, 12.

Anexo 1:

Diccionario de datos:

Tabla: Roles		
Nombre del campo	Tipo de dato	Descripción del campo
id	int	Clave única auto incrementable de la tabla roles
Nombre del campo	varchar(255)	Nombre del rol

Tabla 11. Tabla roles del diccionario de datos (Chávez, 2022).

Tabla: Rol_Usuario		
Nombre del campo	Tipo de dato	Descripción del campo
id	int	Clave única auto incrementable de la tabla rol_usuario
Usuario_id	int	Clave unica de la tabla roles
Rol_id	int	Clave unica de la tabla usuarios

Tabla 12. Tabla rol_usuairo del diccionario de datos (Chávez, 2022).

Tabla: Usuarios		
Nombre del campo	Tipo de dato	Descripción del campo
id	int	Clave única auto incrementable de la tabla usuarios
nombre	varchar(255)	Nombre del usuario
email	varchar(255)	Correo electrónico del usuario
password	varchar(255)	Contraseña encriptada del usuario

Tabla 13. Tabla Usuarios del diccionario de datos (Chávez, 2022).

Tabla: Ordenes		
Nombre del campo	Tipo de dato	Descripción del campo
id	int	Clave única autoincrementable de la tabla rol_usuario
Usuarios_id	int	Clave unica de la tabla usuarios
contacto	varchar(255)	Nombre de la persona a la cual va dirigida la orden
telefono	varchar(255)	Telefono de contacto de la persona a la cual va dirgida la orden
estado	enum(1,2,3,4,5)	Estado de la orden que admite los valores de: "1", "2","3","4" y "5" siendo estos respectivamente "pendiente", "pagado", "enviado", "entregado" y "anulado"

tipo_de_envio	enum(1,2)	Tipo de envío que admite los valores de: "1" y "2" siendo estos de "retiro en tienda" y "envío a domicilio"
costo_de_envio	double	Costo por el envío.
total	double	Sumatoria de todos los costos.
contenido	json	Contenido de los productos
direccion_de_envio	varchar(255)	Dirección a la cual se va a realizar el envío
referencia	varchar(255)	Referencia necesaria para realizar un envio
Provincias_id	int	Clave unica de la tabla provincias
Ciudades_id	int	Clave unica de la tabla ciudades

Tabla 14. Tabla Ordenes del diccionario de datos (Chávez, 2022).

Tabla: Provincias		
Nombre del campo	Tipo de dato	Descripción del campo
id	int	Clave única auto incrementable de la tabla provincias
nombre	varchar(255)	Nombre de la provincia

Tabla 15. Tabla provincias del diccionario de datos (Chávez, 2022).

Tabla: Ciudades		
Nombre del campo	Tipo de dato	Descripción del campo
id	int	Clave única auto incrementable de la tabla ciudades
nombre	varchar(255)	Nombre de la ciudad
precio	double	Precío que se agrega para la ciudad
Provincias_id	int	Clave única de la tabla provincias

Tabla 16. Tabla ciudades del diccionario de datos (Chávez, 2022).

Tabla: Categorías		
Nombre del campo	Tipo de dato	Descripción del campo
id	int	Clave única auto incrementable de la tabla categorias
nombre	varchar(255)	Nombre de la categoría
slug	varchar(255)	Url amigable de la categoría
visible	tinyint(1)	Estado de visibilidad de la categoría

Tabla 17. Tabla de categorías del diccionario de datos (Chávez, 2022).

Tabla: Productos		
Nombre del campo	Tipo de dato	Descripción del campo
id	int	Clave única auto incrementable de la tabla productos
nombre	varchar(255)	Nombre del producto
slug	varchar(255)	Url amigable del producto
descripcion	text(500)	Descripción del producto
precio	dobule	Precio del producto
Categorias_id	int	Clave única de la tabla categorías
cantidad	int	Cantidad disponible del producto
visible	tinyint(1)	Estado de visibilidad del producto

Tabla 18. Tabla de productos del diccionario de datos (Chávez, 2022).

Tabla: Servicios		
Nombre del campo	Tipo de dato	Descripción del campo
id	int	Clave única auto incrementable de la tabla servicios
titulo	varchar(255)	Título del servicio
descripcion	text(500)	Descripción del servicio
visible	tinyint(255)	Estado de visibilidad del producto

Tabla 19. Tabla de servicios del diccionario de datos (Chávez, 2022).

Tabla: Imágenes		
Nombre del campo	Tipo de dato	Descripción del campo
id	int	Clave única auto incrementable de la tabla imágenes
url	varchar(255)	Url de la imagen
imageable_id	int	Clave única de la tabla polimórfica
imageable_type	varchar(255)	Tipo de imagen

Tabla 20. Tabla de imágenes del diccionario de datos (Chávez, 2022).