



UNIVERSIDAD CÉSAR VALLEJO

FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

“Marketplace para el proceso de ventas en la empresa AC  
Brands & Consulting E.I.R.L.”

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:  
Ingeniero de Sistemas**

**AUTOR:**

Perez Arbulu, Marzio Josué ([ORCID: 0000-0001-6115-0246](https://orcid.org/0000-0001-6115-0246))

**ASESOR:**

Dr. Pérez Farfán, Iván Martín ([ORCID: 0000-0001-5833-9400](https://orcid.org/0000-0001-5833-9400))

**LÍNEA DE INVESTIGACIÓN:**

Sistema de información y comunicaciones

LIMA – PERU

2021

## **DEDICATORIA**

Dedicado a mi pequeña hija, que es mi fuerza motora e inspiración para alcanzar mis sueños y que se hagan realidad.

## **AGRADECIMIENTO**

Agradecer a mis asesores y profesores universitarios, los cuales han sido una guía constante a lo largo de esta profesión.

## ÍNDICE DE CONTENIDOS

DEDICATORIA.....	ii
AGRADECIMIENTO .....	iii
ÍNDICE DE CONTENIDOS .....	iv
ÍNDICE DE TABLAS.....	v
ÍNDICE DE FIGURAS .....	vi
RESUMEN.....	vii
ABSTRACT .....	viii
I. INTRODUCCIÓN .....	9
II. MARCO TEÓRICO .....	13
III. METODOLOGÍA .....	26
3.1. Tipo y diseño de investigación.....	26
3.2. Variables y operacionalización.....	27
3.3. Población, muestra y muestreo.....	31
3.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad 33	
3.5. Procedimientos.....	37
3.6. Método de análisis de datos .....	38
3.7. Aspectos éticos .....	42
IV. RESULTADOS.....	43
V. DISCUSIÓN .....	53
VI. CONCLUSIONES.....	54
VII. RECOMENDACIONES.....	55
REFERENCIAS.....	56
ANEXOS.....	61

## ÍNDICE DE TABLAS

<b>Tabla 1: Criterios de evaluación de las metodologías de desarrollo .....</b>	<b>25</b>
<b>Tabla 2: Validación de expertos para la aplicación de la metodología de desarrollo.....</b>	<b>25</b>
<b>Tabla 3: Determinación de la Población.....</b>	<b>31</b>
<b>Tabla 4: Técnicas e Instrumentos de recolección de datos .....</b>	<b>34</b>
<b>Tabla 5: Validación de expertos del Instrumento de Investigación .....</b>	<b>34</b>
<b>Tabla 6: Niveles de Confiabilidad.....</b>	<b>36</b>
<b>Tabla 7: Correlación Calidad de pedidos generados.....</b>	<b>36</b>
<b>Tabla 8: Correlación Ingresos .....</b>	<b>37</b>
<b>Tabla 9: Procedimientos de recolección de datos .....</b>	<b>38</b>
<b>Tabla 10: Análisis descriptivo del indicador Calidad de pedidos .....</b>	<b>43</b>
<b>Tabla 11: Análisis descriptivo del indicador Calidad de pedidos .....</b>	<b>44</b>
<b>Tabla 12: Prueba de normalidad para el indicador calidad de pedidos.....</b>	<b>46</b>
<b>Tabla 13: Prueba de normalidad para el indicador ingresos .....</b>	<b>47</b>
<b>Tabla 14: Aplicación de la prueba t-student para el indicador calidad de pedidos .....</b>	<b>50</b>
<b>Tabla 15: Aplicación de la prueba t-student para el indicador Ingresos.....</b>	<b>51</b>

## ÍNDICE DE FIGURAS

<b>Figura 1: Ingresos marzo 2021</b> .....	10
<b>Figura 2: Calidad de pedidos generados marzo 2021</b> .....	11
<b>Figura 3: Fórmula calidad de pedidos generados</b> .....	19
<b>Figura 4: Fórmula de Ingresos</b> .....	20
<b>Figura 5:Arquitectura de software</b> .....	21
<b>Figura 6: Marco de trabajo SCRUM</b> .....	23
<b>Figura 7: Diseño-de-estudio</b> .....	27
<b>Figura 8: Fórmula de la muestra</b> .....	32
<b>Figura 9: Coeficiente de correlación de Pearson</b> .....	35
<b>Figura 10: Distribución de T de Student</b> .....	40
<b>Figura 11: Distribución de T de Student</b> .....	41
<b>Figura 12: Rango de valores de la distribución T de Student</b> .....	41
<b>Figura 13: Histograma del análisis descriptivo del indicador Calidad de pedidos</b> .....	44
<b>Figura 14: Histograma del análisis descriptivo del indicador Ingresos</b> .....	45
<b>Figura 15: Histograma con curva normal para el indicador calidad de pedidos antes de la implementación del marketplace</b> .....	46
<b>Figura 16: Histograma con curva normal para el indicador calidad de pedidos después de la implementación del marketplace</b> .....	47
<b>Figura 17: Histograma con curva normal para el indicador ingresos antes de la implementación del marketplace</b> .....	48
<b>Figura 18: Histograma con curva normal para el indicador ingresos después de la implementación del marketplace</b> .....	48
<b>Figura 19: Curva de región de rechazo para el indicador calidad de pedidos</b> .....	50
<b>Figura 20: Curva de región de rechazo para el indicador Ingresos</b> .....	52

## RESUMEN

Esta investigación se realizó en la empresa AC BRANDS & CONSULTING E.I.R.L., con la intención de mejorar el proceso de ventas, para esto se planteó como problema principal ¿De qué manera el Marketplace influye en el proceso de ventas de AC Brands & Consulting E.I.R.L.? y en base a este problema se planteó el objetivo principal como determinar de qué manera el Marketplace influye en el proceso de ventas de AC Brands & Consulting E.I.R.L. Para poder lograr realizar la medición de la influencia del marketplace sobre el proceso de ventas se plantearon dos indicadores, el primero fue la calidad de los pedidos y el segundo los ingresos. Lo que se espera es que estos indicadores aumenten. El tipo de estudio es aplicado, ya que se implementará el marketplace para poder evaluar la influencia sobre el proceso de ventas midiendo los indicadores, en diferentes tiempos el primero antes de la implementación y el segundo posterior a la implementación, por esto es que el diseño fue de tipo pre experimental. Para la evaluación de ambos indicadores se definió como población 1500 pedidos y como muestra 306 pedidos, los cuales fueron recolectados por medio de las fichas de recolección de datos.

El desarrollo del marketplace fue con la base de datos MYSQL y el framework laravel, además se utilizó el marco de trabajo de SCRUM, para poder generar entregables funcionales al cliente y pueda ir familiarizándose con el sistema.

Se concluye que para el indicador calidad de pedidos existió un aumento de 60.6% a un 74.3% lo cual equivale a un 13%, y también para el indicador ingresos hubo un incremento desde un promedio de 90.29 soles diarios a un 178.86.

Palabras claves: marketplace, proceso de ventas

## **ABSTRACT**

This research was carried out in the company AC BRANDS & CONSULTING E.I.R.L., with the intention of improving the sales process, for this the main problem was raised: How does the Marketplace influence the sales process of AC Brands & Consulting E.I.R.L.? and based on this problem, the main objective was raised as to determine how the Marketplace influences the sales process of AC Brands & Consulting E.I.R.L. In order to be able to measure the influence of the marketplace on the sales process, two indicators were proposed, the first was the quality of the orders and the second the income. What is expected is that these indicators will increase. The type of research is applied since the marketplace will be implemented in order to evaluate the influence on the sales process by measuring the indicators, in two stages, the first before its implementation and the second after the implementation, for this reason the design was pre-experimental type. For the evaluation of both indicators, 1500 requests were defined as population and 306 requests as a sample, which were collected through the data collection sheets.

The development of the marketplace was with the laravel framework and the MYSQL database, in addition the SCRUM framework was used, to be able to generate functional deliverables to the client and to become familiar with the system.

It is concluded that for the order quality indicator there was an increase from 60.6% to 74.3% which is equivalent to 13%, and also for the income indicator there was an increase from an average of 90.29 soles per day to 178.86.

Keywords: marketplace, sales process



## **I. INTRODUCCIÓN**

Actualmente el comercio electrónico es un tema que la gran mayoría de las personas conocen, habiendo realizado alguna transacción o compra por Internet, esto gracias al crecimiento y globalización continuo sobre la tecnología de información, más aún en este tiempo en donde la pandemia ha generado grandes cambios en el modo de vivir de las personas, y sobretodo en seguir manteniendo la economía apoyada en las tecnologías, y el comercio virtual ha tomado un papel muy importante en la realidad actual.

En el marco internacional el Diario La Vanguardia (2020) el comercio electrónico ha madurado bastante en el tiempo de confinamiento esto debido a la covid-19, los clientes aumentaron sus compras por medio de internet, generando que toda organización se vea obligada a repotenciar sus ventas mediante páginas web. Se asegura que alrededor de un 23% de todos los habitantes han incrementado sus compras por internet durante este periodo. Pero también existieron acontecimientos, según La BBC menciona (2020) que Amazon, eBay entre otras grandes tiendas tuvieron que enfrentar un gran problema de especulación de precios a consecuencia de la pandemia. Los precios de los productos sobre todo los orientados a la salud, tuvieron un aumento exagerado en los precios, aprovechando el temor de las personas, por lo cual Amazon tuvo que tomar medidas bastante drásticas como el de suspender cerca de 1 millón de productos que generaban abuso.

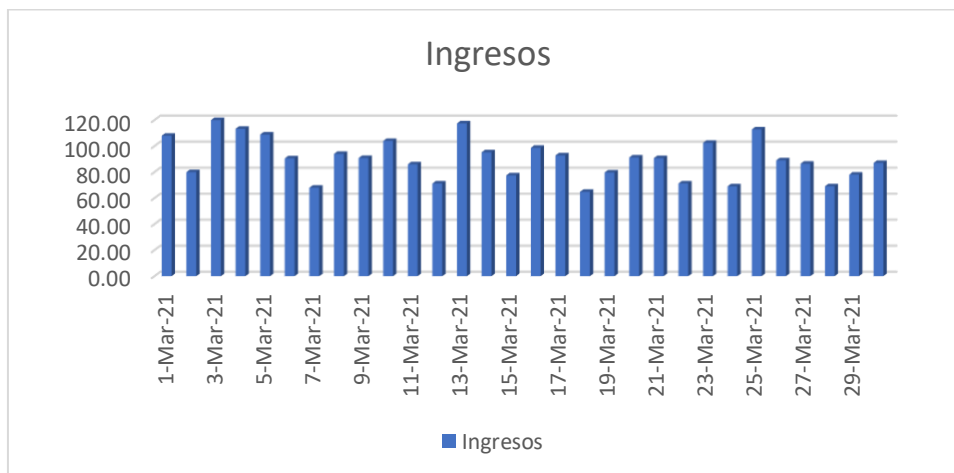
En el plano nacional para el Diario El Comercio (2020) menciona que, a partir de la cuarentena, las empresas peruanas se vieron inmersas a la adopción de soluciones tecnológicas e implementar y E-commerce para poder continuar con su negocio, pero no basta solamente con implementar una plataforma, son más procesos los que vienen de la mano. También, menciona que el 44% de compradores en línea en el Perú realizaron una primera compra digital en la época de confinamiento y que alrededor de 83% siguió con la compra.

La presente investigación se realizará en la empresa AC Brands & Consulting E.I.R.L. registrada con el RUC 20606801417 y ubicada en Jr. Junín #134 Urb. Orbea Lima, Lima - Magdalena del Mar, la cual cuenta con diferentes socios estratégicos comerciales que ofrecen diversos productos y servicios para

mascotas. Se realizó la entrevista al Sr. Juan Carlos Galagarza (Ver anexo 2), Gerente Comercial de la empresa AC Brands & Consulting E.I.R.L. la cual se veía afectada por la coyuntura de la pandemia actual ya que gestionaban sus ventas a través de llamadas, mensajes de WhatsApp y los registraba en libros Excel, donde se almacenaban la información de los pedidos, además de la reserva de paseos o entrenamientos que los clientes solicitan para sus mascotas. Es por ello que la problemática se centraba en el proceso de ventas ya que hoy en día existen diferentes herramientas que permiten la mejor gestión y manejo de la información de estas. De lo indicado anteriormente se valida que la empresa tenía problemas al llevar un proceso de ventas de manera correcta, problemas como cuando se solicitaba un producto, no se sabía en tiempo real el stock actualizado lo cual generaba pérdida de tiempo en consulta de stocks, y muchas veces no se contaba con el stock ya que no existía un informe de consumo del mismo, esto se daba por la falta de un seguimiento de stock por producto, además de la demora en la generación de reportes, ineficiencia en la información de los inventarios y además de gestionar las diferentes comisiones para así percibir sus ganancias al final de cada jornada. Esto finalmente generaba tiempos muertos, pérdida de posibles ventas, pérdida de recursos, y sobre todo incomodidad e insatisfacción de los clientes.

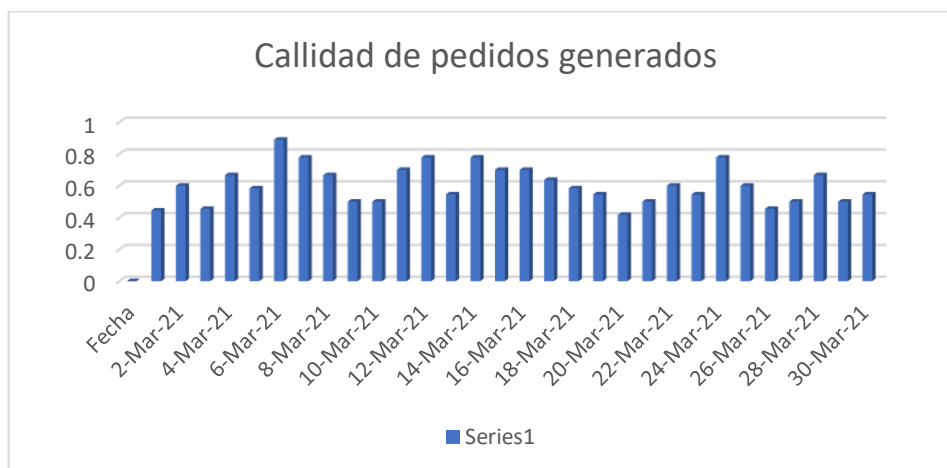
Se realizó un análisis de los indicadores en el mes de marzo, en donde para los ingresos se obtuvo un total de S/2708.71 de ingresos por las comisiones, lo cual está bastante alejado a lo estimado que debería ser por lo menos 3 veces más el monto de dicho mes, el detalle se visualiza en la siguiente figura:

**Figura 1: Ingresos marzo 2021**



Para el indicador calidad de pedidos, se obtuvo un resultado del 61% de un 100% ideal, lo que indica es que los pedidos no se están entregando en el tiempo adecuado, el detalle se visualiza en el siguiente gráfico:

**Figura 2: Calidad de pedidos generados marzo 2021**



En base al análisis de la situación actual de la empresa AC Brands & Consulting E.I.R.L. se presenta la siguiente problemática general: ¿De qué manera el Marketplace influye en el proceso de ventas de AC Brands & Consulting E.I.R.L.? Y las siguientes problemáticas específicas: 1) ¿Cómo el Marketplace influenciará en la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L.? 2) ¿Cómo el Marketplace influenciará en los ingresos en el proceso de ventas de AC Brands & Consulting E.I.R.L.?

Según lo investigado se plantea el siguiente objetivo general: Determinar de qué manera el Marketplace influye en el proceso de ventas de AC Brands & Consulting E.I.R.L. Y los siguientes objetivos específicos: 1) Determinar cómo el Marketplace influye en la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L. 2) Determinar cómo el Marketplace influye en los ingresos en el proceso de ventas de AC Brands & Consulting E.I.R.L.

De los objetivos planteados anteriormente, se permiten plasmar la siguiente hipótesis general: El Marketplace mejora el proceso de ventas de AC Brands & Consulting E.I.R.L. Y las siguientes hipótesis específicas: 1) El Marketplace aumenta la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L. 2) El Marketplace incrementa los ingresos en el proceso de ventas de AC Brands & Consulting E.I.R.L.

En el plano social, la presente investigación pretende brindar a la sociedad un Marketplace que va a aumentar el valor bruto de mercadería y los ingresos en el proceso de ventas en AC Brands & Consulting E.I.R.L., la investigación beneficiará la cartera de clientes y comercios afiliados a la empresa la cual podrá automatizar y digitalizar los datos para medir y cumplir los objetivos que se plantean.

Desde el criterio práctico, el Marketplace ayudará a automatizar el proceso de ventas con el fin de identificar los ingresos, así como aumentar la calidad de pedidos generados, los cuales ahorrarán tiempo que se ejecuta de manera manual y con márgenes de error evidenciados en los últimos tiempos. Así como menciona Castellanos (2016) Una forma de ayudar a las Pymes a integrar su sistema en el uso masivo de la tecnología es diseñar programas que mejoren su resolución y aplicación en sus proyectos; los países en desarrollo han demostrado que la posibilidad de utilizar las TIC's hace que las empresas sean más competitivas y rentables que los países sin apoyo o sin una regulación clara.

Así también, desde el criterio teórico, se logrará incrementar el conocimiento sobre la relevancia y buenas prácticas de implementar un Marketplace en el mercado local como solución a los diferentes procesos de ventas que puedan necesitar los diferentes pequeños y medianos comercios aportando a futuras investigaciones. Pero se debe tener en cuenta lo siguiente para García y Sánchez (2013) mencionan que: "Las TICs pueden no aporta una ventaja competitiva. Puede invertir en tecnologías de la información y la comunicación de alta gama y no utilizarlas para reflejarse o hacer buenas acciones. Recuerde que para lograr mejores resultados que la competencia, las empresas deben utilizar sus recursos a través de estrategias, incluidas las TIC's, y para ello es necesario definir sus objetivos que sean claros".

Finalmente, como como criterio metodológico la implementación de un Marketplace en el proceso de ventas apoyará a la creación de reportes en determinados periodos de tiempo lo que permitirá a la empresa hace un análisis para la mejor toma de decisiones.

## II. MARCO TEÓRICO

(Huayllasco, 2019, p.53), en su investigación sobre la *Aplicación de un modelo de E-MARKETPLACE vertical en el proceso de venta de artículos de vestir en la galería CENTRO GAMARRA*, desarrollada en la Universidad Inca Garcilazo de la Vega, Lima, Perú - 2019. Tuvo como objetivo la determinación de dominio sobre las variables de estudio. El diseño de investigación fue experimental del tipo pre-experimental, aplicado a una población de 150 stands de la Galería “Centro Gamarra” y se tomó una muestra de 20 ventas. En cuanto al resultado obtenido, se pudo determinar que el tiempo para realizar una venta disminuyó de 4.03 minutos (antes) a 1.60 minutos (después) ya que el vendedor sólo confirma o rechaza el pedido previamente registrado desde la plataforma web. Además, se determinó que el número de ventas diarias incrementó de 20 (antes) a 53 (después). Por otro lado, se determinó que el tiempo empleado para la generación de los reportes de ventas disminuyó de 15 minutos (antes) a 1.4 minutos (después). Finalmente, luego de realizar pruebas pertinentes y utilizando el método SERVQUAL se obtuvo el 95% de aceptación. El autor concluye que, según los resultados obtenidos, posterior a la aplicación de E-marketplace en el procedimiento de ventas se llegaron a resultados significativos tanto en el aumento de las ventas, satisfacción del cliente como el tiempo para la generación de los reportes. De este estudio, se tuvo en cuenta los indicadores de medición cuantitativos al ser valores cruciales para la presente investigación.

(Palomino, 2018, p.47), en su investigación titulada *“Estudio sobre un sistema de información en la nube, aplicado a la gestión de ventas de la empresa La Calera de la Ciudad de Chincha”*, desarrollada en la Universidad Nacional San Luis Gonzaga de Ica, Ica, Perú. Tuvo como objetivo determinar cómo el estudio de sistemas de datos en la nube impacta en el procedimiento de venta de la organización La Calera de Chincha, 2018. El diseño del estudio es experimental, tipo cuasi experimental, teniendo como población el universo de todos los procesos de ventas que se llevaron en la primera semana de agosto 2017, teniendo un número de 140, obteniendo una muestra de 58 procesos luego de aplicar la fórmula del muestreo simple. Sobre el resultado obtenido se pudo determinar que para la emisión de una

venta el tiempo disminuyó de 14.6 minutos (antes) a 2.7 minutos (después). Por otro lado, se determinó que el tiempo para almacenar la información disminuyó de 7.8 segundos (antes) a 1.6 segundos (después). Finalmente, se identificó que el tiempo para realizar una venta disminuyó de 21.7 minutos (antes) a 4.9 minutos (después). El autor concluyó que había un nivel de aceptación por lo que recomienda la implementación de herramientas en la nube para el procedimiento de venta sobre la referida empresa. Así mismo, en la investigación se tomó en cuenta los conceptos y medición del indicador para determinar el tiempo en realizar una venta.

(Flores y Valdez, 2019, p.12), en su investigación titulada *“Marketplace para el proceso de ventas de la Empresa Creaciones Peloncito S.A.C.”*, desarrollada en la Universidad César Vallejo, Lima, Perú. Se tuvo como objetivo principal determinar la influencia del Marketplace en el procedimiento de venta de la organización Creaciones Peloncito S.A.C., 2019. El diseño de estudio fue tipo pre experimental, aplicado, La población tomada en la investigación fue de 118 días de registros de ventas para el indicador de productividad ventas, así como una población de 12 semanas de registro de productos devueltos para ver la incidencia en la venta, obteniendo una muestra de 12 registros que se realizaron en 3 meses trabajados. En cuanto a los resultados, sobre la productividad de venta se pudo determinar un incremento de 42.83% (antes) a 71.65% (después) al realizar la implementación del Marketplace. Los autores concluyen que la implementación de un Marketplace en la entidad Peloncito S.A.C. mejoró el proceso de ventas dando resultados óptimos en un corto tiempo, dado el aumento de productividad, así como la disminución del índice de calidad de ventas al tener menos devoluciones de productos vendidos. De esta investigación, se tomó en cuenta el concepto y dimensión de las variables, ya que fue fundamental para el presente estudio.

(Bernal y Pocco, 2018, p.10), En su investigación titulada *“Implementación de un sistema E-commerce, para mejorar el proceso comercial de la empresa SERGETIC S.A.C.”*, desarrollada en la Universidad Autónoma del Perú, Lima, Perú. El estudio tuvo como objetivo determinar la medición de la implementación de un sistema E-commerce mejorando el procedimiento comercial de SEGETIC S.A.C., 2018. La

investigación tuvo un diseño experimental de tipo pre-experimental, empleado a una población de los procesos del área comercial teniendo una muestra de 30 procesos. En cuanto a los resultados, se determinó que la cantidad de pedidos ascendió de 23 (antes) a 33 (después), así como el tiempo en generar reportes del área comercial donde disminuyó de 105.27 minutos (antes) a 17.50 minutos (después), finalmente, se determinó que el tiempo empleado para consultar el stock disminuyó de 40 minutos (antes) a 4 minutos (después), dichos resultados fueron contrastados en cada hipótesis con el criterio de significancia p-valor menor a 0.05, encontrándose diferencias significativas. Los autores concluyen que según los resultados obtenidos ejecutar E-commerce mejora los procedimientos comerciales en cuanto al aumento en índice de pedidos y mejora en el tiempo de reporte de venta en la empresa SEGETIC S.A.C. De la investigación, se tomó en consideración la variable dependiente del proceso comercial al ser un término relacionado con el presente estudio.

(Inca, 2017, p.29), en su estudio titulado "*Sistema web para el proceso de ventas en la empresa Zoe*", desarrollada en la Universidad César Vallejo, Lima, Perú. Se tuvo como objetivo determina la influencia del empleo de un sistema web en la fidelización de cada cliente, así como en la eficacia de las ventas en la entidad Zoe, 2018. El diseño del estudio fue aplicado y de tipo pre-experimental, se empleó a una población de 8 registros de ventas que se generaron en 2 meses, obteniendo una muestra del mismo número al ser una población finita. En cuanto a los resultados, se aumentó la fidelidad de los clientes de 0.28 (antes) a 0.58 (después), así también se obtuvo un mayor número de eficacia en las ventas de 0.47 (antes) a 0.63 (después) luego de haber empleado el software del procedimiento de venta, dichos resultados fueron contrastados en cada hipótesis con el criterio de significancia p-valor menor a 0.05, encontrándose diferencias significativas. El autor concluye que los resultados que se obtuvieron brindan información de mucha importancia en los procesos de la empresa Zoe, dado que la automatización de esta mejora el nivel de eficacia, así como la fidelización de los clientes. De este estudio podemos tomar en cuenta los conceptos respecto a la variable del proceso de ventas, puesto que es un parte parte crucial en la presente tesis.

(Kawa y Wałęsiak, 2019, p.522), en su artículo *“Marketplace como actor clave en redes de valor de comercio electrónico”*, publicado en la revista LogForum, 2019. Dicho artículo proporciona información detallada respecto a los Marketplace donde se describe la división y características de los mismos y se discuten las tendencias potenciales de este rubro. Los autores mencionan en el año 2018 el comercio electrónico incrementó su valor a USD 2,84 T y en el 2019 se espera superar los USD 3,45 T y suba a USD 4,88 billones en el 2021, dando a conocer que los productos más comprados tienen que ver con moda, electrónica, etc. Además, indican que un Marketplace es una plataforma donde se pueden ofrecer diferentes productos y servicios, y es importante que otras empresas puedan vender sus productos, de lo contrario solo sería una tienda. Los autores concluyen indicando que gracias a los Marketplaces se han creado nuevas oportunidades de ventas para la expansión de los vendedores online en Polonia, así también indican que estas empresas no necesitan invertir en una plataforma de ventas o tener conocimientos legales y técnicos. De este artículo podemos tomar en cuenta la importancia de implementar un Marketplace en el mercado local, dado que genera valor para los diferentes pequeños y medianos empresarios.

(Li, Ch'ng, Yee-Loong, y Bao, 2016, p.15), en su artículo *“Predecir el rendimiento de las ventas en el Marketplace en línea: Un enfoque de big data”*, publicado en la revista Computación e Ingeniería Industrial, 2016. Dicho artículo menciona la importancia de realizar de manera eficiente la categorización de los productos que se ofrecen en los Marketplaces, esto a través de la revisión del volumen y calificación para mejorar el desempeño de las ventas. Los autores indican que el rendimiento de las ventas se puede ver potenciado debido a las reseñas, evaluación que los clientes realizan por sobre los productos, es decir, a mayor número de reseñas más información disponible para una posible compra. También mencionan que es importante que los vendedores de los Marketplace respondan a las consultas de sus productos, ya que esto aporta valor para una futura venta y así evitar futuros problemas. Concluyen indicando que parte de algunas estrategias para obtener reseñas o preguntas, es promover los envíos gratuitos, la importancia de este tipo de estrategias es que finalmente el cliente pueda realizar comentarios o dejar reseñas sobre el producto para así conseguir lo que se llaman ventas



cruzadas o por calificación. De este artículo tendremos en cuenta la información y estrategias a adoptar para obtener un mayor volumen de ventas, dado es que es parte de uno de los indicadores de la presente tesis.

(Corrêa da Silva et. al., 2019, p.24), en el artículo *“E-MARKETPLACES: Canales que mejoran el rendimiento de ventas en línea para Ecommerces”*, publicado en la revista Cuaderno de Administración, 2019. Dicho artículo tiene como objetivo realizar el análisis del crecimiento de los comercios electrónicos “E-commerce” en un Marketplace a lo largo del tiempo. En dicho artículo se analizó la información de ventas realizadas en 28 meses (2015 – 2017), la muestra fue de 444 comercios que empezaron a utilizar la plataforma y los datos se obtuvieron en base a 7 Marketplaces. Los autores analizaron los indicadores de valor líquido de ventas, valor líquido vendido por mes, cantidad de pedidos al mes, valor líquido de ventas por Marketplace y cantidad de pedidos registrados por Marketplace. Obteniendo resultados significativos del gran número de pedidos y ventas que los comercios pueden obtener al afiliarse en un Marketplace. Los autores concluyen que los Marketplaces han demostrado ser una fuente de venta competitiva para los comercios electrónicos, ya que son plataformas que aportan ayuda a los vendedores y compradores para que estos puedan tener resultados eficientes a nivel de compra y venta de diferente tipos de artículos. De este artículo tendremos en cuenta la importancia de implementar un Marketplace en el mercado local para un rubro específico, así como el aporte que nos da para la medición en base a los indicadores que utilizaron los investigadores para la presente tesis.

Para un correcto respaldo de la investigación se han obtenido diferentes referencias teóricas sobre el tema de investigación, una de ellas es Marketplace, es cual es mencionado por Wertz y Tran (2015) que refieren que un Marketplace es aquel medio comercial en internet que permite relacionar a aquellos que buscan ofrecer productos o servicios “vendedores” con aquellos usuarios que buscan adquirir esos productos o servicios “clientes”. Ambos elementos pueden haber tenido problemas para encontrarse anteriormente, es por ello que un Marketplace crea una eficiencia en un entorno que de algún modo es ineficiente. Con la tendencia de los

Marketplaces hoy en día existen lugares “plataformas” donde podemos conectar con vendedores de inmuebles, bandas de música, etc. (p.6).

Fruhirth, Rachinger, y Prlja (2020) manifiestan que los Marketplace son plataformas digitales que permiten la interacción entre grupos interdependientes de usuarios “vendedores y clientes” los cuales realizan el intercambio de bienes mediante diferentes estrategias comerciales en donde influye como principal factor el precio del producto o servicio. También indican que este tipo de plataformas son las encargadas de facilitar este tipo de interacciones y como tercer actor, su función es crear valor entre sí (p.5739).

Según Mohammad y Allah, (2020) el Marketplace es una plataforma que permite a los fabricantes vender sus productos en línea, diferentes empresas han utilizado este tipo de plataformas tales como eBay, Amazon, Taobao, etc. El Marketplace permite a los vendedores ofrecer sus productos sin tener que realizar una configuración adicional, además de brindar oportunidades comerciales las cuales se pueden resumir en: El vendedor puede determinar el precio por sí mismo y tiene acceso total a los clientes que están en línea. El vendedor puede acceder al Marketplace y aumentar su conocimiento sobre el potencial del rubro.

Por otro lado, se recolectó información sobre Proceso de ventas, donde según Kotler y Armstrong, (2016) nos indican que el procedimiento de venta, resulta una secuencia de actividades que el comercio emprende, generando una negociación de potencial, teniendo como objetivo generar una acción que se desea sobre el mismo, es decir, sobre la compra. Así también, según Acosta, Salas, Guerra y Jiménez, (2018) una venta está representada por un proceso y diferentes autores coinciden en la importancia de estas fases las cuales son: Preparación, Concertación, Contacto, Sondeo, Argumentación, Objeciones y Cierre. Además, indican que los vendedores deben contemplar a los clientes como parte de sus actividades y funciones diarias ya que esto es parte de constituir un proceso de ventas exitoso. Estas fases se explican de la siguiente manera:

**Preparación:** Se realiza una organización para atender a los clientes existentes y a los posibles clientes llamados “potenciales”, aunque con estos últimos el proceso de organización sea más complejo ya que se desconoce de ellos.

**Concertación de la visita:** Se efectúa el contacto con los clientes mediante un canal de comunicación acorde a ellos. Normalmente se realizan llamadas por teléfono, e-mail o se agenda una reunión preliminar.

**Contacto y presentación:** Se realiza el intercambio de datos de lo que se desea ofrecer con el fin de generar interés.

**Sondeo y necesidades:** Se realiza una búsqueda de las exigencias del cliente para escoger la forma de realizar una venta.

**Argumentación:** Haber llegado a esta etapa significa que el cliente ha generado interés y se debe realizar una explicación completa de las ventajas y aportes que ofrece el producto o servicio por sobre la competencia.

**Objeciones:** En esta fase el cliente puede presentar dudas respecto al producto ofrecido, es aquí donde el convencimiento del vendedor debe actuar de la mejor manera para absolver dichas dudas.

**Cierre:** En esta etapa se puede dar por finalizada la venta, puede suceder que el proceso no se cierre por diferentes factores o que el mismo cliente indique no cerrarla. (p.10-11).

En ese sentido, la dimensión seleccionada para el desarrollo del presente estudio es la de Cierre del proceso de ventas, en donde procedemos a argumentar de manera teórica los siguientes indicadores:

**Indicador: Calidad de los pedidos generados:** Mora (2015) refiere que, el indicador permite la medición del número de pedidos que se han generado en los tiempos acordados entre el cliente y la empresa, y su medición se realiza dividiendo el número de pedidos que se han entregado en los tiempos acordados entre el total de pedidos que se han solicitado. A continuación se detalla en la fórmula:

Fórmula:

**Figura 3: Fórmula calidad de pedidos generados**

$$CPG = \frac{NPET}{NTPS}$$

CPG: calidad de pedidos generados

NPET: número de pedidos entregados a tiempo

NTP: número total de pedidos solicitados

**Ingresos (I):** Según Wertz y Tran, (2015) es aquel valor que recibe la empresa por facilitarle conexiones o ventas a los vendedores. Además es un valor que viene en forma de tarifas por cada transacción, cotización u oferta de servicios que ofrece el vendedor. Se calcula multiplicando el valor bruto de mercadería por la tasa de comisión, en donde el valor bruto de mercadería es el número de transacciones por el valor promedio de pedidos.

Fórmula:

**Figura 4: Fórmula de Ingresos**

$$I = VBM * TC$$

I: Ingresos

VBM: Valor bruto de mercadería

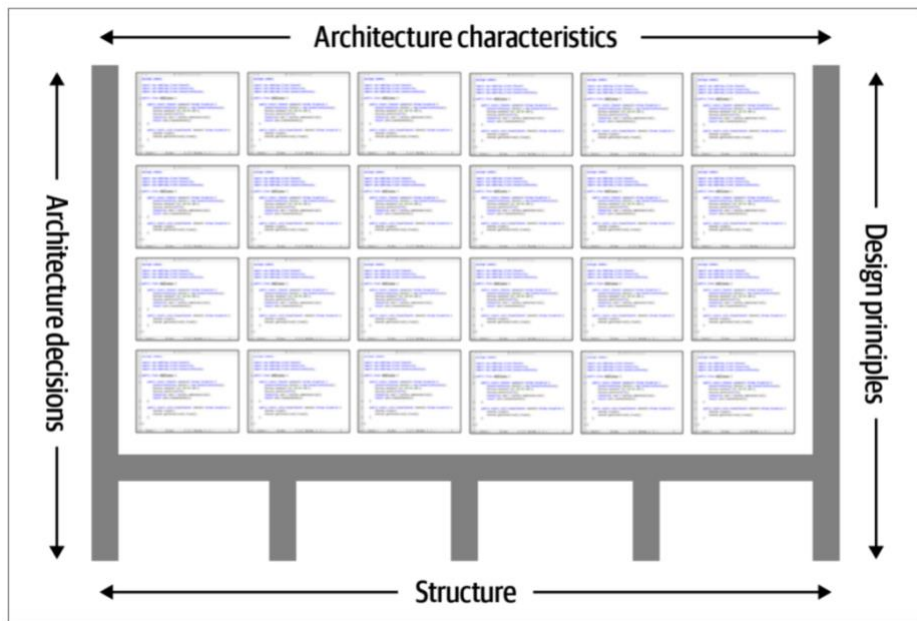
TC: Tasa de comisión

Ambos indicadores permitirán medir de qué manera influye la implementación de la variable independiente sobre la variable dependiente, esperando que se acepte la hipótesis alterna y de esta manera demostrando que la solución es la correcta para la problemática actual.

A continuación se definen otros conceptos relacionados: En primer lugar **Arquitectura de Software**, Para Neal (2020) La industria en su conjunto está tratando de definir una "arquitectura de software" con precisión. Algunos diseñadores llaman a la arquitectura de software un modelo de sistema, mientras que otros la definen como una hoja de ruta de desarrollo del sistema. El problema con estas definiciones comunes es averiguar qué contiene realmente el mapa o la ruta. Por ejemplo, ¿qué analizan los arquitectos cuando analizan las obras de construcción? La siguiente figura muestra una forma de pensar sobre la

arquitectura de software. Una arquitectura de software se define como la estructura del sistema (representada por líneas negras oscuras que soportan la arquitectura), las características de la arquitectura que debe soportar el sistema, la determinación de la arquitectura y finalmente los principios de diseño.

**Figura 5:Arquitectura de software**



Respecto a los lenguajes de programación primero se define el lenguaje PHP según la página oficial de PHP (2021) **PHP**: (preprocesador de hipertexto) es un lenguaje de secuencia de comando extremadamente popular que se emplea para la creación de páginas web. Combinando la sintaxis de los lenguajes C, Java y Perl, el código PHP está incrustado en las páginas HTML para la ejecución del lado del servidor. Se usa comúnmente para extraer datos de internet y poder presentarlos en la página web. Originalmente conocido como "Página de inicio personal", PHP es compatible con todos los servidores web y se usa ampliamente con la base de datos MySQL. Consulte MySQL.

El segundo lenguaje es **Java Script** que según Luna (2019) es un lenguaje de programación liviano utilizado mayormente en entornos web. Su implementación permite que los scripts del lado del navegador, llamado Cliente, interactúe con el usuario final, creando páginas web dinámicas. Java Script es un lenguaje interpretado con capacidades orientadas a objetos.

Se utilizará el **Framework Laravel**, que según su portal oficial (2021) menciona que Laravel es un software con una sintaxis expresiva y elegante. Nos indican también, que el desarrollo debe ser una experiencia creativa y placentera para obtener resultados satisfactorios. Esta herramienta pretende aliviar los problemas que suelen haber en el desarrollo, de esta manera toda tarea común utilizada en los diversos proyectos, como la autenticación, enrutamiento, manejo de sesiones y el almacenamiento en caché.

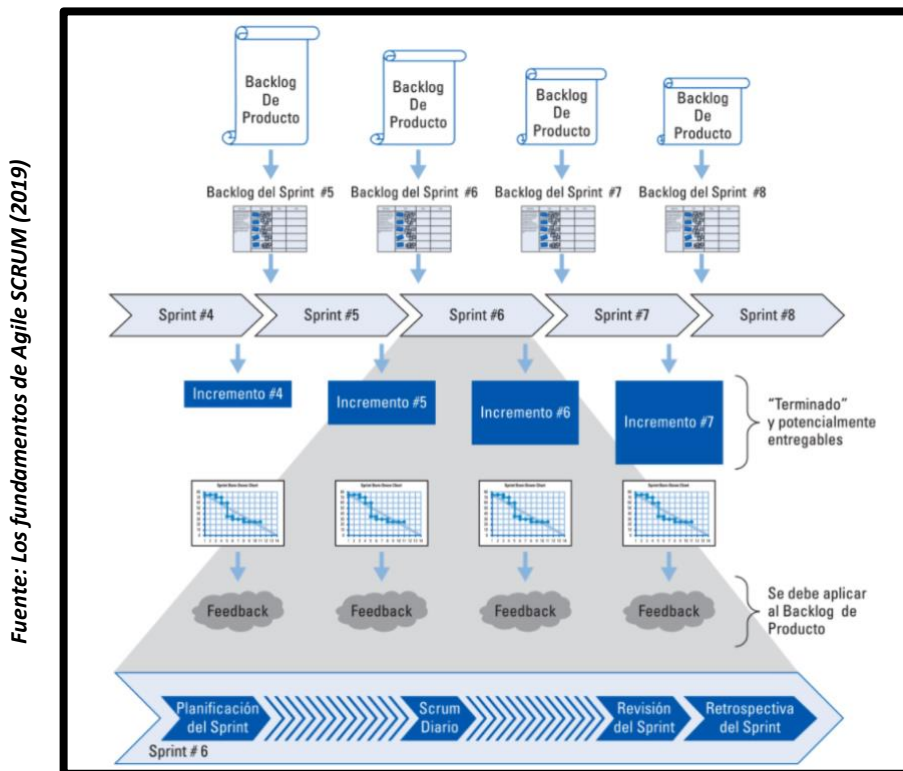
Y también para el desarrollo se utilizará **React JS**, que para Code Cademy (2021) menciona que ReactJS ofrece soluciones elegantes para algunos de los problemas más persistentes de la programación de front-end, lo que le permite crear aplicaciones web dinámicas e interactivas con facilidad. Es rápido, escalable, flexible, potente y cuenta con una sólida comunidad de desarrolladores que está creciendo rápidamente. Nunca ha habido un mejor momento para aprender React.

Respecto a la base de información, se empleará el MYSQL que según el portal oficial de MYSQL (2021) define que MYSQL es una herramienta de tipo racional que guarda todo dato en tablas aisladas en lugar de almacenar todo en un determinado lugar. Internamente, la estructura de la base de información se organiza en archivos físicos ordenados para tener una mayor velocidad al momento de consultarlos. Además, MySQL configura reglas para gobernar las conexiones entre diversos campos de datos, entre uno y varios, siendo obligatorio, único y opcional entre las diversas tablas. De esta manera la base de información permite que se cumpla con toda regla, de modo que si la base de datos fue diseñada de manera correcta, la aplicación no tendrá inconsistencias, datos duplicados, datos huerfanos o faltantes.

Durante la elaboración de la presente investigación se consideraron 3 metodologías de desarrollo:

**SCRUM:** Según Nader K. y Turley, (2019) es un marco de trabajo que da soporte a situaciones complicadas y puede tener iteraciones llamados "Sprints" hasta que: Se terminan todos los ítems del Backlog; El cliente se da cuenta que con el último incremental es capaz de cerrar el proyecto y no existe justificación para invertir más tiempo añadiendo funciones.

**Figura 6: Marco de trabajo SCRUM**



SCRUM está dividido por 3 roles, los cuales son: **Product Owner:** También llamado dueño del producto y persona responsable de darle valor al proyecto, es el principal encargado de gestionar el Backlog y mantener una comunicación constante con el cliente, desarrolladores, stakeholders, etc; **SCRUM Master:** Es el sujeto encargado de asegurarse que se cumpla el marco de SCRUM de forma correcta, realización la formación, acompañamiento “coaching” y debe contar con la capacidad de resolver problemas; **Development Team:** Es el equipo técnico y a su vez perfiles multifuncionales con la capacidad de autogestionarse y los que son llamados a desarrollar el producto final. (p.18-19)

**XP:** Según Aladeen, Ali, y Ali Khamees, (2021) fue concebido por Kent Beck y Martin Fowler, el nombre refleja la idea de que los grupos de desarrollo puedan tomar buenas prácticas en la ingeniería de software. Este marco de trabajo enfatiza que el cliente pueda tener satisfacción a través del desarrollo rápido de un producto de alto valor. La metodología se divide en iteraciones de 1 a 4 semanas como máximo. (p.4) Según Ruiz, (2017) las fases de XP son: **Exploración:** Aquí se define

el alcance del proyecto normalmente dura dos semanas y es aquí donde el cliente define su necesidad mediante la redacción de lo que se llaman las Historias de Usuario. El equipo de desarrollo estima los tiempos de ejecución de estas tareas las cuales pueden ser variables en cada iteración; **Planificación:** Esta fase es corta, ya que los dueños del producto así como el equipo de desarrollo coordinan el orden de la ejecución de las Historias de usuario, teniendo como entregable el documento *Release Plan* (Plan de entregas); **Iteraciones:** Es la base esencial de la metodología XP ya que aquí es donde se construye cada funcionalidad solicitada y se ejecutan las iteraciones, donde en cada iteración se presenta un entregable funcional, al inicio de cada iteración se analizan las actividades y si es necesario se despejan las dudas con el cliente; **Puesta en producción:** Como bien se menciona en la fase anterior se realiza la entrega de un producto funcional y sin errores, es importante que el cliente tenga como conocimiento que el sistema no se pondrá en producción hasta que todo esté funcionalmente completado, en ese sentido, pueden existir fases necesarias, llamadas *fine tuning* (tareas de ajuste). (p.281).

**RUP:** Según Hashmi y Ahmed, (2020), es una metodología interactiva para llevar acabo el desarrollo del software basada en la arquitectura y casos de uso. La metodología RUC se basa en el lenguaje de modelado unificado (UML). Por otro lado, nos indican que la Metodología RUP tiene puntos de control clave o críticos a través del desarrollo. En otras palabras, cada fase de la metodología RUP debe terminar con algún control, puntos clave o críticos donde se suman los resultados obtenidos y se planifican las direcciones futuras en base a los resultados. Finalmente, nos mencionan que la metodología RUP tiene 4 principales fases: **Inicio:** En este primer punto definimos el alcance de los proyectos con el cual la principal fase es definida en mérito al proyecto con los principales stakeholders y son identificados al riesgo asociado de proyectos de desarrollo; **Elaboración:** Se realiza el diseño de la solución y se define el caso de usos lo cual permite definir las bases del sistema y así tener un dominio de la problemática a solucionar; **Desarrollo y construcción:** Se enfoca en la elaboración del producto en base a los casos de uso propuestos en la fase anterior; **Transición:** En esta fase se realiza el cierre del proyecto y tiene como objetivo asegurar que se está entregando un software funcional a los usuarios finales.



Luego de analizar las metodologías de desarrollo para un Marketplace, se realizó la siguiente tabla la cual establece los criterios de evaluación de las mismas:

**Tabla 1: Criterios de evaluación de las metodologías de desarrollo**

Ítem	Criterio	Descripción
1	Hitos de desarrollo corto	Las fases de desarrollo pueden ser cortos (De 7 a 14 días) según el tipo del proyecto
2	Resultado incremental	Los resultados se pueden visualizar en cada fase del proyecto
3	Personas involucradas	El proyecto puede llevarse a cabo con un equipo pequeño de personas
4	Presupuesto bajo asignado para el proyecto	Se realiza una mínima inversión con tal de conseguir lo necesario para el proyecto
5	Dar valor al cliente	El cliente puede percibir el valor del producto en cada fase del proyecto
6	Resultados rápidos	Son visibles en corto tiempo
7	Se adapta a los cambios	Se ajusta a los diferentes cambios que puedan suceder bajo alguna coyuntura en específico
8	Escala del proyecto pequeño	Los hitos se pueden llevar a cabo en el tiempo establecido

Fuente: Elaboración propia

Con dichos criterios se realizó la evaluación de expertos correspondiente al tipo de metodología de desarrollo a emplear en el presente estudio.

**Tabla 2: Validación de expertos para la aplicación de la metodología de desarrollo**

Experto	Grado académico	Valoración de la metodología			
		SCRUM	XP	RUP	Elección
Iván Perez Farfán	Mgtr. En Ingeniería de Sistemas	23	22	16	SCRUM
Hilario Aradiel Castañeda	Ing. de Sistemas	24	16	8	SCRUM

Guillermo Johnson Romero	MSc en Ingeniería de Sistemas	23	17	14	SCRUM
<b>Promedio</b>		70	55	38	SCRUM

Fuente: Elaboración propia

En la validación de expertos se obtuvo que la puntuación más alta de las 3 metodologías propuestas fue el marco de SCRUM, dado que obtuvo una puntuación de 70 por parte de los tres especialistas en la materia. De esta manera, la metodología SCRUM será utilizada en el proceso de desarrollo del Marketplace.

### III. METODOLOGÍA

#### 3.1. Tipo y diseño de investigación

##### **Tipo de investigación:**

El tipo de estudio es aplicado, al respecto de esta clasificación cabe señalar que se llaman aplicadas, puesto que se basa en el resultado obtenido de la investigación, se formula un problema e hipótesis para solucionar (Ñaupas, Valdivia, Palacios, & Romero, 2018, p 136) refieren que la investigación es aplicada, porque se busca dar solución a una problemática, en este caso sobre la realidad empresarial planteada en esta investigación.

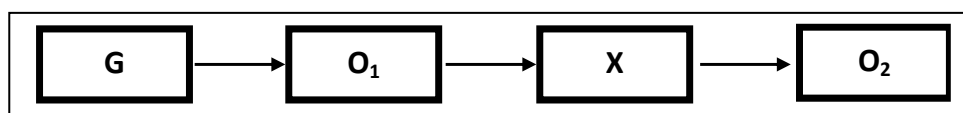
En tanto que, según el enfoque es cuantitativo ya que se emplea el análisis y la recolección de información de estos para responder a la pregunta de estudio buscando probar la hipótesis formulada anteriormente. De igual forma, se confía en la medición de la variable e instrumento del estudio (Ñaupas et. al., 2018, p 140). Para esta investigación el enfoque será cuantitativo porque pretende recolectar datos para poder probar la hipótesis con análisis estadístico y base numérica.

Asimismo, el nivel de esta investigación es explicativa, son aquellos estudios en los que se busca como objetivo el establecimiento de la causa, problema, suceso o fenómeno que se estudia (Hernández & Mendoza, 2018). El presente estudio es explicativo dado que busca determinar las causas o factores que originan el problema materia de investigación.

### Diseño del estudio:

Por último, en cuanto al diseño de estudio es experimental, puesto que se manipulará de manera intencional el objeto de investigación (variable independiente), para observar y analizar sus efectos (variable dependiente) (Silvestre y Huamán, 2019). En tanto el diseño de investigación corresponde a ser pre experimental, puesto que se manipula la variable independiente para ver los efectos en la variable dependiente analizada. El diseño pre-experimental se puede visualizar de manera gráfica en la siguiente figura:

**Figura 7: Diseño de estudio**



### Dónde:

**G:** Grupo experimental, es aquel grupo de población que se van a estudiar para realizar las pruebas de Pre y Post.

**X:** Implementación de la solución, la cual vendría a ser el Marketplace.

**O<sub>1</sub>:** Resultado de pruebas que se ejecutan sin el Marketplace

**O<sub>2</sub>:** Resultado de pruebas que se ejecutan con el Marketplace

### 3.2. Variables y operacionalización

En primer lugar, se obtuvo la definición conceptual de la variable independiente (VI), la cual es Marketplace, la cual se define como plataformas digitales que permiten la interacción entre grupos interdependientes de usuarios llamados “vendedores y clientes” los cuales realizan el intercambio de bienes mediante diferentes estrategias comerciales en donde influye como principal factor el precio del producto o servicio, esto según Fruhwirth, Rachinger, y Prlja (2020).

Por otro lado, se obtuvo la definición conceptual de la variable dependiente (VD), la cual es Proceso de ventas, definiéndose como una secuencia de pasos que un comercio realiza para generar una negociación con un potencial cliente y que tiene como objetivo generar una acción deseada, es decir, la compra, esto según Kotler y Armstrong, (2016).

Así también, se realizó la definición operacional de la variable independiente (VI), la cual es Marketplace y se define como una plataforma tecnológica que permite la interacción entre diferentes sectores sociales para la realización del intercambio de algún producto o servicio, permitiendo automatizar el proceso de ventas de los diferentes comercios que lo conforman.

Finalmente, se tuvo la definición operacional de la variable dependiente (VD), el cual es Proceso de ventas y se define como una serie de pasos que realiza un comercio para llegar a un fin en concreto, que es la venta de productos o servicios, estos pasos son: Preparación, Concertación, Contacto, Sondeo, Argumentación, Objeciones y Cierre.

**Tabla 3: Operacionalización de variables**

Variable	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Escala de medición
Marketplace	El Marketplace permite a los vendedores ofrecer sus productos sin tener que realizar una configuración adicional, además de brindar oportunidades comerciales. (Mohammad y Allah, 2020)	Permite a los vendedores mostrar sus productos para obtener oportunidades comerciales.			
Proceso de ventas	La venta está representada por un proceso y diferentes autores coinciden en la importancia de estas fases las cuales son: Preparación, Concertación, Contacto, Sondeo, Argumentación, Objeciones y Cierre. (Acosta, Salas, Guerra, & Jiménez, 2018, p.11)	El proceso de ventas es analizado mediante las siguientes dimensiones: Margen comercial, conversión de ventas.	Cierre	Calidad de pedidos generados	De razón
				Ingresos	De razón

Fuente: Elaboración propia

**Tabla 4: Descripción de indicadores**

Dimensión	Indicador	Descripción	Técnica	Instrumento	Unidad de medida	Fórmula
Cierre	Calidad de los pedidos generados	Mora (2015) este indicador permite medir el número de pedidos entregados en los tiempos acordados entre la empresa y el cliente, y se mide dividiendo el número de pedidos entregados a tiempo entre el total de pedidos solicitados.	Fichaje	Ficha de registro	Porcentaje	$CPG = \frac{NPET}{NTPS}$ <p>CPG: calidad de pedidos generados                      NPET: número de pedidos entregados a tiempo                      NTP: número total de pedidos solicitados</p>
	Ingresos	Según Wertz y Tran, (2015) es aquel valor que recibe la empresa por facilitarle conexiones o ventas a los vendedores del Marketplace. Además es un valor que viene en forma de tarifas por cada transacción, cotización u oferta de servicios que ofrece el vendedor.	Fichaje	Ficha de registro	Unidad	$I = VBM * TC$ <p>I = Ingresos                      VBM = Valor bruto de mercadería                      TC = Tasa de comisión</p>

Fuente: Elaboración propia

### 3.3. Población, muestra y muestreo

Según Rodríguez, Pierdant y Rodríguez (2016, p. 144), nos indican que la población es un grupo conformado por una cantidad determinada o indeterminada de elementos, los cuales pueden ser personas, fenómenos, objetos, etc. Los cuales presentan semejanzas en algunas características hacia un objeto de estudio. Por otro lado, Sánchez y Solís, (2016, p. 5), nos indican que un grupo de animales u objetos los cuales son sometidos a un estudio estadístico se denomina población. En ese sentido, la presente investigación tiene una población de 1500 registros de ventas obtenidos en 30 días, dichos datos se utilizarán en los indicadores Valor Bruto de Mercadería e Ingresos.

**Tabla 3: Determinación de la Población**

Población	Tiempo	Indicador
1500 pedidos	1 mes	Calidad de los pedidos generados
1500 registro de ventas	1 mes	Ingresos

Fuente: Elaboración Propia

Así también, según Sánchez y Solís (2016, p.144), “Cuando la población es muy numerosa, los datos suelen recogerse de una parte de esta denominada muestra”. Por otro lado, Rodríguez, Pierdant y Rodríguez (2016, p.6), nos indican que la muestra está formada por elementos de una población estudiada (N). Debiendo cumplir ciertos requisitos los cuales son: Los elementos conformados deben pertenecer a la población de estudio y estos deben ser extraídos a través de la técnica de muestreo.

En la presente investigación, se tuvo en cuenta como nivel de confianza un 95% y un margen de error del 5%, además se utilizará la fórmula que plantean Castejón, Lafuente y Faura (2015):

**Figura 8: Fórmula de la muestra**

$$n = \frac{z^2 N}{z^2 + 4N(EE^2)}$$

Dónde:

n = Tamaño de la muestra que se desea obtener

Z = Nivel de confianza al 95% (1.96)

N = Población total de estudio

EE = Error estimado (al 5%)

Se contó con 1500 registros de ventas como población, con dicho dato se procedió a calcular la muestra para ambos indicadores.

$$n = \frac{(1.96)^2 * 1500}{(1.96)^2 + 4 * 1500 * (0.05)^2}$$

$$n = \frac{3.8416 * 1500}{3.8416 + 1500(0.0025)}$$

$$n = 305.83 \cong 306$$

La muestra para ambos indicadores es de 306 unidades, las cuales representan las ventas o pedidos.

Rodríguez, Pierdant y Rodríguez (2016, p. 246), nos indican que existe una gran cantidad de técnicas de muestreo que se pueden aplicar para la resolución de un problema y con estas técnicas se pueden obtener los mismos resultados como si se utilizara toda la población, esto en un menor tiempo y utilizando cantidades menores de recursos financieros, humanos y materiales. El objetivo del muestreo es identificar inferencias de la población sin tener que estudiar cada elemento de la población.

La presente investigación usará el tipo de muestreo aleatorio simple, ya que, según Rodríguez, Pierdant y Rodríguez (2016, p.248), nos indican que el muestreo



aleatorio simple es generalizado y sencillo, dado que se usan dos técnicas básicas para la selección de los elementos: Tabla de números aleatorios y la urna. De esta manera, se pudo tomar en consideración que la población de estudio es finita y que los elementos tienen una elevada probabilidad de ser elegidos.

Además, Ruiz et al. (2020), detalla que en el tipo de muestreo probabilístico existe un muestreo denominado aleatoria estratificado, el cual detalla que la población se puede agrupar por segmentos los cuales se denominan estratos, y luego se genera una muestra aleatoria simple de estos estratos, generando una nueva muestra, la cual es la cantidad de estratos definidos.

Por tal motivo según Ruiz, la muestra para esta investigación será de 30 registros.

### **3.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad**

Según Gil (2016, p. 9), nos indica que la técnica como el método para la recolección de datos son empleados para registrar las observaciones o también facilitar el tratamiento los mismos.

En el presente estudio se utilizará la técnica del fichaje, donde Muñoz (2016, p. 109), nos indica que en las fichas y el fichaje existe mucha información donde nos menciona los principales tipos: De localización y de investigación.

Así también, para el instrumento de recolección de datos, Alamilla (2016) refiere que las fichas de registro sirven para recopilar datos de diversos recintos en una investigación.

La técnica de fichaje fue empleada en el presente proyecto para obtener la información de los indicadores.

Así también, García (2017) nos explica que la entrevista sirve como instrumento de recolección de datos para obtener información a partir de la interacción de dos partes, las cuales son el entrevistado y entrevistador. En este estudio se empleó la

entrevista como instrumento para la recolección de datos, la cual fue utilizada desde el inicio de esta para evidenciar cada inconveniente de la empresa AC Brands & Consulting E.I.R.L.

Respecto a la confiabilidad, según Guevara, Cárdena y Hernández (2017), nos indican que la confiabilidad de un instrumento.

**Tabla 4: Técnicas e Instrumentos de recolección de datos**

Indicador	Técnica	Instrumento	Informante
Calidad de pedidos generados	Fichaje	Ficha	Cliente
Ingreso	Fichaje	Ficha	Cliente

Fuente: Elaboración propia

Las fichas de recolección de datos fueron valoradas y analizadas por expertos, realizando una valoración sobre el adecuado uso de los instrumentos empleados en el estudio, A continuación, se detalla la puntuación dada por los expertos:

**Tabla 5: Validación de expertos del Instrumento de Investigación**

Experto	Puntuación	
	Calidad de pedidos generados	Ingreso
Iván Perez Farfán	70%	70%
Hilario Aradiel Castañeda	80%	80%
Guillermo Johnson Romero	74%	74%
<b>TOTAL</b>	75%	75%

Fuente: Elaboración Propia

### Confiabilidad

Según Carrasco (2015) “El método de test retest también es conocida como la aplicación repetida, esta consiste en realizar la aplicación de una prueba de manera

sucesivas en el instrumento para poder determinar que las respuestas que se obtienen son estables y confiables.

### **Método Test – Retest**

Según Carrasco (2015), menciona que los factores de confiabilidad de las pruebas se pueden definir como correlaciones entre los puntajes de las pruebas y los autoinformes. Por tanto, una posible forma de obtener un estímulo para este valor es aplicar la prueba a una muestra de sujetos en dos casos diferentes. De esta forma puede calcular la correlación.

### **Técnica**

Coeficiente de correlación de Pearson: Correa y Barrera. (2016), manifiestan que el coeficiente de correlación de Pearson da solución al problema anterior porque no depende de la unidad de medida de la variable ni del rango de valores de -1 a 1. De hecho, el coeficiente de correlación de Pearson es una varianza compleja. Está estandarizado. Los valores cercanos a 0 indican que no hay relación lineal, los valores cercanos a 1 indican una relación lineal directa muy fuerte y los valores cercanos a -1 indican una relación lineal inversa. Si el valor del coeficiente de correlación es exactamente 1 o -1, esto indica una relación perfectamente lineal, directa o viceversa, respectivamente. En otras palabras, cada punto de la nube de puntos forma una línea completa.

La fórmula se muestra a continuación:

**Figura 9: Coeficiente de correlación de Pearson**

Fuente: Correa y  
Barrera (2016)

$$\begin{array}{l} \text{Población: } \rho_{xy} = \frac{\delta xy}{\delta x \cdot \delta y} \\ \text{Muestra: } r_{xy} = \frac{S_{xy}}{S_x \cdot S_y} \end{array}$$

Se definen niveles de confiabilidad, para poder validar que la ficha de recolección de datos mantenga una confianza aceptable para poder ser implementada en la recolección de información, estos niveles se detallan en la siguiente tabla:

**Tabla 6: Niveles de Confiabilidad**

Escala	Nivel
0.00 < sig. < 0.20	Muy bajo
0.20 ≤ sig. < 0.40	Bajo
0.40 ≤ sig. < 0.60	Regular
0.60 ≤ sig. < 0.80	Aceptable
0.80 ≤ sig. < 1.00	Elevado

Fuente: Correa y Barrera (2016)

La regla para medir el nivel de confiabilidad es que mientras el valor de significancia sea más cercano a uno quiere decir que la confianza de la ficha es mayor, debería tener por lo menos 1 valor mayor o igual a 0.6 puntos para poder decir que la ficha es confiable.

Se realizó la prueba de test y retest para los indicadores ingresos y calidad de pedidos generados, en donde para el indicador ingreso se obtuvo los siguientes resultados:

**Tabla 7: Correlación Calidad de pedidos generados**

Correlaciones			
		Calidad_pedidos_test	Calidad_pedidos_retest
Calidad_pedidos_test	Correlación de Pearson	1	,932**
	Sig. (bilateral)		,000
	N	30	30
Calidad_pedidos_retest	Correlación de Pearson	,932**	1
	Sig. (bilateral)	,000	
	N	30	30

\*\* . La correlación es significativa en el nivel 0,01 (bilateral).

Así como detalle en la tabla anterior, el nivel de significancia para ambos tiempos de evaluación es de 0.932, lo cual valida que el nivel de confianza de la ficha es elevado.

**Tabla 8: Correlación Ingresos**

Correlaciones			
		Ingresos_Test	Ingresos_Retest
Ingresos_Test	Correlación de Pearson	1	,981**
	Sig. (bilateral)		,000
	N	30	30
Ingresos_Retest	Correlación de Pearson	,981**	1
	Sig. (bilateral)	,000	
	N	30	30

\*\* . La correlación es significativa en el nivel 0,01 (bilateral).

Así como detalle en la tabla anterior, el nivel de significancia para ambos tiempos de evaluación es de 0.981, lo cual valida que el nivel de confianza de la ficha es elevado.

### **3.5. Procedimientos**

El procedimiento que se empleará para poder realizar esta investigación en primer lugar será la recolección de información a partir de las entrevistas realizadas con el personal de la empresa, de tal manera que se puedan completar las fichas de registro para posteriormente realizar la programación y desarrollo del software, y luego realizar su implementación y con el mismo poder medir los indicadores, concluyendo que el software genere reporte de cada uno de estos, para luego poder realizar la comparación del tiempo antes de la implementación del Marketplace y el tiempo después de la implementación del Marketplace.

**Tabla 9: Procedimientos de recolección de datos**

Datos generales				
<b>Organización</b>	AC BRANDS & CONSULTING E.I.R.L.			
<b>Coordinación</b>	Administrador			
<b>Recolección</b>	Proceso de ventas			
Especificaciones				
Indicador	Técnica	Instrumento	Fuente	Informante
Calidad de pedidos generados	Fichaje	Ficha de registro	Recolección de la misma empresa	Cliente
Ingresos	Fichaje	Ficha de registro	Recolección de la misma empresa	Cliente

Fuente: elaboración propia

### 3.6. Método de análisis de datos

El método de análisis de datos tiene como función principal definir tanto la hipótesis alterna como la hipótesis nula, para luego de esto poder rechazar la hipótesis nula y aceptar la alterna, validando de esta forma que la solución realmente efectiva para la problemática que actualmente se tiene.

**H1:** El Marketplace aumenta la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L.

**Indicador: Calidad de pedidos generados**

**Dónde:**

CPG<sub>a</sub>: Calidad de pedidos generados antes del sistema web

CPG<sub>d</sub>: Calidad de pedidos generados después del sistema web

**Hipótesis Nula H1<sub>0</sub>:** El Marketplace no aumenta la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L.

$$H1_0: CPG_a - CPG_d \leq 0$$

$$H1_0: CPG_a > CPG_d$$

**Hipótesis Alternativa H1a:** El Marketplace aumenta la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L.

$$H1_a: CPG_a - CPG_d > 0$$

$$H1_a: CPG_d > CPG_a$$

**H2:** El Marketplace incrementa los **ingresos** en el proceso de ventas de AC Brands & Consulting E.I.R.L.

**Indicador: Ingresos**

**Dónde:**

**SP<sub>a</sub>:** Ingresos antes del sistema web.

**SP<sub>d</sub>:** Ingresos después del sistema web.

**Hipótesis H2<sub>0</sub>:** El Marketplace no incrementa los ingresos en el proceso de ventas de AC Brands & Consulting E.I.R.L.

$$H1_0: I_d - I_a \leq 0$$

$$H1_0: I_d > I_a$$

**Hipótesis H2<sub>a</sub>:** El Marketplace no incrementa los ingresos en el proceso de ventas de AC Brands & Consulting E.I.R.L.

$$H1_a: I_d - I_a > 0$$

$$H1_a: I_a > I_d$$

El nivel de significancia usado fue de  $\alpha = 5\%$  siendo este último valor el margen de error el cual equivale a 0.05, permitiendo realizar la comparación para lograr tomar la decisión de aceptar o rechazar la hipótesis.

- Nivel de confiabilidad:  $(1-\alpha) = 0.95$
- Margen de error  $\alpha = 0.05$

En ese sentido, se utilizó como método de análisis de datos, la prueba T de Student. Donde según Galindo (2020), nos indica que la T de Student de una prueba que ayuda a estimar los valores poblacionales a partir de los datos muestrales, ayudando a pronosticar la probabilidad de 2 promedios pertenezcan a una misma población o que sean parte de distintas poblaciones. La prueba T de Student se expresa en la siguiente fórmula:

**Figura 10: Distribución de T de Student**

$$t = \frac{\bar{x} - \mu}{\frac{s_x}{\sqrt{n}}}$$

**Dónde:**

Grados de libertad =  $df = n - 1$

$\bar{x}$  = Media

$\mu$  = Valor a analizar

$s_x$  = Desviación estándar

$n$  = Tamaño de la muestra

Región de rechazo es  $t = t_x$

Donde  $t_x$  es tal que:

$P [t > t_x] = 0.05$ , donde  $t_x$  = Valor tabular

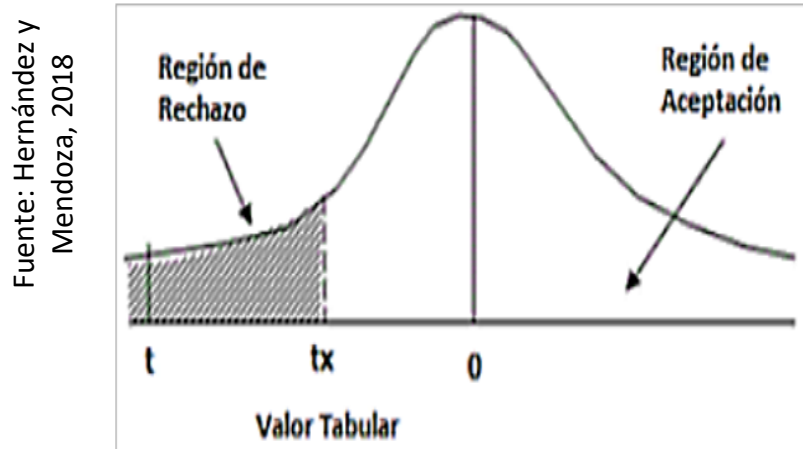
Luego, la región de rechazo:  $t > t_x$

La T de Student realiza el cálculo científico el cual nos va a permitir validar la existencia de separación sobre 2 muestras usando los promedios como punto de estudio.



En la figura 11, se muestra es gráfico de la repartición T de Student mostrar el área rechazada y el área aceptada además del valor de t identificado en su ubicación en el trazado.

Figura 11: Distribución de T de Student



Fuente: Hernández y  
Mendoza, 2018

Asimismo, se contó con cada rango de distribución T de Student, mostrando su grado de libertad y valor correspondiente, esto se pudo observar en la figura 12.

Figura 12: Rango de valores de la distribución T de Student

n l a	0,30	0,25	0,20	0,10	0,05	0,025	0,01	0,005	0,0025	0,001	0,0005
1	0,7265	1,0000	1,3764	3,0777	6,3137	12,7062	31,8210	63,6559	127,3213	318,3088	636,6192
2	0,6172	0,8185	1,0607	1,8856	2,9200	4,3027	6,9645	9,9250	14,0690	22,3271	31,5991
3	0,5844	0,7649	0,9785	1,6377	2,3534	3,1824	4,5407	5,8408	7,4533	10,2145	12,9240
4	0,5686	0,7407	0,9410	1,5332	2,1318	2,7765	3,7489	4,6041	5,5976	7,1732	8,5103
5	0,5594	0,7267	0,9195	1,4759	2,0150	2,5706	3,3649	4,0321	4,7733	5,8934	6,9688
6	0,5534	0,7176	0,9057	1,4398	1,9432	2,4469	3,1427	3,7074	4,3168	5,2076	5,9588
7	0,5491	0,7111	0,8960	1,4149	1,8946	2,3646	2,9979	3,4995	4,0293	4,7853	5,4079
8	0,5459	0,7064	0,8889	1,3968	1,8595	2,3060	2,8965	3,3554	3,8325	4,5008	5,0413
9	0,5435	0,7027	0,8834	1,3830	1,8331	2,2622	2,8214	3,2498	3,6897	4,2968	4,7809
10	0,5415	0,6998	0,8791	1,3722	1,8125	2,2281	2,7638	3,1693	3,5814	4,1437	4,5809
11	0,5399	0,6974	0,8755	1,3634	1,7959	2,2010	2,7181	3,1058	3,4966	4,0247	4,4370
12	0,5386	0,6955	0,8726	1,3562	1,7823	2,1768	2,6810	3,0545	3,4284	3,9296	4,3178
13	0,5375	0,6938	0,8702	1,3502	1,7709	2,1604	2,6503	3,0123	3,3725	3,8520	4,2208
14	0,5366	0,6924	0,8681	1,3450	1,7613	2,1448	2,6245	2,9768	3,3257	3,7874	4,1405
15	0,5357	0,6912	0,8662	1,3406	1,7531	2,1315	2,6025	2,9467	3,2860	3,7328	4,0728
16	0,5350	0,6901	0,8647	1,3368	1,7459	2,1199	2,5835	2,9208	3,2520	3,6862	4,0150
17	0,5344	0,6892	0,8633	1,3334	1,7396	2,1098	2,5689	2,8982	3,2224	3,6456	3,9651
18	0,5338	0,6884	0,8620	1,3304	1,7341	2,1009	2,5524	2,8784	3,1966	3,6105	3,9216
19	0,5333	0,6876	0,8610	1,3277	1,7291	2,0930	2,5395	2,8609	3,1737	3,5794	3,8834
20	0,5329	0,6870	0,8600	1,3253	1,7247	2,0860	2,5280	2,8453	3,1534	3,5518	3,8495
21	0,5325	0,6864	0,8591	1,3232	1,7207	2,0796	2,5176	2,8314	3,1352	3,5272	3,8193
22	0,5321	0,6858	0,8583	1,3212	1,7171	2,0739	2,5083	2,8188	3,1188	3,5050	3,7921
23	0,5317	0,6853	0,8575	1,3195	1,7139	2,0687	2,4999	2,8073	3,1040	3,4850	3,7676
24	0,5314	0,6848	0,8569	1,3178	1,7109	2,0639	2,4922	2,7970	3,0905	3,4668	3,7454
25	0,5312	0,6844	0,8562	1,3163	1,7081	2,0595	2,4851	2,7874	3,0782	3,4502	3,7251
26	0,5309	0,6840	0,8557	1,3150	1,7056	2,0555	2,4786	2,7787	3,0669	3,4350	3,7068
27	0,5306	0,6837	0,8551	1,3137	1,7033	2,0518	2,4727	2,7707	3,0565	3,4210	3,6896
28	0,5304	0,6834	0,8546	1,3125	1,7011	2,0484	2,4671	2,7633	3,0469	3,4082	3,6739
29	0,5302	0,6830	0,8542	1,3114	1,6991	2,0452	2,4620	2,7564	3,0380	3,3962	3,6594
30	0,5300	0,6828	0,8538	1,3104	1,6973	2,0423	2,4573	2,7500	3,0298	3,3852	3,6460

Fuente: Hernández y  
Mendoza, 2018

### **3.7. Aspectos éticos**

En el presente proyecto de investigación se tuvo en cuenta cada lineamiento establecido por la Universidad César Vallejo Lima Norte, donde se cumplieron las políticas para la elaboración del presente documento.

Además, el investigador garantiza que esta investigación no ha sido copiada ni de manera parcial ni total, siendo de autoría propia.

Por otro lado, en el transcurso de la presente investigación se realizó una ardua búsqueda para la obtención de datos necesarios, los cuales se encuentran libre de plagios, alteraciones o cambios, en ese sentido, la recolección se realizó de manera correcta respetando la política de transparencia y confiabilidad de la información.

Finalmente, el investigador garantiza que todo el contenido extraído de libros externos ha sido citado correctamente y referenciado en la bibliografía.

## IV. RESULTADOS

### Análisis descriptivo

Para realizar el análisis estadístico de los resultados, se deben realizar tres pasos fundamentales, el primero es el análisis descriptivo, en donde se hace la comparación general entre los dos tiempos de evaluación antes de la implementación del sistema y después de la implementación del sistema, para luego proseguir con la prueba de normalidad y saber si los resultados se distribuyen de manera normal o no normal. Para finalmente pasar a la prueba hipótesis la cual pretende rechazar la hipótesis nula y aceptar hipótesis alterna.

### Calidad de pedidos

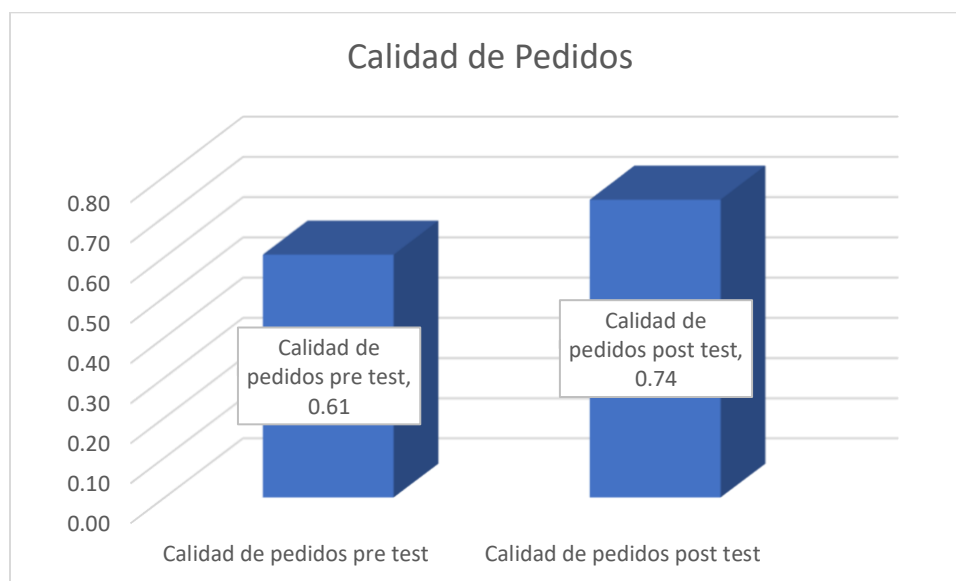
**Tabla 10: Análisis descriptivo del indicador Calidad de pedidos**

Estadísticos descriptivos					
	N	Mínimo	Máximo	Media	Desv. Desviación
Calidad_pedidos_pre	30	,42	,89	,6060	,12073
Calidad_pedidos_post	30	,50	,89	,7430	,10761
N válido (por lista)	30				

En el análisis descriptivo anterior se puede visualizar que existe un aumento de los resultados entre los dos tiempos de evaluación, en el primer tiempo el promedio fue de 0,6 o 60.6%, en el segundo fue de 0.74 o 74.3%. Para el primer tiempo el valor mínimo fue de 42%, el máximo de 89% y la desviación estándar de 12%. En el segundo tiempo el valor mínimo fue de 50% y el máximo de 89% con una desviación estándar de 10%. Con estos resultados a simple vista se puede validar la efectividad del sistema para este indicador.

En la siguiente figura se puede visualizar un histograma para tener una visión más general sobre como se generó el aumento antes de la implementación del sistema hasta después de su implementación.

**Figura 13: Histograma del análisis descriptivo del indicador Calidad de pedidos**



En esta figura se puede visualizar el aumento del indicador de un 60.6% a un 74.3%, lo que equivale a un 13% de aumento.

## Ingresos

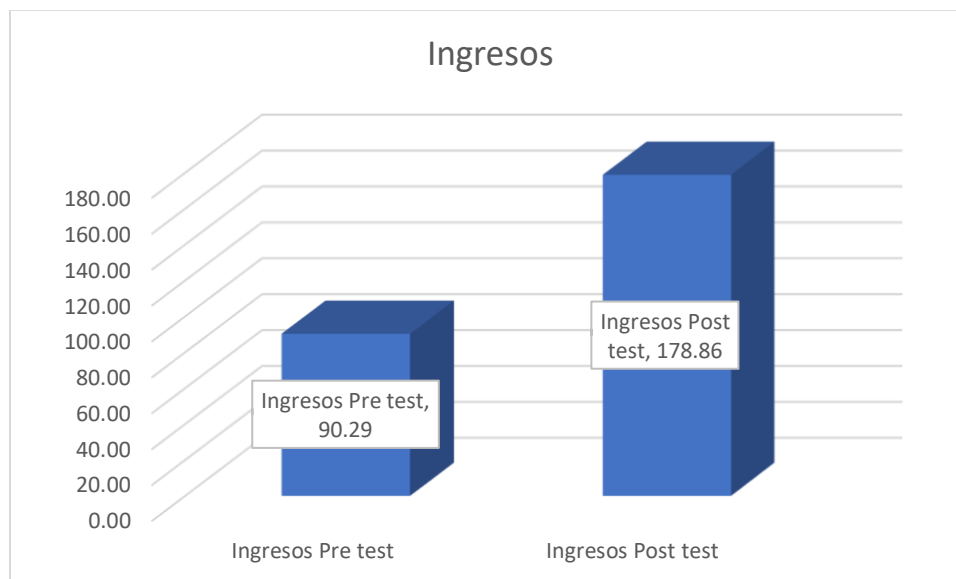
**Tabla 11: Análisis descriptivo del indicador Ingresos**

Estadísticos descriptivos					
	N	Mínimo	Máximo	Media	Desv. Desviación
Ingresos_pre	30	64,86	120,00	90,2903	15,66565
Ingresos_post	30	141,68	239,48	178,8607	25,53555
N válido (por lista)	30				

En el análisis descriptivo anterior se puede visualizar que existe un aumento de los resultados entre los dos tiempos de evaluación, en el primer tiempo el promedio fue de 90.29 en el segundo fue de 178.86. Para el primer tiempo el valor mínimo fue de 64.86 el máximo de 120.0 y la desviación estándar de 15.66%. En el segundo tiempo el valor mínimo fue de 141.68 y el máximo de 178.86 con una desviación estándar de 25%. Con estos resultados a simple vista se puede validar la efectividad del sistema para este indicador.

En la siguiente figura se puede visualizar un histograma para tener una visión más general sobre como se generó el aumento antes de la implementación del sistema hasta después de su implementación.

**Figura 14: Histograma del análisis descriptivo del indicador Ingresos**



En esta figura se puede visualizar el aumento del indicador de un 90.29 a un 178.86 lo que equivale a un 88.57 de aumento.

### **Prueba de normalidad**

La prueba de normalidad para Galindo (2020) se divide en dos grupos las pruebas normales o paramétricas y las pruebas no normales o no paramétricas. Si es que la muestra es pequeña o menor a 50 objetos, la prueba que se aplica será la de Shapiro-Wilk, de lo contrario se aplicará la prueba de Kolmogorov. Para poder determinar la distribución, se debe seguir la siguiente regla: si ambos valores del nivel de significancia tienen un valor mayor o igual a 0.05 entonces la distribución será de tipo normal, de lo contrario entonces la distribución será de tipo no normal.

Para ambos indicadores la muestra de evaluación fue de 30 objetos, los que representan los 30 días del mes, el mes de evaluación.

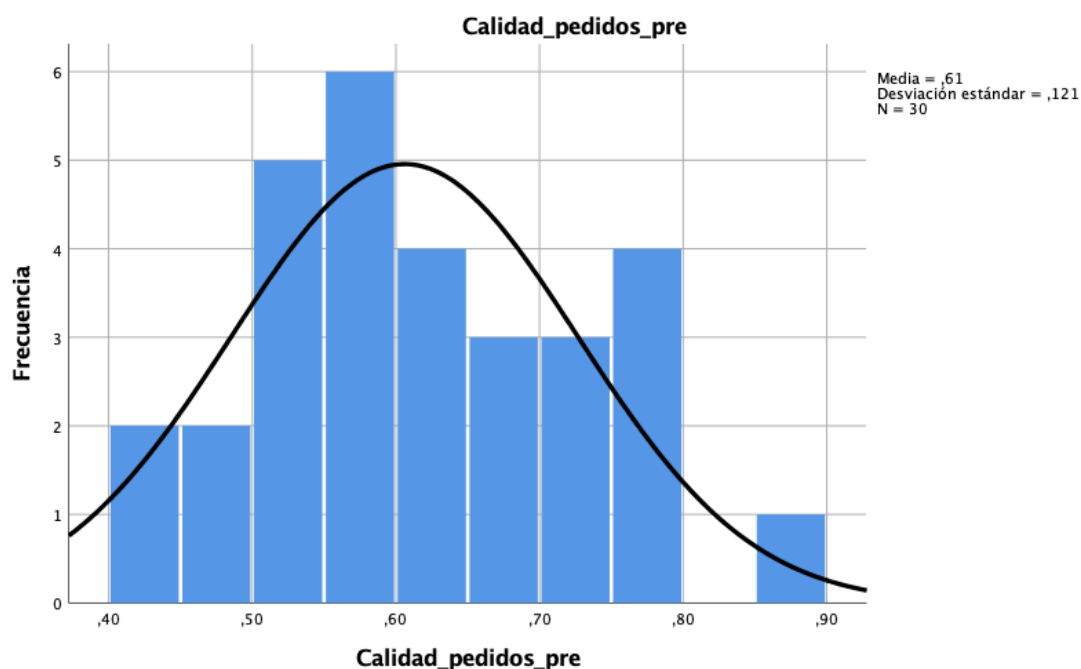
## Calidad de pedidos

**Tabla 12: Prueba de normalidad para el indicador calidad de pedidos**

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Calidad_pedidos_pre	,954	20	,213
Calidad_pedidos_post	,941	20	,099

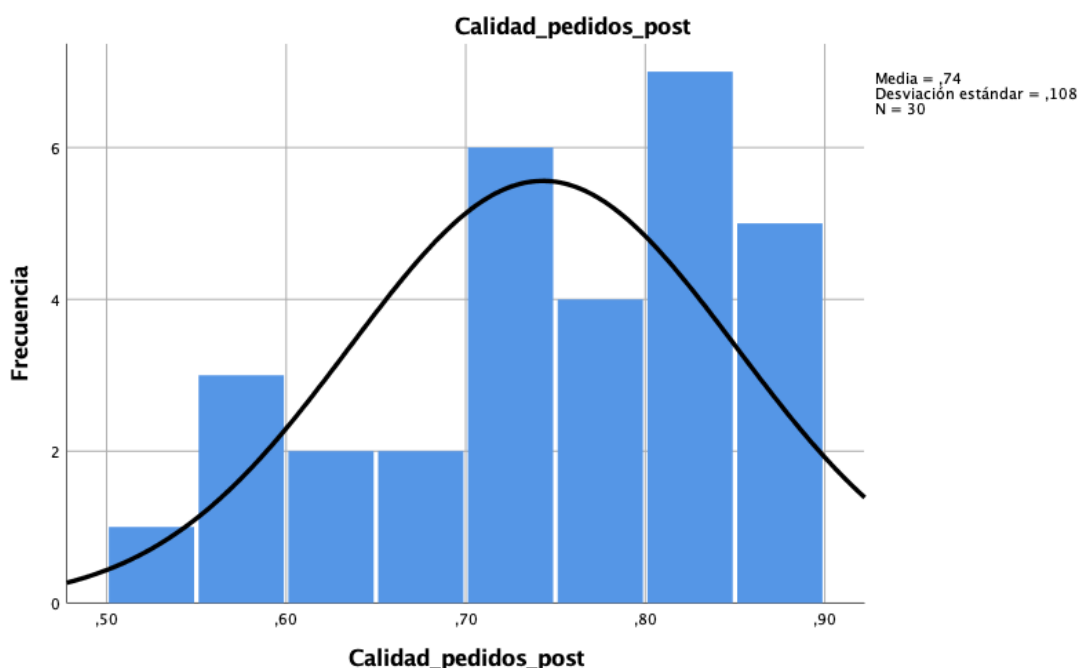
Como se visualiza en la tabla anterior los dos valores del nivel de significancia tanto para el pre test como el post test son mayores a 0.05, por lo tanto, para este indicador la distribución es de tipo normal o paramétrica.

**Figura 15: Histograma con curva normal para el indicador calidad de pedidos antes de la implementación del marketplace**



En la figura anterior se visualiza como se han distribuido los valores resultantes antes de la implementación del Marketplace, obteniendo un valor promedio de 61%.

**Figura 16: Histograma con curva normal para el indicador calidad de pedidos después de la implementación del Marketplace**



En la figura anterior se visualiza como se han distribuido los valores resultantes después de la implementación del Marketplace, obteniendo un valor promedio de 74%.

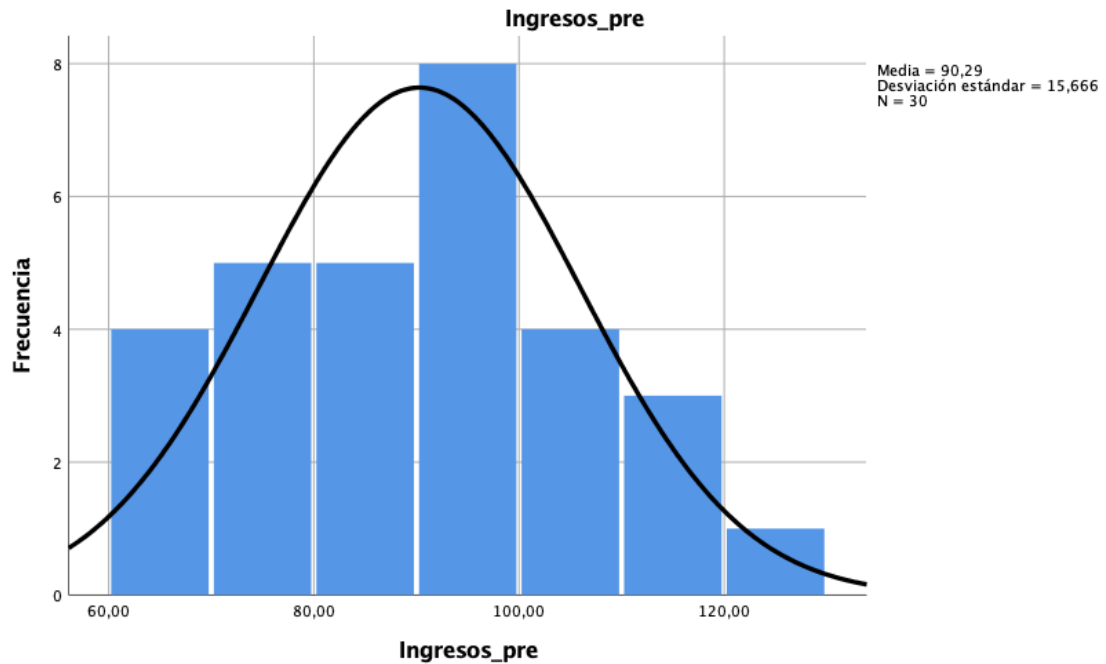
## Ingresos

**Tabla 13: Prueba de normalidad para el indicador ingresos**

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Ingresos_pre	,962	20	,345
Ingresos_post	,958	20	,281

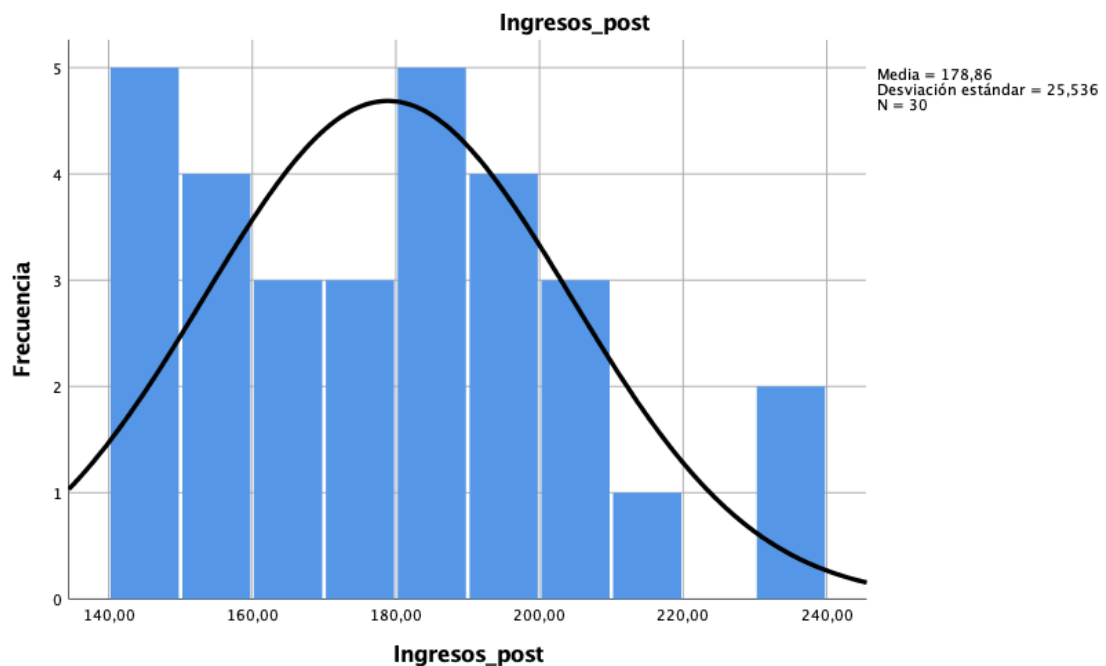
Como se visualiza en la tabla anterior los dos valores del nivel de significancia tanto para el pre test como el post test son mayores a 0.05, por lo tanto, para este indicador la distribución es de tipo normal o paramétrica.

**Figura 17: Histograma con curva normal para el indicador ingresos antes de la implementación del marketplace**



En la figura anterior se visualiza como se han distribuido los valores resultantes antes de la implementación del Marketplace, obteniendo un valor promedio de 90.29.

**Figura 18: Histograma con curva normal para el indicador ingresos después de la implementación del marketplace**





En la figura anterior se visualiza como se han distribuido los valores resultantes después de la implementación del Marketplace, obteniendo un valor promedio de 178.86.

### **Prueba de hipótesis**

Ahora se realiza la prueba de hipótesis, en donde se tiene que aplicar la prueba t-student ya que para ambos indicadores la distribución ha sido normal, para luego buscar el rechazo de la hipótesis nula y la aceptación de la alterna.

### **Calidad de pedidos**

**H1:** El Marketplace aumenta la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L.

### **Indicador: Calidad de pedidos generados**

#### **Dónde:**

CPG<sub>a</sub>: Calidad de pedidos generados antes del sistema web

CPG<sub>d</sub>: Calidad de pedidos generados después del sistema web

**Hipótesis Nula H1<sub>0</sub>:** El Marketplace no aumenta la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L.

$$\text{H1}_0: \text{CPG}_a - \text{CPG}_d \leq 0$$

$$\text{H1}_0: \text{CPG}_a > \text{CPG}_d$$

**Hipótesis Alterna H1<sub>a</sub>:** El Marketplace aumenta la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L.

$$\text{H1}_a: \text{CPG}_a - \text{CPG}_d > 0$$

$$\text{H1}_a: \text{CPG}_d > \text{CPG}_a$$

## Prueba de t- student

Para Lopez y Expósito (2016) detallan que este proceso de evaluación consiste en realizar la comparación de medias para un grupo de casos. Los que se espera es que los sujetos se encuentren asignados de forma aleatoria a los grupos de tal forma que cualquier diferencia que pueda existir en las respuestas se trate de manera adecuada.

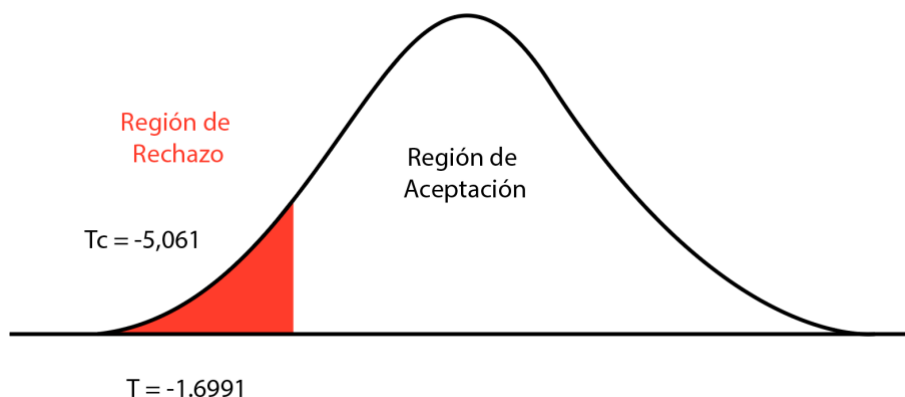
Para el indicador calidad de pedidos se obtuvo como valor de contraste -5.061, el cual se comparó con el valor de intersección de la tabla t-studet, cuyo valor fue de -1.6991, en donde se puede concluir que este valor es mayor al valor de contraste, el detalle de la prueba se puede visualizar en la siguiente tabla:

**Tabla 14: Aplicación de la prueba t-student para el indicador calidad de pedidos**

	Media	Desv. Desviación	t	gl	Sig. (bilateral)	
Par 1	Calidad_pedidos_pre - Calidad_pedidos_post	-,13700	,14826	-5,061	29	,000

En base a lo mencionado se concluye que se realiza el rechazo de la hipótesis nula y se realiza la aceptación de la alterna, a un 95% de confianza, concluyendo que un Marketplace aumenta la calidad de los pedidos en el proceso de ventas. El gráfico de la región de rechazo se visualiza a continuación:

**Figura 19: Curva de región de rechazo para el indicador calidad de pedidos**



## Ingresos

**H2:** El Marketplace incrementa los **ingresos** en el proceso de ventas de AC Brands & Consulting E.I.R.L.

**Indicador: Ingresos**

**Dónde:**

**SP<sub>a</sub>:** Ingresos antes del sistema web.

**SP<sub>d</sub>:** Ingresos después del sistema web.

**Hipótesis H2<sub>0</sub>:** El Marketplace no incrementa los ingresos en el proceso de ventas de AC Brands & Consulting E.I.R.L.

$$H1_0: I_d - I_a \leq 0$$

$$H1_0: I_d > I_a$$

**Hipótesis H2<sub>a</sub>:** El Marketplace no incrementa los ingresos en el proceso de ventas de AC Brands & Consulting E.I.R.L.

$$H1_a: I_d - I_a > 0$$

$$H1_a: I_A > I_d$$

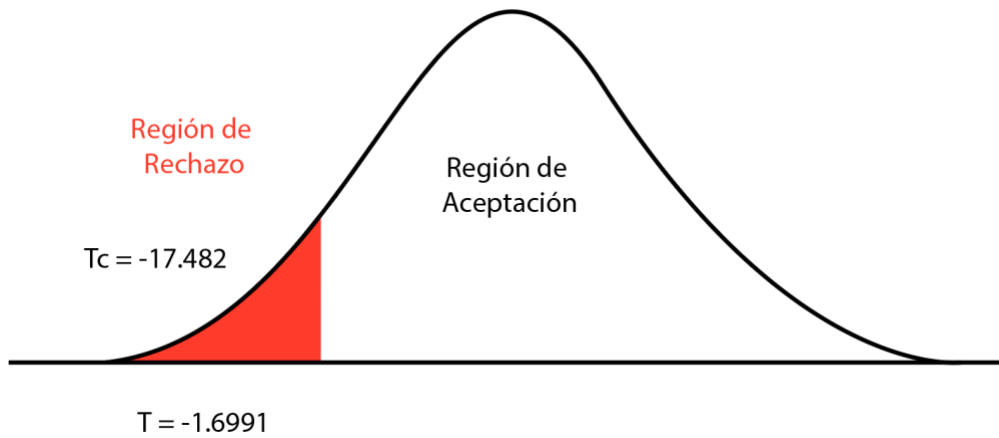
Para el indicador calidad de pedidos se obtuvo como valor de contraste -17.482, el cual se comparó con el valor de intersección de la tabla t-studet, cuyo valor fue de -1.6991, en donde se puede concluir que este valor es mayor al valor de contraste, el detalle de la prueba se puede visualizar en la siguiente tabla:

**Tabla 15: Aplicación de la prueba t-student para el indicador Ingresos**

		Media	Desv. Desviación	t	gl	Sig. (bilateral)
Par 1	Ingresos_pre - Ingresos_post	-88,57033	5,06645	-17,482	29	,000

En base a lo mencionado se concluye que se realiza el rechazo de la hipótesis nula y se realiza la aceptación de la alterna, a un 95% de confianza, concluyendo que un Marketplace aumenta los ingresos en el proceso de ventas. El gráfico de la región de rechazo se visualiza a continuación:

**Figura 20: Curva de región de rechazo para el indicador Ingresos**



## V. DISCUSIÓN

La presente investigación obtuvo como resultado, gracias a la solución planteada, el aumento en la calidad de pedidos generados, de un 60.6% a un 74.3%, lo que equivale a un 13% de aumento. Del mismo modo, Flores y Valdez en su investigación “Marketplace para el proceso de ventas de la Empresa Creaciones Peloncito S.A.C.”, 2019, concluyen que la implementación de dicha herramienta incrementó las ventas de 42.83% a 71.65% al tener menos devoluciones lo que equivale a un 28.8% de aumento.

Asimismo, de la presente investigación se obtuvo un aumento en los ingresos, de un 90.29 a un 178.86 lo que equivale a un 88.57 de aumento. De la misma manera Huayllasco, en su investigación “Aplicación de un modelo de E-MARKETPLACE vertical en el proceso de venta de artículos de vestir en la galería CENTRO GAMARRA”, 2019, concluye que luego de la implementación hubo un incremento en las ventas generando ingresos importantes.

A través del uso del Marketplace fueron evidenciadas las mejoras en la empresa AC Brands & Consulting E.I.R.L; dado que los administradores pueden registrar las ventas de en la misma plataforma, así como los comercios afiliados pueden visualizar sus ingresos y manejar sus inventarios de manera adecuada. De forma similar, Simões de Arruda Corrêa da Silva, Borges Senra, Porfírio Jacomino, Guarnieri, y Ruiz Golemba de Britto, en su artículo “E-MARKETPLACES: Canales que mejoran el rendimiento de ventas en línea para Ecommerces”, 2019, concluyen que los Marketplaces han demostrado ser una fuente de venta competitiva para los comercios electrónicos, ya que son plataformas que aportan ayuda a los vendedores y compradores para que estos puedan tener resultados eficientes a nivel de compra y venta de diferente tipos de artículos.

## **VI. CONCLUSIONES**

Se concluyó que el Marketplace mejoró el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L. de manera social, tecnológica, teórica y con un impacto positivo a nivel económico.

Por otro lado, se concluyó que existe una mejora, dado que se generó un aumento en la calidad de pedidos generados lo que permitió una buena reputación de la empresa ya que se tiene un mejor control de todos los ítems que se ofrecen desde la misma.

Finalmente, se concluyó que el Marketplace aumentó los ingresos en casi un 50% esto ya que se puede contar con una herramienta disponible y al alcance de todos los usuarios a nivel nacional, generando oportunidades de manera Online como Offline para la empresa.

## **VII. RECOMENDACIONES**

Para estudios futuros y de índole similar, se recomienda utilizar las métricas de medición de calidad de pedidos generados, así como de ingresos, dado que brindan enfoques importantes para el proceso de ventas de cualquier tipo de emprendimiento o negocio a nivel de ventas.

Adicionalmente, se recomienda implementar este tipo de soluciones en otro tipo de negocios, dado que su impacto puede ser relevante e importante para los diferentes rubros de comercio electrónico.

Se recomienda a la empresa AC Brands & Consulting E.I.R.L. continuar con las innovaciones tecnológicas para así generar nuevas oportunidades en otros canales de comunicación con los clientes.

Finalmente, se recomienda realizar el seguimiento de los picos de ventas y seguimiento de entregas, con el fin de mantener una buena reputación a nivel organizacional, evitando retrasos en las entregas y ofreciendo una atención de calidad.

## REFERENCIAS

ACOSTA, Marjorie, y otros. 2018. *La Administración de Ventas*. España: 3Ciencias, 2018. ISBN: 9788494825729.

ALAUDEEN ABDULRAHMAN, Rana, ALI HAMID, Sawsan y ALI KHAMEES, Ruaa. 2021. *Las mejores prácticas de calidad de Programación eXtrema (XP)*. Bagdad, Iraq: Tendencias recientes en tecnología de la información y su aplicación.

BAENA, Guillermo, 2017. *Metodología de la Investigación*. Mexico: Grupo Editorial Patria. ISBN: 9786077447481

BBC, 2020. *Coronavirus: qué están haciendo Amazon, eBay y las grandes tiendas online para evitar la especulación de precios por la pandemia del covid-19*. Disponible en: <https://bbc.in/3zNpCX9>

BERNAL, Enrique y POCCO, Juan Carlos. 2018. *Implementación de un sistema E-commerce para mejorar el proceso comercial de la empresa SERGETIC S.A.C.* Lima, Perú : Universidad Autónoma del Perú.

CARRASCO S. 2015. *Metodología de la investigación científica. Aplicaciones en educación y otras ciencias sociales*. Lima, Perú: Editorial San Marcos. ISBN: 9972342425.

CASTEJÓN, Martín, LECHUGA, Lafuente y MARTÍNEZ, Faura, 2015. *Guía práctica de estadística aplicada a la empresa y al marketing*. Madrid, España: Parainfo. ISBN: 9788428337489.

CASTELLANOS José, LOAIZA Martha y CUESTA Carlos. 2016. *Importancia de las TIC para la competitividad de las Pymes en Colombia*. Colombia. Disponible en: <https://bit.ly/3vSC6tq>.



CODE CADEMY, 2021. *Learn React*. Disponible en: <https://bit.ly/3d3gEer>

CORREA DA SILVA, Simoes, et al., 2019. *E-MARKETPLACES: Crecimiento de las ventas para los E-commerces*. Brasil: Caderno de Administração, págs. 21-40.

CORREA, Juan y BARRERA, Carlos, 2016. *Introducción a la Estadística Bayesiana*. Colombia. ISBN: 9789585412723.

EL COMERCIO, 2020. *¿Cómo las empresas pueden aprovechar mejor la explosión del E-commerce?* Disponible en: <https://bit.ly/2TRnYTV>

Luna F, 2020. *JavaScript – Aprende a programar en el lenguaje de la web*. Disponible en: <https://bit.ly/3xEKNZp>

FLORES, Karina y VALDEZ, Christian. 2019. *Marketplace para el proceso de ventas de la Empresa Creaciones Pelocito S.A.C*. Lima, Perú : Universidad César Vallejo.

FORD, Neal y RICHARDS Mark, 2020. *Fundamentals of Software Architecture: An Engineering Approach*. Estados Unidos: O'Reilly Media. ISBN: 9781498043454.

FRUHWIRTH, Michael, RACHINGER, Michael y PRLJA, Emina. 2020. *Descubriendo los modelos comerciales de los Marketplaces*. Honolulu, Hawai : HICSS, 2020. Disponible en: <http://hdl.handle.net/10125/64446>.

GALINDO, Hector. 2020. *Estadística para no estadísticos*. s.l. : 3Ciencias. ISBN: 9788412145939.

GARCÍA, María y SÁNCHEZ, Blanca. 2013. *El uso de las tecnologías de información y comunicación TIC en las micro, pequeñas y medianas empresas (MIPyME) industriales mexicanas*. Enl@ce: Revista Venezolana de Información, tecnología y conocimiento. Disponible en: <https://bit.ly/3gle53Q>

HASHMI, Saleen y AHMED, Imtiaz. 2020. *Enciclopedia de materiales renovables y sostenibles*. Kuala, Malasia: ELSEVIER.

HERNÁNDEZ, Roberto y MENDOZA, Christian, 2018. *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta*. Mexico DF: Mc Graw Hill. ISBN: 9781456260965.

HERRERA, Jenny. 2019. *Plan de Marketing Paso a paso: Creación de una tienda online del equipo Delfín S.C.* España: 3Ciencias. ISBN: 9788412075656.

HUAYLLASCO, Alex. 2019. *Aplicación de un modelo de E-marketplace vertical en el proceso de venta de artículos de vestir en la galería "Centro Gamarra"*. Lima, Perú: Universidad Inca Garcilaso de la Vega.

INCA, Sandra. 2017. *Sistema web para el proceso de ventas en la empresa Zoe*. Lima, Perú: Universidad César Vallejo.

KAWA, Arkadiusz y WALESIAK, Magdalena. 2019. *Marketplace como actor clave en redes de valor de comercio electrónico*. Poznań, Polonia: LogForum.

KOTLER, Phillip y ARMSTRONG, Gary. 2016. *Fundamentos de Marketing 6ta Edición*. s.l. Mexico: Pearson Prentice Hall. ISBN: 9789702604006. Disponible en: <https://bit.ly/3oiZNZq>.

LARAVEL, 2021. *Introduction*. Disponible en: <https://laravel.com/docs/8.x>

LA VANGUARDIA, 2020. *El comercio electrónico vive una maduración exprés con la pandemia*. Disponible en: <https://bit.ly/3iVxHD4>

LI, Boying, y otros. 2016. *Predecir el rendimiento de las ventas en el Marketplace en línea: Un enfoque de big data*. China: Computación e Ingeniería Industrial.

LOPEZ, Fernandez F. y Expósito Gazquez A. Administración pública y nuevas tecnologías. 2016. Disponible en: <https://books.google.com.pe/books?id=cwotDgAAQBAJ&pg=PA146&dq=prueba+t+student&hl=en&sa=X&ved=2ahUKEwigpLWC-NTwAhVdEVkFHURaBokQ6AEwAnoECAEQAg#v=onepage&q=prueba%20t%20student&f=false>

MOHAMMAD, Amir y ALLAH, Ata. 2020. *Marketplace, revendedor o canal de tienda web: El impacto de la política de devoluciones y el desbordamiento entre canales de Marketplace*. Tehran, Iran : Universidad de Tehran. Disponible es: <https://bit.ly/33G4p2y>.

MORA, Luis, 2012. *Gestión Logística Integral*. Bogotá, Colombia: Starbook Editorial. ISBN: 9788492650880.

MYSQL, 2021. *What is MySQL?* Disponible en: <https://bit.ly/35FiNsz>

NADER, Rad y TURLEY, Frank. 2019. *Los Fundamentos de Agile Scrum*. Países Bajos: Van Haren Publishing. ISBN: 9789401805407.

ÑAUPAS, Humberto y otros. 2018. *Metodología de la investigación cuantitativa - cualitativa y redacción de tesis*. Bogotá, Colombia: Ediciones de la U. ISBN: 9789587628760.

PALOMINO, Gaby. 2018. *Estudio sobre un sistema de información en la nube, aplicado a la gestión de ventas de la empresa La Calera de la Ciudad de Chincha*. Ica, Perú: Universidad Nacional San Luis Gonzaga de Ica.

PHP, 2021. *PHP Hypertext Preprocesso*. Disponible en: <https://www.php.net>

RODRÍGUEZ FRANCO J., PIERDANT RODRÍGUEZ A. y RODRÍGUEZ JIMÉNEZ E. 2016. *Estadística para Administración 2da Edición*. Ciudad de México, México: Patria. ISBN: 9786077444909.

RUIZ, Elena. 2017. *Nuevas tendencias en los sistemas de información*. Madrid, España: Centro de estudios Ramon Areces SA. ISBN: 9788499612690.

RUIZ, María, et at. 2020. *Matemáticas aplicadas a las Ciencias Sociales II*. Madrid, España: Editorial Editex. ISBN: 9788413212128.

SALAZAR, Jorge. 2018. *Desarrollo e implementación de un web site corporativo y de un marketplace online para la empresa Planeamiento Digital S.A.C*. Lima, Perú: Universidad Tecnológica del Perú.

SÁNCHEZ RUIZ M. y SOLÍS FRAILE R. 2016. *PMAR – Ámbito Científico y Matemático I: Programa de mejora de aprendizaje y del rendimiento*. Madrid, España: Editex. ISBN: 9788490787724.

SILVESTRE, Irenzon y HUAMÁN, Cecilia, 2019. *Pasos para elaborar la investigación y redacción de la tesis universitaria*. Lima, Perú: San Marcos. ISBN: 9786123155827

VALDERRAMA, Santiago, 2019. *Pasos para Elaborar Proyectos de investigación Científica*. Lima, Perú: Editorial San Marcos. ISBN: 9786123028787

WERTZ, Boris y TRAN, Angela. 2015. *Una guía para el Marketplace*. Vancouver, Canadá: Version One.

## **ANEXOS**

### Anexo 1: Matriz de consistencia

Problema	Objetivo	Hipótesis	Variables	Dimensiones	Indicador	Metodología
<b>General</b>	<b>General</b>	<b>General</b>	<b>Independiente</b>			<b>Tipo de investigación</b>
¿De qué manera el Marketplace mejorará el proceso de ventas de AC Brands & Consulting E.I.R.L.?	Determinar de qué manera el Marketplace mejorará el proceso de ventas de AC Brands & Consulting E.I.R.L.	El Marketplace mejora el proceso de ventas de AC Brands & Consulting E.I.R.L.	Marketplace			Aplicada
						<b>Diseño de investigación</b>
						Pre-experimental
<b>Específicos</b>	<b>Específicos</b>	<b>Específicos</b>	<b>Dependiente</b>			<b>Población</b>
¿Cómo el Marketplace aumentará la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L.?	Determinar cómo el Marketplace aumentará la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L.	El Marketplace aumenta la calidad de pedidos generados en el proceso de ventas de AC Brands & Consulting E.I.R.L.			Calidad de pedidos generados	1500 pedidos
						<b>Muestra</b>
						306 pedidos
						<b>Muestreo</b>
						Muestreo aleatorio simple
						<b>Metodología de trabajo</b>
						SCRUM
						<b>Técnica de investigación</b>
						Fichaje
						<b>Instrumento de recolección</b>
						Ficha de registros
¿Cómo el Marketplace incrementará los ingresos en el proceso de ventas de AC Brands & Consulting E.I.R.L.?	Determinar cómo el Marketplace incrementará los ingresos en el proceso de ventas de AC Brands & Consulting E.I.R.L.	El Marketplace incrementa los ingresos en el proceso de ventas de AC Brands & Consulting E.I.R.L.	Proceso de ventas	Cierre	Ingresos	

## Anexo 2: Carta de autorización AC BRAND & CONSULTING E.I.R.L.

### CARTA DE AUTORIZACIÓN

10 de mayo del 2021

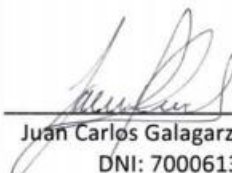
Señores:  
Universidad César Vallejo

De mi mayor consideración:

Yo, Juan Carlos Galagarza Chunga, Gerente Comercial de la empresa AC BRANDS & CONSULTING E.I.R.L. con RUC 20606801417, autorizo a Marzio Josué Perez Arbulú estudiante de la Universidad César Vallejo, de la E.A.P de Ing. de Sistemas, para que realice el proyecto de investigación del Marketplace en mi empresa.

Sin más que decir, me despido y dejo constancia del compromiso de mi persona con el estudiante.

Atentamente,



---


Juan Carlos Galagarza Chunga

DNI: 70006137

Gerente Comercial de AC BRANDS & CONSULTING E.I.R.L.

### Anexo 3: Fichas de registro PreTest

Ficha de Registro					
Investigador	Perez Arbulu, Marzio Josué	Tipo de Prueba			PreTest
Empresa investigada	AC BRANDS & CONSULTING E.I.R.L.				
Motivo de Investigación	PROCESO DE VENTAS				
Fecha Inicio	01 Marzo	Fecha fin		31 Marzo	
Indicador	Ingresos	Formula		I = VBM /TC	
Num	Fecha	Cant. Pedidos	Valor Bruto de mercadería	Tasa de comisión	Ingresos
1	1-Mar-21	9	540	0.2	108.00
2	2-Mar-21	10	400	0.2	80.00
3	3-Mar-21	11	600	0.2	120.00
4	4-Mar-21	9	566.4	0.2	113.28
5	5-Mar-21	12	544.5	0.2	108.90
6	6-Mar-21	9	453	0.2	90.60
7	7-Mar-21	9	340	0.2	68.00
8	8-Mar-21	9	470	0.2	94.00
9	9-Mar-21	10	454.3	0.2	90.86
10	10-Mar-21	12	520	0.2	104.00
11	11-Mar-21	10	430.12	0.2	86.02
12	12-Mar-21	9	356.3	0.2	71.26
13	13-Mar-21	11	587.4	0.2	117.48
14	14-Mar-21	9	476.3	0.2	95.26
15	15-Mar-21	10	387.4	0.2	77.48
16	16-Mar-21	10	493.4	0.2	98.68
17	17-Mar-21	11	464.3	0.2	92.86
18	18-Mar-21	12	324.3	0.2	64.86
19	19-Mar-21	11	398.6	0.2	79.72
20	20-Mar-21	12	456.4	0.2	91.28
21	21-Mar-21	10	453.6	0.2	90.72
22	22-Mar-21	10	356.4	0.2	71.28
23	23-Mar-21	11	512.43	0.2	102.49
24	24-Mar-21	9	345.3	0.2	69.06
25	25-Mar-21	10	564.3	0.2	112.86
26	26-Mar-21	11	445.4	0.2	89.08
27	27-Mar-21	10	432.3	0.2	86.46
28	28-Mar-21	9	345.4	0.2	69.08
29	29-Mar-21	10	390.4	0.2	78.08
30	30-Mar-21	11	435.3	0.2	87.06






Ficha de Registro				
Investigador	Perez Arbulu, Marzio Josué	Tipo de Prueba		PreTest
Empresa investigada	AC BRANDS & CONSULTING E.I.R.L.			
Motivo de Investigación	PROCESO DE VENTAS			
Fecha Inicio	01 Marzo	Fecha fin	30 Marzo	
Indicador	Calidad de pedidos generados	Formula	CPG = NPET/NTPS	
Num.	Fecha	Número de pedidos entregados a tiempo	Número total de pedidos solicitados	Calidad de pedidos generados
1	1-Mar-21	4	9	0.44
2	2-Mar-21	6	10	0.60
3	3-Mar-21	5	11	0.45
4	4-Mar-21	6	9	0.67
5	5-Mar-21	7	12	0.58
6	6-Mar-21	8	9	0.89
7	7-Mar-21	7	9	0.78
8	8-Mar-21	6	9	0.67
9	9-Mar-21	5	10	0.50
10	10-Mar-21	6	12	0.50
11	11-Mar-21	7	10	0.70
12	12-Mar-21	7	9	0.78
13	13-Mar-21	6	11	0.55
14	14-Mar-21	7	9	0.78
15	15-Mar-21	7	10	0.70
16	16-Mar-21	7	10	0.70
17	17-Mar-21	7	11	0.64
18	18-Mar-21	7	12	0.58
19	19-Mar-21	6	11	0.55
20	20-Mar-21	5	12	0.42
21	21-Mar-21	5	10	0.50
22	22-Mar-21	6	10	0.60
23	23-Mar-21	6	11	0.55
24	24-Mar-21	7	9	0.78
25	25-Mar-21	6	10	0.60
26	26-Mar-21	5	11	0.45
27	27-Mar-21	5	10	0.50
28	28-Mar-21	6	9	0.67
29	29-Mar-21	5	10	0.50
30	30-Mar-21	6	11	0.55

#### Anexo 4: Fichas de registro Re-test

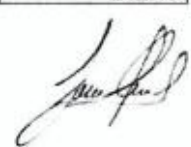
Ficha de Registro					
Investigador	Perez Arbulu, Marzio Josué		Tipo de Prueba	Re test	
Empresa investigada	AC BRANDS & CONSULTING E.I.R.L.				
Motivo de Investigación	PROCESO DE VENTAS				
Fecha Inicio	01 Abril	Fecha fin	30 Abril		
Indicador	Ingresos	Formula		I = VBM /TC	
Num	Fecha	Cant. Pedidos	Valor Bruto de mercadería	Tasa de comisión	Ingresos
1	1-Abr-21	9	530	0.2	106.00
2	2-Abr-21	10	389	0.2	77.80
3	3-Abr-21	11	540	0.2	108.00
4	4-Abr-21	9	575.5	0.2	115.10
5	5-Abr-21	12	543.3	0.2	108.66
6	6-Abr-21	9	440	0.2	88.00
7	7-Abr-21	9	334	0.2	66.80
8	8-Abr-21	9	480	0.2	96.00
9	9-Abr-21	10	454.3	0.2	90.86
10	10-Abr-21	12	530	0.2	106.00
11	11-Abr-21	10	431	0.2	86.20
12	12-Abr-21	9	357	0.2	71.40
13	13-Abr-21	11	580.3	0.2	116.06
14	14-Abr-21	9	465.3	0.2	93.06
15	15-Abr-21	10	376.2	0.2	75.24
16	16-Abr-21	10	493.4	0.2	98.68
17	17-Abr-21	11	430.4	0.2	86.08
18	18-Abr-21	12	340	0.2	68.00
19	19-Abr-21	11	398.6	0.2	79.72
20	20-Abr-21	12	456.4	0.2	91.28
21	21-Abr-21	10	455	0.2	91.00
22	22-Abr-21	10	356.4	0.2	71.28
23	23-Abr-21	11	512.43	0.2	102.49
24	24-Abr-21	9	344.3	0.2	68.86
25	25-Abr-21	10	564.3	0.2	112.86
26	26-Abr-21	11	445.4	0.2	89.08
27	27-Abr-21	10	445	0.2	89.00
28	28-Abr-21	9	354	0.2	70.80
29	29-Abr-21	10	360	0.2	72.00
30	30-Abr-21	11	425.4	0.2	85.08



Ficha de Registro				
Investigador	Perez Arbulu, Marzio Josué	Tipo de Prueba		Re test
Empresa investigada	AC BRANDS & CONSULTING E.I.R.L.			
Motivo de Investigación	PROCESO DE VENTAS			
Fecha Inicio	01 Abril	Fecha fin	30 Abril	
Indicador	Calidad de pedidos generados	Formula	CPG = NPET/NTPS	
Num.	Fecha	Número de pedidos entregados a tiempo	Número total de pedidos solicitados	Calidad de pedidos generados
1	1-Abr-21	4	10	0.40
2	2-Abr-21	6	9	0.67
3	3-Abr-21	5	11	0.45
4	4-Abr-21	6	10	0.60
5	5-Abr-21	7	11	0.64
6	6-Abr-21	8	9	0.89
7	7-Abr-21	7	10	0.70
8	8-Abr-21	6	9	0.67
9	9-Abr-21	5	11	0.45
10	10-Abr-21	6	10	0.60
11	11-Abr-21	7	10	0.70
12	12-Abr-21	7	9	0.78
13	13-Abr-21	6	10	0.60
14	14-Abr-21	7	9	0.78
15	15-Abr-21	7	10	0.70
16	16-Abr-21	7	9	0.78
17	17-Abr-21	7	10	0.70
18	18-Abr-21	7	11	0.64
19	19-Abr-21	6	11	0.55
20	20-Abr-21	5	12	0.42
21	21-Abr-21	5	11	0.45
22	22-Abr-21	6	10	0.60
23	23-Abr-21	6	12	0.50
24	24-Abr-21	7	9	0.78
25	25-Abr-21	6	11	0.55
26	26-Abr-21	5	11	0.45
27	27-Abr-21	5	11	0.45
28	28-Abr-21	6	9	0.67
29	29-Abr-21	5	11	0.45
30	30-Abr-21	6	10	0.60

## Anexo 5: Ficha de registro Post Test

Ficha de Registro					
Investigador	Perez Arbulu, Marzio Josué	Tipo de Prueba		PostTest	
Empresa investigada	AC BRANDS & CONSULTING E.I.R.L.				
Motivo de Investigación	PROCESO DE VENTAS				
Fecha Inicio	01 Octubre	Fecha fin		30 Octubre	
Indicador	Ingresos	Formula		I = VBM /TC	
Num	Fecha	Cant. Pedidos	Valor Bruto de mercadería	Tasa de comisión	Ingresos
1	1-Oct-21	11	874.4	0.2	174.88
2	2- Oct -21	9	819.4	0.2	163.88
3	3- Oct -21	10	1019	0.2	203.80
4	4- Oct -21	9	987.4	0.2	197.48
5	5- Oct -21	11	1155.4	0.2	231.08
6	6- Oct -21	9	913.4	0.2	182.68
7	7- Oct -21	12	1022.4	0.2	204.48
8	8- Oct -21	9	725.9	0.2	145.18
9	9- Oct -21	11	903.5	0.2	180.70
10	10- Oct -21	12	993.4	0.2	198.68
11	11- Oct -21	9	734.4	0.2	146.88
12	12- Oct -21	9	786.4	0.2	157.28
13	13- Oct -21	10	880.4	0.2	176.08
14	14- Oct -21	11	925.4	0.2	185.08
15	15- Oct -21	10	791.4	0.2	158.28
16	16- Oct -21	9	734.4	0.2	146.88
17	17- Oct -21	11	850.4	0.2	170.08
18	18- Oct -21	10	1004.4	0.2	200.88
19	19- Oct -21	9	833.4	0.2	166.68
20	20- Oct -21	12	985.4	0.2	197.08
21	21- Oct -21	10	767.4	0.2	153.48
22	22- Oct -21	9	794.4	0.2	158.88
23	23- Oct -21	10	831.4	0.2	166.28
24	24- Oct -21	9	708.4	0.2	141.68
25	25- Oct -21	11	906.4	0.2	181.28
26	26- Oct -21	10	910.4	0.2	182.08
27	27- Oct -21	10	957.4	0.2	191.48
28	28- Oct -21	11	930.9	0.2	186.18
29	29- Oct -21	11	1197.4	0.2	239.48
30	30- Oct -21	12	1062.4	0.2	212.48



Ficha de Registro				
Investigador	Perez Arbulu, Marzio Josué	Tipo de Prueba		PostTest
Empresa investigada	AC BRANDS & CONSULTING E.I.R.L.			
Motivo de Investigación	PROCESO DE VENTAS			
Fecha Inicio	01 Octubre	Fecha fin	30 Octubre	
Indicador	Calidad de pedidos generados	Formula	CPG = NPET/NTPS	
Num.	Fecha	Número de pedidos entregados a tiempo	Número total de pedidos solicitados	Calidad de pedidos generados
1	1-Oct-21	7	11	0.64
2	2- Oct -21	8	9	0.89
3	3- Oct -21	7	10	0.70
4	4- Oct -21	7	9	0.78
5	5- Oct -21	6	11	0.55
6	6- Oct -21	7	9	0.78
7	7- Oct -21	6	12	0.50
8	8- Oct -21	8	9	0.89
9	9- Oct -21	8	11	0.73
10	10- Oct -21	7	12	0.58
11	11- Oct -21	6	9	0.67
12	12- Oct -21	8	9	0.89
13	13- Oct -21	6	10	0.60
14	14- Oct -21	8	11	0.73
15	15- Oct -21	8	10	0.80
16	16- Oct -21	7	9	0.78
17	17- Oct -21	8	11	0.73
18	18- Oct -21	8	10	0.80
19	19- Oct -21	8	9	0.89
20	20- Oct -21	9	12	0.75
21	21- Oct -21	8	10	0.80
22	22- Oct -21	6	9	0.67
23	23- Oct -21	8	10	0.80
24	24- Oct -21	8	9	0.89
25	25- Oct -21	9	11	0.82
26	26- Oct -21	8	10	0.80
27	27- Oct -21	7	10	0.70
28	28- Oct -21	8	11	0.73
29	29- Oct -21	9	11	0.82
30	30- Oct -21	7	12	0.58

## Anexo 6: Evaluación de expertos para el uso de la metodología de desarrollo



### EVALUACIÓN DE METODOLOGÍA DE DESARROLLO DE DESARROLLO TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombres del experto: Mgtr. Iván Martín Perez Farfán

Título y/o Grado: Magíster en Ingeniería de Sistema

Fecha: 05/06/2021

#### TÍTULO DE TESIS

MARKETPLACE PARA EL PROCESOS DE VENTAS EN LA EMPRESA  
AC BRANDS & CONSULTING S.A.C.

#### EVALUACIÓN DE METODOLOGÍA DE DESARROLLO

En la presente tabla de evaluación de experto, usted tiene la facultad de realizar la calificación de las metodologías de desarrollo involucradas para la presente investigación, mediante una serie de criterios con puntuaciones específicas los cuales se indican al final de la tabla. En ese sentido, se le exhorta la correcta determinación de la metodología de desarrollo del Marketplace para el proceso de ventas en la empresa AC BRANDS & CONSULTING S.A.C. Si hubiese algunas sugerencias por favor colocarlas líneas abajo.

Ítem	Criterios	Metodologías		
		SCRUM	XP	RUP
1	Las fases de desarrollo pueden ser de corto tiempo (De 7 a 14 días) según el tipo del proyecto	3	3	2
2	Los resultados se pueden visualizar conforme van transcurriendo las fases del proyecto	3	2	2
3	El proyecto puede llevarse a cabo con un equipo pequeño de personas	2	3	2
4	La inversión económica puede ser mínima de acuerdo con las fases que sean necesarias para el proyecto	3	3	2
5	El cliente puede percibir el valor del producto en cada fase del proyecto	3	3	2
6	Los resultados son visibles en un corto tiempo	3	3	2
7	Se ajusta a los diferentes cambios que puedan suceder bajo alguna coyuntura en específico	3	3	2
8	Los hitos se pueden llevar a cabo en el tiempo establecido	3	2	2
<b>Total</b>		<b>23</b>	<b>22</b>	<b>16</b>

La escala para evaluar es de 1: Malo, 2: Regular y 3: Bueno

Sugerencias:

-----  
-----

-----  
Firma Experto

**EVALUACIÓN DE METODOLOGÍA DE DESARROLLO DE DESARROLLO TABLA  
DE EVALUACIÓN DE EXPERTOS**

**Apellidos y Nombres del experto:** Ing. Hilario Aradiel Castañeda  
**Título y/o Grado:** \_\_\_\_\_  
**Fecha:** 13/06/2021

**TÍTULO DE TESIS**  
**MARKETPLACE PARA EL PROCESOS DE VENTAS EN LA EMPRESA**  
**AC BRANDS & CONSULTING S.A.C.**

**EVALUACIÓN DE METODOLOGÍA DE DESARROLLO**

En la presente tabla de evaluación de experto, usted tiene la facultad de realizar la calificación de las metodologías de desarrollo involucradas para la presente investigación, mediante una serie de criterios con puntuaciones específicas los cuales se indican al final de la tabla. En ese sentido, se le exhorta la correcta determinación de la metodología de desarrollo del Marketplace para el proceso de ventas en la empresa AC BRANDS & CONSULTING S.A.C. Si hubiese algunas sugerencias por favor colocarlas líneas abajo.

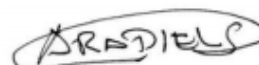
Ítem	Criterios	Metodologías		
		SCRUM	XP	RUP
1	Las fases de desarrollo pueden ser de corto tiempo (De 7 a 14 días) según el tipo del proyecto	3	2	1
2	Los resultados se pueden visualizar conformen van transcurriendo las fases del proyecto	3	2	1
3	El proyecto puede llevarse a cabo con un equipo pequeño de personas	3	2	1
4	La inversión económica puede ser mínima de acuerdo con las fases que sean necesarias para el proyecto	3	2	1
5	El cliente puede percibir el valor del producto en cada fase del proyecto	3	2	1
6	Los resultados son visibles en un corto tiempo	3	2	1
7	Se ajusta a los diferentes cambios que puedan suceder bajo alguna coyuntura en específico	3	2	1
8	Los hitos se pueden llevar a cabo en el tiempo establecido	3	2	1
<b>Total</b>		<b>24</b>	<b>16</b>	<b>8</b>

La escala para evaluar es de **1: Malo, 2: Regular y 3: Bueno**

**Sugerencias:**

-----  
 -----  
 -----

Firma Experto



**EVALUACIÓN DE METODOLOGÍA DE DESARROLLO DE DESARROLLO TABLA DE EVALUACIÓN DE EXPERTOS**

**Apellidos y Nombres del experto:** Ing. Guillermo Miguel Johnson Romero

**Título y/o Grado:** MSc

**Fecha:** 18/06/2021

**TÍTULO DE TESIS**

**MARKETPLACE PARA EL PROCESOS DE VENTAS EN LA EMPRESA AC BRANDS & CONSULTING S.A.C.**

**EVALUACIÓN DE METODOLOGÍA DE DESARROLLO**

En la presente tabla de evaluación de experto, usted tiene la facultad de realizar la calificación de las metodologías de desarrollo involucradas para la presente investigación, mediante una serie de criterios con puntuaciones específicas los cuales se indican al final de la tabla. En ese sentido, se le exhorta la correcta determinación de la metodología de desarrollo del Marketplace para el proceso de ventas en la empresa AC BRANDS & CONSULTING S.A.C. Si hubiese algunas sugerencias por favor colocarlas líneas abajo.

Ítem	Criterios	Metodologías		
		SCRUM	XP	RUP
1	Las fases de desarrollo pueden ser de corto tiempo (De 7 a 14 días) según el tipo del proyecto	3	3	1
2	Los resultados se pueden visualizar conformen van transcurriendo las fases del proyecto	3	2	3
3	El proyecto puede llevarse a cabo con un equipo pequeño de personas	3	1	1
4	La inversión económica puede ser mínima de acuerdo con las fases que sean necesarias para el proyecto	2	2	2
5	El cliente puede percibir el valor del producto en cada fase del proyecto	3	1	2
6	Los resultados son visibles en un corto tiempo	3	2	1
7	Se ajusta a los diferentes cambios que puedan suceder bajo alguna coyuntura en específico	3	3	2
8	Los hitos se pueden llevar a cabo en el tiempo establecido	3	3	2
<b>Total</b>		<b>23</b>	<b>17</b>	<b>14</b>

La escala para evaluar es de 1: **Malo**, 2: **Regular** y 3: **Bueno**

**Sugerencias:**

-----  
 -----  
 -----



**Firma Experto**



## Anexo 7: Evaluación de expertos para el uso del instrumento de medición



### TABLA DE EVALUACIÓN DE EXPERTOS

**Apellidos y Nombres del experto:** Mgtr. Iván Martín Perez Farfán

**Título y/o Grado:** Magíster en Ingeniería de Sistema

**Fecha:** 05/06/2021

### TÍTULO DE TESIS

**MARKETPLACE PARA EL PROCESOS DE VENTAS EN LA EMPRESA  
AC BRANDS & CONSULTING S.A.C.**

**Tabla de Evaluación de Expertos para el Indicador: Ingresos**

Indicadores	CRITERIOS	Deficiente 0% - 19%	Regular 20% - 39%	Bueno 40% - 60%	Muy Bueno 61% - 80%	Excelente 91% - 100%
Claridad	Está formado con el lenguaje apropiado.				70%	
Objetividad	Está expresado en conducta expresable				70%	
Organización	Está adecuado al avance de la ciencia y la tecnología				70%	
Suficiencia	Comprende los aspectos de cantidad y calidad				70%	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico				70%	
Consistencia	Está basado en aspectos técnicos, científicos acordes a la tecnología adecuada				70%	
Coherencia	Entre los índices indicadores y dimensiones				70%	
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr				70%	
Pertinencia	El instrumento es adecuado al tipo de investigación				70%	
Total Promedio					70%	

**PROMEDIO DE VALORACIÓN:** 70%

**OPCIÓN DE APLICABILIDAD:**

- ( X ) El instrumento puede ser aplicado, tal como está elaborado.  
( ) El instrumento debe ser mejorado, antes de ser aplicado.

Firma Experto

**TABLA DE EVALUACIÓN DE EXPERTOS**

**Apellidos y Nombres del experto:** Mgtr. Iván Martín Perez Farfán

**Título y/o Grado:** Magíster en Ingeniería de Sistema

**Fecha:** 05/06/2021

**TÍTULO DE TESIS**

**MARKETPLACE PARA EL PROCESOS DE VENTAS EN LA EMPRESA  
AC BRANDS & CONSULTING S.A.C.**

**Tabla de Evaluación de Expertos para el Indicador: Calidad de registros generados**

Indicadores	CRITERIOS	Deficiente 0% - 19%	Regular 20% - 39%	Bueno 40% - 60%	Muy Bueno 61% - 80%	Excelente 91% - 100%
Claridad	Está formado con el lenguaje apropiado.				70%	
Objetividad	Está expresado en conducta expresable				70%	
Organización	Está adecuado al avance de la ciencia y la tecnología				70%	
Suficiencia	Comprende los aspectos de cantidad y calidad				70%	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico				70%	
Consistencia	Está basado en aspectos técnicos, científicos acordes a la tecnología adecuada				70%	
Coherencia	Entre los índices indicadores y dimensiones				70%	
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr				70%	
Pertinencia	El instrumento es adecuado al tipo de investigación				70%	
Total Promedio					70%	

**PROMEDIO DE VALORACIÓN:** 70%

**OPCIÓN DE APLICABILIDAD:**

- ( x ) El instrumento puede ser aplicado, tal como está elaborado.  
( ) El instrumento debe ser mejorado, antes de ser aplicado.



Firma Experto

**TABLA DE EVALUACIÓN DE EXPERTOS**

Apellidos y Nombres del experto: Ing. Hilario Aradiel Castañeda  
 Título y/o Grado: \_\_\_\_\_  
 Fecha: 13/06/2021

**TÍTULO DE TESIS**  
**MARKETPLACE PARA EL PROCESOS DE VENTAS EN LA EMPRESA**  
**AC BRANDS & CONSULTING S.A.C.**

**Tabla de Evaluación de Expertos para el Indicador: Ingresos**

Indicadores	CRITERIOS	Deficiente 0% - 19%	Regular 20% - 39%	Bueno 40% - 60%	Muy Bueno 61% - 80%	Excelente 91% - 100%
Claridad	Está formado con el lenguaje apropiado.				80	
Objetividad	Esta expresado en conducta expresable				80	
Organización	Esta adecuado al avance de la ciencia y la tecnología				80	
Suficiencia	Comprende los aspectos de cantidad y calidad				80	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico				80	
Consistencia	Está basado en aspectos técnicos, científicos acordes a la tecnología adecuada				80	
Coherencia	Entre los índices indicadores y dimensiones				80	
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr				80	
Pertinencia	El instrumento es adecuado al tipo de investigación				80	
Total Promedio					80	

**PROMEDIO DE VALORACIÓN: 80**
**OPCIÓN DE APLICABILIDAD:**

- ( x ) El instrumento puede ser aplicado, tal como está elaborado.  
 ( ) El instrumento debe ser mejorado, antes de ser aplicado.

  
**Firma Experto**

**TABLA DE EVALUACIÓN DE EXPERTOS**
**Apellidos y Nombres del experto:** Ing. Hilario Aradiel Castañeda

**Título y/o Grado:**
**Fecha:** 13/06/2021

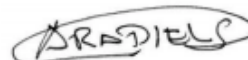
**TÍTULO DE TESIS**
**MARKETPLACE PARA EL PROCESOS DE VENTAS EN LA EMPRESA**
**AC BRANDS & CONSULTING S.A.C.**
**Tabla de Evaluación de Expertos para el Indicador: Calidad de registros generados**

Indicadores	CRITERIOS	Deficiente 0% - 19%	Regular 20% - 39%	Bueno 40% - 60%	Muy Bueno 61% - 80%	Excelente 91% - 100%
Claridad	Está formado con el lenguaje apropiado.				80	
Objetividad	Esta expresado en conducta expresable				80	
Organización	Esta adecuado al avance de la ciencia y la tecnología				80	
Suficiencia	Comprende los aspectos de cantidad y calidad				80	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico				80	
Consistencia	Está basado en aspectos técnicos, científicos acordes a la tecnología adecuada				80	
Coherencia	Entre los índices indicadores y dimensiones				80	
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr				80	
Pertinencia	El instrumento es adecuado al tipo de investigación				80	
Total Promedio					80	

**PROMEDIO DE VALORACIÓN:** 80

**OPCIÓN DE APLICABILIDAD:**
 El instrumento puede ser aplicado, tal como está elaborado.

 El instrumento debe ser mejorado, antes de ser aplicado.



 -----  
**Firma Experto**

**TABLA DE EVALUACIÓN DE EXPERTOS**
**Apellidos y Nombres del experto:** Ing. Guillermo Miguel Johnson Romero

**Título y/o Grado:** \_\_\_\_\_

**Fecha:** 18/06/2021

**TÍTULO DE TESIS**
**MARKETPLACE PARA EL PROCESOS DE VENTAS EN LA EMPRESA  
AC BRANDS & CONSULTING S.A.C.**
**Tabla de Evaluación de Expertos para el Indicador: Ingresos**

Indicadores	CRITERIOS	Deficiente 0% - 19%	Regular 20% - 39%	Bueno 40% - 60%	Muy Bueno 61% - 80%	Excelente 91% - 100%
Claridad	Está formado con el lenguaje apropiado.				70	
Objetividad	Esta expresado en conducta expresable				75	
Organización	Esta adecuado al avance de la ciencia y la tecnología				77	
Suficiencia	Comprende los aspectos de cantidad y calidad				80	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico				72	
Consistencia	Está basado en aspectos técnicos, científicos acordes a la tecnología adecuada				69	
Coherencia	Entre los índices indicadores y dimensiones				78	
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr				77	
Pertinencia	El instrumento es adecuado al tipo de investigación				71	
Total Promedio					74	

**PROMEDIO DE VALORACIÓN:** 74

**OPCIÓN DE APLICABILIDAD:**
 El instrumento puede ser aplicado, tal como está elaborado.

 El instrumento debe ser mejorado, antes de ser aplicado


**Firma Experto**

**TABLA DE EVALUACIÓN DE EXPERTOS**
**Apellidos y Nombres del experto:** Ing. Guillermo Miguel Johnson Romero

**Título y/o Grado:** \_\_\_\_\_

**Fecha:** 18/06/2021

**TÍTULO DE TESIS**
**MARKETPLACE PARA EL PROCESOS DE VENTAS EN LA EMPRESA  
AC BRANDS & CONSULTING S.A.C.**
**Tabla de Evaluación de Expertos para el Indicador:** Calidad de registros generados

Indicadores	CRITERIOS	Deficiente 0% - 19%	Regular 20% - 39%	Bueno 40% - 60%	Muy Bueno 61% - 80%	Excelente 91% - 100%
Claridad	Está formado con el lenguaje apropiado.				65	
Objetividad	Esta expresado en conducta expresable				76	
Organización	Esta adecuado al avance de la ciencia y la tecnología				77	
Suficiencia	Comprende los aspectos de cantidad y calidad				80	
Intencionalidad	Adecuado para valorar aspectos del sistema metodológico y científico				77	
Consistencia	Está basado en aspectos técnicos, científicos acordes a la tecnología adecuada				69	
Coherencia	Entre los índices indicadores y dimensiones				75	
Metodología	Responde al propósito del trabajo bajo los objetivos a lograr				76	
Pertinencia	El instrumento es adecuado al tipo de investigación				80	
Total Promedio					75	

**PROMEDIO DE VALORACIÓN:** 75
**OPCIÓN DE APLICABILIDAD:**
 El instrumento puede ser aplicado, tal como está elaborado.

 El instrumento debe ser mejorado, antes de ser aplicado.


**Firma Experto**

### Anexo 8: Tabla t- Student

Grados de libertad	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123
14	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768
15	0.6912	1.3406	1.7531	2.1315	2.6025	2.9467
16	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208
17	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982
18	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784
19	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609
20	0.6870	1.3253	1.7247	2.0860	2.5280	2.8453
21	0.6864	1.3232	1.7207	2.0796	2.5176	2.8314
22	0.6858	1.3212	1.7171	2.0739	2.5083	2.8188
23	0.6853	1.3195	1.7139	2.0687	2.4999	2.8073
24	0.6848	1.3178	1.7109	2.0639	2.4922	2.7970
25	0.6844	1.3163	1.7081	2.0595	2.4851	2.7874
26	0.6840	1.3150	1.7056	2.0555	2.4786	2.7787
27	0.6837	1.3137	1.7033	2.0518	2.4727	2.7707
28	0.6834	1.3125	1.7011	2.0484	2.4671	2.7633
29	0.6830	1.3114	1.6991	2.0452	2.4620	2.7564
30	0.6828	1.3104	1.6973	2.0423	2.4573	2.7500
31	0.6825	1.3095	1.6955	2.0395	2.4528	2.7440
32	0.6822	1.3086	1.6939	2.0369	2.4487	2.7385
33	0.6820	1.3077	1.6924	2.0345	2.4448	2.7333
34	0.6818	1.3070	1.6909	2.0322	2.4411	2.7284
35	0.6816	1.3062	1.6896	2.0301	2.4377	2.7238
36	0.6814	1.3055	1.6883	2.0281	2.4345	2.7195
37	0.6812	1.3049	1.6871	2.0262	2.4314	2.7154
38	0.6810	1.3042	1.6860	2.0244	2.4286	2.7116
39	0.6808	1.3036	1.6849	2.0227	2.4258	2.7079
40	0.6807	1.3031	1.6839	2.0211	2.4233	2.7045

t

# Anexo 9: Reporte de similitud Turnitin

feedback studio MARZIO JOSUE PEREZ ARBULU TURNITIN - TESIS 10MO - MARKETPLACE PARA EL PROCESO DE...

**UNIVERSIDAD CÉSAR VALLEJO**

**FACULTAD DE INGENIERÍA**  
**ESCUELA PROFESIONAL DE**  
**INGENIERÍA DE SISTEMAS**

**PROYECTO DE INVESTIGACIÓN**  
MARKETPLACE PARA EL PROCESO DE VENTAS  
EN LA EMPRESA AC BRANDS & CONSULTING E.I.R.L.

**AUTOR:**  
Perez Arbulu, Marzio Josué (ORCID: 0000-0001-6115-0246)

**ASESOR:**  
Dr. Pérez Farfán, Iván Martín (ORCID: 0000-0001-5833-9400)

**LÍNEA DE INVESTIGACIÓN:**  
Sistema de información y comunicaciones

**Resumen de coincidencias**

**22 %**

Coincidencia 1 de 44

1	repositorio.ucv.edu.pe	10 %
2	Entregado a Universida...	8 %
3	repositorio.uigv.edu.pe	1 %
4	Entregado a University ...	<1 %
5	Entregado a Universida...	<1 %
6	repositorio.autonoma.e...	<1 %
7	repositorio.unica.edu.pe	<1 %
8	repositorio.upci.edu.pe	<1 %
9	www.theibfr.com	<1 %
10	bibliotecadigital.fgv.br	<1 %

Página: 1 de 43    Número de palabras: 9098    Versión solo texto del informe    Alta resolución    Activado



## Anexo 10: Acta de implementación del Marketplace en la empresa

### CARTA DE AUTORIZACIÓN

29 de octubre del 2021

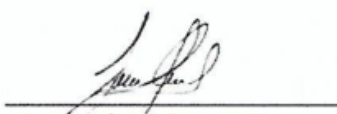
Señores:  
Universidad César Vallejo

De mi mayor consideración:

Yo, Juan Carlos Galagarza Chunga, mediante la presente me es grato dirigirme a Usted a fin de saludarlo cordialmente a nombre de la empresa AC Brands & Consulting E.I.R.L. y a la vez informar el correcto desarrollo e implementación de la plataforma tecnológica, la cual brindó mejoras a nivel tecnológico y económico, perteneciente al proyecto: **"MARKETPLACE PARA EL PROCESO DE VENTAS EN LA EMPRESA AC BRANDS & CONSULTING E.I.R.L."**, al estudiante Marzio Josué Perez Arbulú del ciclo X de la escuela de Ingeniería de Sistemas, al cual depositamos nuestra confianza para desarrollar dicho proyecto y esté a esperas de futuras actualizaciones.

Sin más que decir, me despido y dejo constancia del compromiso de mi persona con el estudiante.

Atentamente,



Juan Carlos Galagarza Chunga  
DNI: 70006137  
Gerente Comercial de AC BRANDS & CONSULTING E.I.R.L.

## **Anexo 11: Desarrollo de la metodología de software**

Este documento detalla la implementación de la metodología de desarrollo de software SCRUM en la investigación titulada “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Este método propone analizar entregables funcionales, las cuales son denominados como Sprints, cada uno de estos entregables serán 100% funcionales y permitirán la implementación de ciertos módulos para su uso sin la necesidad que el software esté completo al 100%.

### **Alcance:**

Según lo analizado, a continuación, se detallan los objetivos del software:

- El primer nivel administrativo corresponde a los usuarios cuyo acceso les permitirá la gestión general de los comercios afiliados además de la información general de los productos y servicios que los mismos ofrecen en la plataforma, asimismo se podrán tener la gestión del stock y la administración de las zonas de reparto las cuales se podrán dibujar en forma de polígonos en un mapa de Google.
- El segundo nivel administrativo corresponde a los comercios afiliados los cuales podrán gestionar sus productos, servicios (para el caso de los entrenador, paseadores o veterinarias afiliadas), inventario, así como gestión de sus ventas o reservas que realicen los clientes desde la plataforma.
- El tercer nivel corresponde a los usuarios de tipo cliente, cuyo rol en la plataforma es poder realizar la compra de los diferentes productos o servicios que se ofrecen en la misma, teniendo un flujo de compra personalizado que les realizar sus transacciones en 3 pasos. Donde el primer paso consiste en verificar la cantidad de productos a comprar o la fecha y hora del servicio a solicitar, el segundo paso consiste en seleccionar

su forma de despacho para cada grupo de productos a adquirir de cada comercio y finalmente como tercer paso la realización de su compra donde podrán seleccionar entre los métodos de pago: Transferencia bancaria o por tarjeta de crédito o débito, donde en este último método de pago se realiza la integración con la pasarela de pagos Culqi.

**Roles:**

Los roles que se tendrán para esta investigación se detallan en el siguiente cuadro:

**Tabla\_ 1 : Nombre y Roles del Proyecto**

<b>ROL</b>	<b>NOMBRE</b>
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

**FUENTE:** Elaboración Propia

## 1. Planificación: Historias de usuario

**Tabla\_ 1: Historia de Usuario 1**

<b>Historia de Usuario: Login de administradores</b>	
<b>Número:</b> 1	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Login de administradores	<b>Tiempo Estimado:</b> 2 días
<b>Iteración:</b> 1	<b>Prioridad:</b> Media
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder ingresar a la plataforma con un usuario y contraseña con el fin de poder ver las opciones que tengo disponibles e interactuar con cada una de ellas	
<b>Criterios de Aceptación:</b>	
<ul style="list-style-type: none"> <li>· El usuario debe ingresar al sistema con credenciales correctas, caso contrario se mostrará un mensaje de error.</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 2: Historia de Usuario 2**

<b>Historia de Usuario:Dashboard</b>	
<b>Número:</b> 2	<b>Usuario:</b> Todos
<b>Nombre Historia:</b> Dashboard	<b>Tiempo Estimado:</b> 2 días
<b>Iteración:</b> 1	<b>Prioridad:</b> Baja
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder visualizar a manera de Dashboard, un reporte de ingresos mensuales a fin de mantenerme al tanto de lo que se ha realizado.	
<b>Criterios de Aceptación:</b>	
<ul style="list-style-type: none"> <li>· Se debe visualizar los gráficos informativos más resaltantes del sistema.</li> <li>· Como vista principal se tiene una interfaz amigable con el usuario y de fácil acceso a las funcionalidades del sistema.</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 3: Historia de Usuario 3**

<b>Historia de Usuario: Administración de comercios</b>	
<b>Número: 3</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de comercios	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 1	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder gestionar los comercios que se afilian a la plataforma con el fin de dar de baja o alta a los nuevos comercios.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"><li>- Listado de comercios.</li><li>- CRUD para la gestión de sucursales.</li><li>- Validación de los datos entrantes.</li></ul>	

Fuente: Elaboración Propia

**Tabla\_ 4: Historia de Usuario 4**

<b>Historia de Usuario: Aprobación de comercio afiliado</b>	
<b>Número: 4</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Aprobar nuevos comercios	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 1	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	
<b>Descripción:</b> Como administrador del Marketplace necesito poder darle de alta al comercio/entrenador registrado con el fin de enviarle sus datos de acceso por correo.	

**Criterio de Aceptación:**

- Lista de comercios o entrenadores/paseadores pendientes de aprobación.
- Cambiar de estado a APROBADO.
- Enviar correo con accesos.

Fuente: Elaboración Propia

**Tabla\_ 5: Historia de Usuario 5**

<b>Historia de Usuario: Administración de categorías</b>	
<b>Número: 5</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de categorías	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 1	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	
<b>Descripción:</b> Como administrador del Marketplace necesito poder realizar el mantenimiento de las categorías de los productos que se van a vender en la plataforma, con el fin de distribuir de manera ordenada los mismos.	
<b>Criterio de Aceptación:</b>	
<ul style="list-style-type: none"> <li>- Listado de categorías existentes.</li> <li>- CRUD para la gestión de categorías.</li> <li>- Validación de los datos entrantes.</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 6: Historia de Usuario 6**

<b>Historia de Usuario: Administración de marcas</b>	
<b>Número: 6</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de marcas	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 1	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	

**Descripción:** Como administrador del Marketplace necesito poder realizar el mantenimiento de las marcas que se van a vender en la plataforma con el fin de distribuir los productos por cada marca.

**Criterio de Aceptación:**

- Listado de marcas existentes.
- CRUD de marcas.
- Validación de los datos entrantes.

Fuente: Elaboración Propia

**Tabla\_ 7: Historia de Usuario 7**

<b>Historia de Usuario: Administración de productos y servicios</b>	
<b>Número: 7</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de productos	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 2	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder visualizar los productos que se ofrecen en la plataforma con el fin de llevar el control de estos.	
<b>Criterio de Aceptación:</b>	
<ul style="list-style-type: none"> <li>- Listado de productos existentes.</li> <li>- CRUD para la gestión de productos</li> <li>- Validación de los datos entrantes.</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 8: Historia de Usuario 8**

<b>Historia de Usuario: Administración de productos y servicios</b>	
<b>Número: 8</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de servicios	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 2	<b>Prioridad:</b> Alta

<b>Programador responsable:</b> Marzio Perez Arbulú
<b>Descripción:</b> Como administrador del Marketplace necesito poder visualizar los servicios que se ofrecen en la plataforma con el fin de llevar el control de estos.
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- Listado de servicios existentes</li> <li>- CRUD para la gestión de servicios.</li> <li>- Validación de los datos entrantes.</li> </ul>

Fuente: Elaboración Propia

**Tabla\_ 9: Historia de Usuario 9**

<b>Historia de Usuario: Administración de clientes</b>	
<b>Número: 9</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de Clientes	<b>Tiempo Estimado:</b> 4 días
<b>Iteración: 2</b>	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder visualizar el listado de clientes registrados en la plataforma con el fin tener una administración correcta de sus datos.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- Listado de clientes existentes.</li> <li>- CRUD para la gestión de clientes.</li> <li>- Validación de los datos entrantes.</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 10: Historia de Usuario 10**

<b>Historia de Usuario: Administración de banners</b>	
<b>Número: 10</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de banners	<b>Tiempo Estimado:</b> 3 días



<b>Iteración:</b> 2	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder registrar los banners que se mostrarán en la plataforma con el fin de poder presentar ofertas o priorizar segmentos de productos.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- Listado de banners.</li> <li>- CRUD de banners en sus diferentes medidas.</li> <li>- Validación de los datos entrantes.</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 11: Historia de Usuario 11**

<b>Historia de Usuario: Administración de cupones</b>	
<b>Número:</b> 11	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de cupones	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 2	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder registrar los cupones de descuento con el fin de poder generar cupones públicos o privados para clientes de la plataforma y poder delimitarlos en un tiempo estimado.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- Listado de cupones de descuento.</li> <li>- CRUD de cupones de descuento.</li> <li>- Registro por rangos de fecha</li> <li>- Validación de los datos entrantes.</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 12: Historia de Usuario 12**

<b>Historia de Usuario: Administración de zonas de reparto</b>	
<b>Número:</b> 12	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de zonas de reparto	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 2	<b>Prioridad:</b> Alta

<b>Programador responsable:</b> Marzio Perez Arbulú
<b>Descripción:</b> Como administrador del Marketplace necesito poder gestionar las zonas de reparto con el fin de poder trazar líneas en un mapa de google y delimitar las zonas a donde se podrán realizar las entregas de los pedidos.
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- Mapa de Google con opción para dibujar polígonos de zonas de reparto.</li> <li>- Formulario con opciones de precios para cálculo de envío en moto o auto de acuerdo con el peso y tiempo.</li> <li>- CRUD de zonas de reparto</li> <li>- Validación de los datos entrantes.</li> </ul>

Fuente: Elaboración Propia

**Tabla\_ 13: Historia de Usuario 13**

<b>Historia de Usuario: Administración de centrales de reparto</b>	
<b>Número: 13</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de centrales de reparto	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 3	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder gestionar las centrales de reparto las cuales se encuentran en cada zona de reparto con el fin de que los clientes puedan venir a recoger sus productos en dicho establecimiento.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- Listado de centrales de reparto.</li> <li>- CRUD de centrales de reparto.</li> <li>- Relación con zonas de reparto.</li> <li>- Debe permitir ingresar una dirección con Google places</li> <li>- Validación de los datos entrantes.</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 14: Historia de Usuario 14**

<b>Historia de Usuario: Ventas</b>	
<b>Número: 14</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Gestión de ventas	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 3	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder visualizar el listado de ventas y su detalle, con el fin de poder conocer el estado de las entregas y pagos que se realizan de las mismas.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"><li>- Listado de las ventas realizadas.</li><li>- Vista de detalle de venta.</li><li>- Opciones para cambiar el estado de cada venta</li></ul> Validación de los datos entrantes.	

Fuente: Elaboración Propia

**Tabla\_ 15: Historia de Usuario 15**

<b>Historia de Usuario: Gestión multimedia</b>	
<b>Número: 15</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Importar archivos multimedia	<b>Tiempo Estimado:</b> 2 días
<b>Iteración:</b> 3	<b>Prioridad:</b> Baja
<b>Programador responsable:</b> Nicolás Segovia Manrique	
<b>Descripción:</b> Como administrador del Marketplace necesito poder gestionar las herramientas multimedia como imágenes para poder asignarle a los banners, productos, categorías u otros elementos de la plataforma.	

**Criterio de Aceptación:**

- Gestión de archivos Multimedia
- Carga de archivo de fotos de forma masiva.
- Validación de los datos entrantes.

Fuente: Elaboración Propia

**Tabla\_ 16: Historia de Usuario 16**

<b>Historia de Usuario: Gestión de perfiles</b>	
<b>Número: 16</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Administración de perfiles y roles administrativos	<b>Tiempo Estimado:</b> 2 días
<b>Iteración:</b> 3	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder gestionar los perfiles y roles de la plataforma con el fin de poder delimitar qué módulos podrán observar dichos usuarios.	
<b>Criterio de Aceptación:</b>	
<ul style="list-style-type: none"> <li>- Listado de roles administrativos.</li> <li>- Asignación de módulos para cada rol.</li> <li>- Validación de los datos entrantes.</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 17: Historia de Usuario 17**

<b>Historia de Usuario: Punto de venta</b>	
<b>Número: 17</b>	<b>Usuario:</b> Admin
<b>Nombre Historia:</b> Módulo de punto de venta	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 3	<b>Prioridad:</b> Muy alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito registrar pedidos a través de un punto de venta con el fin de poder tener un control de los pedidos que vienen por otros canales de atención.	

**Criterio de Aceptación:**

- Formulario para registrar pedidos de forma manual.
- Mostrar listado de clientes registrados u opción para ingresar un nuevo cliente.
- Validación de datos entrantes.

Fuente: Elaboración Propia

Usuario Comercio

**Tabla\_ 18: Historia de Usuario 18**

<b>Historia de Usuario: Login de comercio</b>	
<b>Número: 18</b>	<b>Usuario:</b> Comercio
<b>Nombre Historia:</b> Ingresar como comercio	<b>Tiempo Estimado:</b> 2 días
<b>Iteración:</b> 4	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como comercio afiliado al Marketplace necesito poder ingresar con los datos que el administrador me ha proporcionado con el fin de tener acceso a los diferentes módulos que me correspondan.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- Iniciar sesión</li> <li>- Direccionar al comercio a su zona respectiva</li> <li>- Validación de datos entrantes</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 19: Historia de Usuario 19**

<b>Historia de Usuario: Gestión de productos de comercio</b>	
<b>Número: 19</b>	<b>Usuario:</b> Comercio
<b>Nombre Historia:</b> Administración de productos como comercio	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 4	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	

**Descripción:** Como comercio afiliado al Marketplace necesito poder gestionar mis productos con el fin de que se muestran en la plataforma y los clientes los puedan adquirir.

**Criterio de Aceptación:**

- Listado de productos pertenecientes al comercio
- CRUD de productos
- Validación de datos entrantes

Fuente: Elaboración Propia

**Tabla\_ 20: Historia de Usuario 20**

<b>Historia de Usuario: Gestión de ventas de comercio</b>	
<b>Número: 20</b>	<b>Usuario:</b> Comercio
<b>Nombre Historia:</b> Administración de ventas como comercio	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 4	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como comercio afiliado al Marketplace necesito poder visualizar las ventas que he generado con el fin de poder verificar el detalle de los pedidos que he recibido.	
<b>Criterio de Aceptación:</b>	
<ul style="list-style-type: none"> <li>- Listado de pedidos generados para el comercio afiliado.</li> <li>- Actualizar el estado del despacho</li> <li>- Visualizar el detalle</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 21: Historia de Usuario 21**

<b>Historia de Usuario: Gestión de sucursales del comercio</b>	
<b>Número: 21</b>	<b>Usuario:</b> Comercio
<b>Nombre Historia:</b> Administración de sucursales como comercio	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 4	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	

**Descripción:** Como comercio afiliado al Marketplace necesito poder gestionar las sucursales de mi negocio con el fin de poder tener mis ubicaciones de entrega y almacén de productos.

**Criterio de Aceptación:**

- Listado sucursales por comercio
- CRUD de datos de sucursal
- Validación de datos entrantes

Fuente: Elaboración Propia

**Tabla\_ 22: Historia de Usuario 22**

<b>Historia de Usuario: Gestión de inventario por comercio</b>	
<b>Número: 22</b>	<b>Usuario:</b> Comercio
<b>Nombre Historia:</b> Administración de inventario como comercio	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 5	<b>Prioridad:</b> Muy alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	
<b>Descripción:</b> Como comercio afiliado al Marketplace necesito poder gestionar el inventario de mis productos con el fin de poder generar ingresos y egresos respecto a los productos que tengo en cada sucursal.	
<b>Criterio de Aceptación:</b>	
<ul style="list-style-type: none"> <li>- Listado productos inventariados</li> <li>- Formulario para generar ingreso y egreso</li> <li>- Detalle de movimientos</li> <li>- Validación de datos entrantes</li> </ul>	

Fuente: Elaboración Propia

Usuario Entrenador

**Tabla\_ 23: Historia de Usuario 23**

<b>Historia de Usuario: Gestión de servicios como comercio</b>	
<b>Número: 23</b>	<b>Usuario:</b> Entrenador
<b>Nombre Historia:</b> Administración de servicios como comercio	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 5	<b>Prioridad:</b> Media

<b>Programador responsable:</b> Nicolás Segovia Manrique
<b>Descripción:</b> Como comercio afiliado al Marketplace necesito poder registrar mis servicios con el fin de que los clientes puedan realizar reservas en la plataforma.
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>- Listado servicios</li> <li>- CRUD de servicios</li> <li>- Validación de datos entrantes</li> </ul>

Fuente: Elaboración Propia

**Tabla\_ 24: Historia de Usuario 24**

<b>Historia de Usuario: Calendario de reservas</b>	
<b>Número: 24</b>	<b>Usuario:</b> Entrenador
<b>Nombre Historia:</b> Administración de reservas generadas como comercio	<b>Tiempo Estimado:</b> 4 días
<b>Iteración: 5</b>	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como comercio afiliado al Marketplace necesito poder visualizar mis reservas en un calendario con el fin de poder llevar un control de estas.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Calendario de reservas</li> <li>· Vista de detalle de reserva</li> <li>· Cambio de estado de la reserva</li> <li>· Validación de datos entrantes</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 25: Historia de Usuario 25**

<b>Historia de Usuario: Calendario de reservas</b>	
<b>Número: 25</b>	<b>Usuario:</b> Entrenador
<b>Nombre Historia:</b> Administración de horarios de atención	<b>Tiempo Estimado:</b> 4 días
<b>Iteración: 4</b>	<b>Prioridad:</b> Muy alta



<b>Programador responsable:</b> Marzio Perez Arbulú
<b>Descripción:</b> Como comercio afiliado al Marketplace necesito poder gestionar mis horarios de atención, con el fin de que los clientes sepan mi disponibilidad
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Listado de días</li> <li>· Cambio de horas por días</li> </ul>

Fuente: Elaboración Propia

Usuario Cliente

**Tabla\_ 26: Historia de Usuario 26**

<b>Historia de Usuario: Página de inicio</b>	
<b>Número: 26</b>	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Interacción página de inicio	<b>Tiempo Estimado:</b> 2 días
<b>Iteración:</b> 6	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	
<b>Descripción:</b> Como cliente del Marketplace necesito poder visualizar la página de inicio con la finalidad de interactuar con todos los elementos	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Vista amigable según diseño</li> <li>· Animación de banners y componentes</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 27: Historia de Usuario 27**

<b>Historia de Usuario: Carrito de compras</b>	
<b>Número: 27</b>	<b>Usuario:</b> Cliente

<b>Nombre Historia:</b> Interacción carrito de compras	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 6	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	
<b>Descripción:</b> Como cliente del Marketplace necesito poder agregar productos a mi carrito de compras con el fin de realizar el proceso de compra	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Agregar productos y servicios al carrito de compras</li> <li>· Validar stock de productos a ingresar</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 28: Historia de Usuario 28**

<b>Historia de Usuario: Detalle de productos y servicios</b>	
<b>Número:</b> 28	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Detalle de productos y servicios	<b>Tiempo Estimado:</b> 4 días
<b>Iteración:</b> 6	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	
<b>Descripción:</b> Como cliente del Marketplace deseo poder visualizar el detalle de los productos o servicios con la finalidad de agregarlos a mi carrito de compras	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Detalle de producto o servicio</li> <li>· Agregar producto o servicio al carrito de compras</li> <li>· Validar stock de productos a ingresar</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 29: Historia de Usuario 29**

<b>Historia de Usuario: Registro de cliente</b>
---

<b>Número: 29</b>	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Formulario de registro de cliente	<b>Tiempo Estimado:</b> 2 días
<b>Iteración:</b> 6	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	
<b>Descripción:</b> Como cliente del Marketplace necesito poder registrarme en la plataforma con el fin de poder realizar compras	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Formulario de registro</li> <li>· Solicitar Nombres y Apellidos, E-mail, Contraseña y DNI</li> <li>· Validación de datos entrantes</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 30: Historia de Usuario 30**

<b>Historia de Usuario: Login de cliente</b>	
<b>Número: 30</b>	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Formulario de ingreso de cliente	<b>Tiempo Estimado:</b> 2 días
<b>Iteración:</b> 6	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	
<b>Descripción:</b> Como cliente del Marketplace necesito poder ingresar a la plataforma con mis datos de registro con el fin de poder realizar compras	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Formulario de ingreso</li> <li>· Solicitar E-mail y Contraseña</li> <li>· Validación de datos entrantes</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 31: Historia de Usuario 31**

<b>Historia de Usuario: Perfil de cliente</b>
---

<b>Número: 31</b>	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Perfil de cliente	<b>Tiempo Estimado:</b> 3 días
<b>Iteración:</b> 6	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como cliente del Marketplace necesito poder visualizar mi perfil con el fin de poder actualizar mis datos	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Formulario con datos del cliente</li> <li>· Actualización de datos</li> <li>· Validación de datos entrantes</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 32: Historia de Usuario 32**

<b>Historia de Usuario: Historial de pedidos del cliente</b>	
<b>Número: 32</b>	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Historial de pedidos	<b>Tiempo Estimado:</b> 2 día
<b>Iteración:</b> 7	<b>Prioridad:</b> Alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como cliente del Marketplace necesito poder visualizar mi historial de pedidos con el fin de poder acceder al detalle de los mismos.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Listado de pedidos realizados</li> <li>· Visualizar el detalle de pedidos</li> <li>· Validar que sean los pedidos realizados por el cliente</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 33: Historia de Usuario 33**

<b>Historia de Usuario: Vista de categorías</b>	
<b>Número: 33</b>	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Vista de categoría de productos	<b>Tiempo Estimado:</b> 2 días
<b>Iteración:</b> 7	<b>Prioridad:</b> Muy alta
<b>Programador responsable:</b> Nicolás Segovia Manrique	
<b>Descripción:</b> Como cliente del Marketplace necesito poder visualizar las categorías de los productos con el fin de poder encontrar el/los productos o servicios que deseo adquirir.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Sección de categoría</li> <li>· Mostrar producto con filtros de precios</li> <li>· Opción para agregar al carrito de compras</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 34: Historia de Usuario 34**

<b>Historia de Usuario: Paso 1 carrito de compras</b>	
<b>Número: 34</b>	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Paso 1 proceso de compra	<b>Tiempo Estimado:</b> 1 días
<b>Iteración:</b> 7	<b>Prioridad:</b> Muy alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como cliente del Marketplace necesito poder realizar el primer paso del proceso de compras con el fin de poder validar los productos o servicios que voy adquirir	

**Criterio de Aceptación:**

- Validación de stock de cada producto
- Validación de disponibilidad de servicios en la fecha y hora solicitada
- Validación de cupones de descuento
- Opción para eliminar cada producto o servicio
- Opción para agregar cantidad de productos

Fuente: Elaboración Propia

**Tabla\_ 35: Historia de Usuario 35**

<b>Historia de Usuario: Paso 2 carrito de compras</b>	
<b>Número: 35</b>	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Paso 2 proceso de compra	<b>Tiempo Estimado:</b> 1 días
<b>Iteración: 7</b>	<b>Prioridad:</b> Muy alta
<b>Programador responsable:</b> Marzio Perez Arbulú	

**Descripción:** Como cliente del Marketplace necesito poder realizar el segundo paso del proceso de compras con el fin de poder seleccionar la forma de envío de los productos de cada comercio que estoy adquiriendo

**Criterio de Aceptación:**

- Agrupar productos por comercio
- En el caso de las tiendas permitir la selección de la forma de despacho:
- Recojo en tienda
- Envío por Delivery: Se activa casilla para ingresar la dirección con Google Autocomplete y se valida que la dirección se encuentre en las zonas de reparto. Si se encuentra la zona de reparto de calcula el envío
- Recojo en sede Central: Se debe seleccionar la sede central donde cliente desee recoger los productos

Fuente: Elaboración Propia

**Tabla\_ 36: Historia de Usuario 36**

<b>Historia de Usuario: Paso 3 carrito de compras</b>	
<b>Número: 36</b>	<b>Usuario:</b> Cliente
<b>Nombre Historia:</b> Paso 3 proceso de compra	<b>Tiempo Estimado:</b> 1 días
<b>Iteración: 7</b>	<b>Prioridad:</b> Muy alta

<b>Programador responsable:</b> Marzio Perez Arbulú
<b>Descripción:</b> Como cliente del Marketplace necesito poder realizar el tercer paso del proceso de compras con el fin de poder seleccionar el método de pago de interés
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Seleccionar los datos para el comprobante:</li> <li>· Boleta: Solicitar Nombre, Apellido, Celular, Correo y DNI del cliente. Estos datos se obtienen del perfil del cliente.</li> <li>· Factura: Se solicitan los datos de la Boleta y se adiciona los campos de RUC, Razón social y Dirección fiscal.</li> <li>· Seleccionar la forma de pago:</li> <li>· Transferencia: Mostrar los datos bancarios del Marketplace para el respectivo depósito.</li> <li>· Yape: Mostrar el número y código QR para pagar con Yape</li> <li>· Tajeta de crédito/débito: Se integra con la pasarela de pagos Culqi.</li> <li>· Al finalizar el cliente debe aceptar los Términos y condiciones de forma obligatoria.</li> <li>· Se envía detalle por correo al Cliente, Administrador y cada Vendedor.</li> </ul>

Fuente: Elaboración Propia

Usuario administrador

**Tabla\_ 37: Historia de Usuario 37**

<b>Historia de Usuario: Indicador Ingresos</b>	
<b>Número: 37</b>	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Indicador Ingresos de ventas	<b>Tiempo Estimado:</b> 2 días



<b>Iteración:</b> 7	<b>Prioridad:</b> Muy alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder generar el reporte de calidad de pedidos generados en un determinado periodo de tiempo con el fin de poder saber las comisiones que voy a recibir.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Filtro de rango de fechas, inicio y fin</li> <li>· Descargar información en formato Excel</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 38: Historia de Usuario 38**

<b>Historia de Usuario: Indicador Calidad de pedidos generados</b>	
<b>Número:</b> 38	<b>Usuario:</b> Administrador
<b>Nombre Historia:</b> Indicador Calidad de pedidos generados	<b>Tiempo Estimado:</b> 2 días
<b>Iteración:</b> 7	<b>Prioridad:</b> Muy alta
<b>Programador responsable:</b> Marzio Perez Arbulú	
<b>Descripción:</b> Como administrador del Marketplace necesito poder generar el reporte de calidad de pedidos generados en un determinado periodo de tiempo con el fin de poder determinar si los pedidos se están entregando en las fechas pactadas.	
<b>Criterio de Aceptación:</b> <ul style="list-style-type: none"> <li>· Filtro de rango de fechas, inicio y fin</li> <li>· Descargar información en formato Excel</li> </ul>	

Fuente: Elaboración Propia

**Tabla\_ 39 : Product Backlog**

ITEM	H.U	Nombre de Historia	Iteración	Tiempo Estimado	Prioridad
1	1	Login de administradores	1	2	Media
2	2	Dashboard	1	2	Baja

<b>3</b>	<b>3</b>	Gestión de comercios	1	3	Alta
<b>4</b>	<b>4</b>	Aprobar nuevos comercios	1	3	Alta
<b>5</b>	<b>5</b>	Gestión de categorías	1	3	Alta
<b>6</b>	<b>6</b>	Gestión de marcas	1	3	Alta
<b>7</b>	<b>7</b>	Gestión de productos	2	4	Alta
<b>8</b>	<b>8</b>	Gestión de servicios	2	4	Alta
<b>9</b>	<b>9</b>	Gestión de Clientes	2	4	Alta
<b>10</b>	<b>10</b>	Gestión de banners	2	3	Alta
<b>11</b>	<b>11</b>	Gestión de cupones	2	3	Alta
<b>12</b>	<b>12</b>	Gestión de zonas de reparto	2	3	Alta
<b>13</b>	<b>13</b>	Gestión de centrales de reparto	3	3	Alta
<b>14</b>	<b>14</b>	Gestión de ventas	3	4	Alta
<b>15</b>	<b>15</b>	Importar archivos multimedia	3	2	Alta
<b>16</b>	<b>16</b>	Administración de perfiles y roles administrativos	3	3	Alta
<b>17</b>	<b>17</b>	Módulo de punto de venta	3	4	Alta
<b>18</b>	<b>18</b>	Ingresar como comercio	4	2	Alta
<b>19</b>	<b>19</b>	Administración de productos como comercio	4	3	Alta
<b>20</b>	<b>20</b>	Administración de ventas como comercio	4	3	Alta
<b>21</b>	<b>21</b>	Administración de sucursales como comercio	4	3	Alta

22	22	Administración de inventario como comercio	5	4	Alta
23	23	Administración de servicios como comercio	5	4	Alta
24	24	Administración de reservas generadas como comercio	5	4	Alta
25	25	Administración de horarios de atención	5	4	Alta
26	26	Interacción página de inicio	6	2	Alta
27	27	Interacción carrito de compras	6	4	Alta
28	28	Detalle de productos y servicios	6	4	Alta
29	29	Formulario de registro de cliente	6	2	Alta
30	30	Formulario de ingreso de cliente	6	2	Alta
31	31	Perfil de cliente	6	3	Alta
32	32	Historial de pedidos	7	2	Alta
33	33	Vista de categoría de productos	7	2	Alta
34	34	Paso 1 proceso de compra	7	1	Alta
35	35	Paso 2 proceso de compra	7	1	Alta
36	36	Paso 3 proceso de compra	7	1	Alta
37	37	Indicador Ingresos de ventas	7	2	Media
38	38	Indicador Calidad de pedidos generados	7	2	Media

**Fuente:** Elaboración Propia

**Tabla\_ 40 :Requerimientos Funcionales**

<b>Código</b>	<b>Requerimiento Funcional</b>	<b>Prioridad</b>	<b>Estimación en días</b>	<b>Tiempo Real</b>
<b>RF1</b>	El sistema debe permitir el inicio de sesión a los administradores, por medio de un usuario correo y una contraseña	Mediana	2	2
<b>RF2</b>	El sistema debe permitir la vista de un resumen de reportes para cada usuario, denominado dashboard	Baja	2	2
<b>RF3</b>	El sistema debe permitir registrar, modificar, buscar, consultar y eliminar los comercios que se afilian a la plataforma. En donde se deben tener en cuenta los campos: código, nombre, correo, teléfono, celular, dirección, banco, número de cuenta, número de cuenta cci, comisión, y opciones de envío.	Alta	3	3
<b>RF4</b>	El sistema permite la aprobación de nuevos comercios, y debe enviar sus credenciales por correo	Alta	3	3
<b>RF5</b>	El sistema debe permitir registrar, modificar, buscar, consultar y eliminar las categorías que servirán para el registro de productos, estos además pueden tener una estructura de padre e hijo.	Alta	3	3
<b>RF6</b>	El sistema debe permitir registrar, modificar, buscar, consultar y eliminar las marcas para el registro posterior de los productos	Alta	3	3
<b>RF7</b>	El sistema debe permitir registrar, modificar, buscar, consultar y eliminar los productos que se ofrecen en la plataforma. Con los campos: Sku, nombre, precio, precio de oferta, marca, tienda, producto destacado, categorías, descripción, foto, imagen destacada,	Alta	4	4

	presentaciones y características.			
<b>RF8</b>	El sistema debe permitir registrar, modificar, buscar, consultar y eliminar los servicios que se ofrecen en la plataforma. Con los siguientes campos: sku, nombre, precio, entrenador, categorías, fotos, imagen destacada, presentaciones, características, cargar imágenes, metadatos	Alta	4	4
<b>RF9</b>	El sistema debe permitir registrar, modificar, buscar, consultar y eliminar los clientes, con los campos: nombres, apellidos, tipo de documento, número de documento, teléfono, dirección, email y contraseña.	Alta	4	4
<b>RF10</b>	El sistema debe permitir registrar, modificar, buscar, consultar y eliminar banners informativos, para las diferentes promociones, con los campos: nombre, url, orden, URL de Imagen 1900 x 800, URL de Imagen 1400 x 700, URL de Imagen 800 x 700, URL de Imagen 600 x 700	Alta	3	3
<b>RF11</b>	El sistema permite la gestión de cupones para generar descuentos por parte de los clientes, de manera pública o privada y además permitiendo limitar el tiempo de su uso.	Alta	3	3
<b>RF12</b>	El sistema permite el registro de zonas de reparto, con el fin de poder trazar líneas en un mapa de google y delimitar las zonas a donde se podrán realizar las entregas de los pedidos.	Alta	3	3
<b>RF13</b>	El sistema debe permitir registrar, modificar, buscar, consultar y eliminar de las centrales de reparto, las cuales se encuentran en cada zona de reparto con el fin de que los clientes puedan venir a recoger sus productos en dicho establecimiento,	Alta	3	3

	con los campos: nombre, email, teléfono, zona de reparto, dirección.			
<b>RF14</b>	El sistema permite visualizar las listas de las ventas y su detalle, con el fin de poder conocer el estado de las entregas y pagos que se realizan de las mismas.	Alta	4	4
<b>RF15</b>	El sistema permite gestionar las herramientas multimedia como imágenes para poder asignarle a los banners, productos, categorías u otros elementos de la plataforma.	Alta	2	2
<b>RF16</b>	El sistema permite gestionar los perfiles y roles de la plataforma con el fin de poder delimitar qué módulos podrán observar dichos usuarios	Alta	3	3
<b>RF17</b>	El sistema permite registrar pedidos a través de un punto de venta con el fin de poder tener un control de los pedidos que vienen por otros canales de atención.	Alta	4	4
<b>RF18</b>	El sistema permite ingresar con los datos que el administrador me ha proporcionado con el fin de tener acceso a los diferentes módulos que me correspondan.	Alta	2	2
<b>RF19</b>	El sistema permite gestionar los productos con el fin de que se muestran en la plataforma y los clientes los puedan adquirir, desde el usuario comercio.	Alta	3	3
<b>RF20</b>	El sistema permite visualizar las ventas que he generado con el fin de poder verificar el detalle de los pedidos que he recibido, por parte del usuario comercio	Alta	3	3
<b>RF21</b>	El sistema permite gestionar las sucursales del negocio del comerciante con el fin de poder tener sus ubicaciones de entrega y almacén de productos, por parte del usuario comercio	Alta	3	3

<b>RF22</b>	El sistema permite gestionar el inventario de los productos con el fin de poder generar ingresos y egresos respecto a los productos que tengo en cada sucursal, por parte del usuario comercio	Alta	4	4
<b>RF23</b>	El sistema permite registrar los servicios con el fin de que los clientes puedan realizar reservas en la plataforma, por parte del usuario entrenador	Alta	4	4
<b>RF24</b>	El sistema permite visualizar las reservas en un calendario con el fin de poder llevar un control de las mismas, por el usuario entrenador	Alta	4	4
<b>RF25</b>	El sistema permite gestionar los horarios de atención, con el fin de que los clientes sepan la disponibilidad de los comercios, como usuario entrenador	Alta	4	4
<b>RF26</b>	El sistema permite que el cliente pueda visualizar la página de inicio con la finalidad de interactuar con todos los elementos	Alta	2	2
<b>RF27</b>	El sistema permite poder agregar productos al carrito de compras con el fin de realizar el proceso de compra, con el usuario cliente	Alta	4	4
<b>RF28</b>	El sistema permite al cliente poder visualizar el detalle de los productos o servicios con la finalidad de agregarlos a mi carrito de compras	Alta	4	4
<b>RF29</b>	El sistema permite que los clientes puedan registrarse para poder realizar las compras	Alta	2	2
<b>RF30</b>	El sistema permite que los usuarios registrados puedan iniciar sesión.	Alta	2	2
<b>RF31</b>	El sistema permite que el cliente pueda visualizar y actualizar sus datos en la sección de "mi perfil"	Alta	3	3

<b>RF32</b>	El sistema permite que el cliente pueda visualizar sus pedidos, con el fin de visualizar el detalle de cada uno	Alta	2	2
<b>RF33</b>	El sistema permite que el cliente pueda visualizar las categorías de los productos con el fin de poder encontrar el/los productos o servicios que desea adquirir.	Alta	2	2
<b>RF34</b>	El sistema permite realizar el primer paso del proceso de compras con el fin de poder validar los productos o servicios que va adquirir	Alta	1	1
<b>RF35</b>	El sistema permite realizar el segundo paso del proceso de compras con el fin de poder seleccionar la forma de envío de los productos de cada comercio que va adquiriendo	Alta	1	1
<b>RF36</b>	El sistema permite realizar el tercer paso del proceso de compras con el fin de poder seleccionar el método de pago de interés	Alta	1	1
<b>RF37</b>	El sistema permite generar el reporte de calidad de pedidos generados en un determinado periodo de tiempo con el fin de poder saber las comisiones que va a recibir.	Medi a	2	2
<b>RF38</b>	El sistema permite generar el reporte de calidad de pedidos generados en un determinado periodo de tiempo con el fin de poder determinar si los pedidos se están entregando en las fechas pactadas.	Medi a	2	2

Fuente: Elaboración Propia

**Tabla\_ 41: Requerimientos No Funcionales**

<b>Código</b>	<b>Tipo</b>	<b>Requerimiento No Funcional</b>
<b>RNF1</b>	<b>Usabilidad</b>	El sistema debe ser lo suficientemente intuitivo

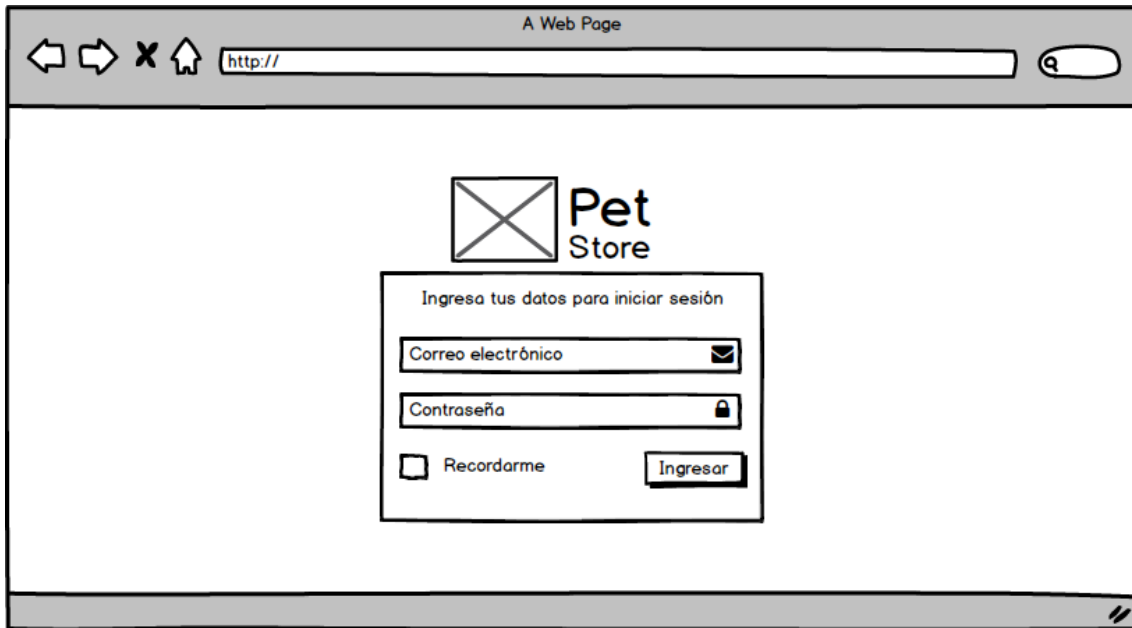


		como para que los usuarios puedan aprender de manera muy sencilla y rápida el uso del mismo
		El sistema debe contener el diseño gráfico bien plasmado y orientado a la línea gráfica de la empresa
		La experiencia de usuario del sistema debe ser amigable y muy sencilla de entender
<b>RNF2</b>	<b>Fiabilidad</b>	El sistema debe garantizar que la información que se maneja es estrictamente sólo analizada por la empresa.
		El sistema debe tener la capacidad de poder soportar ataques externos
<b>RNF3</b>	<b>Rendimiento</b>	El sistema debe ser lo suficientemente rápido y debe soportar la gran cantidad de usuarios que se conectan al mismo tiempo
<b>RNF4</b>	<b>Disponibilidad</b>	El sistema de debe de tener una disponibilidad 24/7 para que no exista ningún problema al conectarse los usuarios desde cualquier lugar en cualquier momento
<b>RNF5</b>	<b>Soporte</b>	El sistema debe ser sencillo de analizar y de entender el código para poder generar un soporte sencillo y rápido
<b>RNF6</b>	<b>Seguridad</b>	El sistema debe permitir y brindar un nivel de Seguridad lo suficientemente bueno como para poder diferenciar las funcionalidades de cada uno de los perfiles y para evitar el robo de información de cualquier tipo

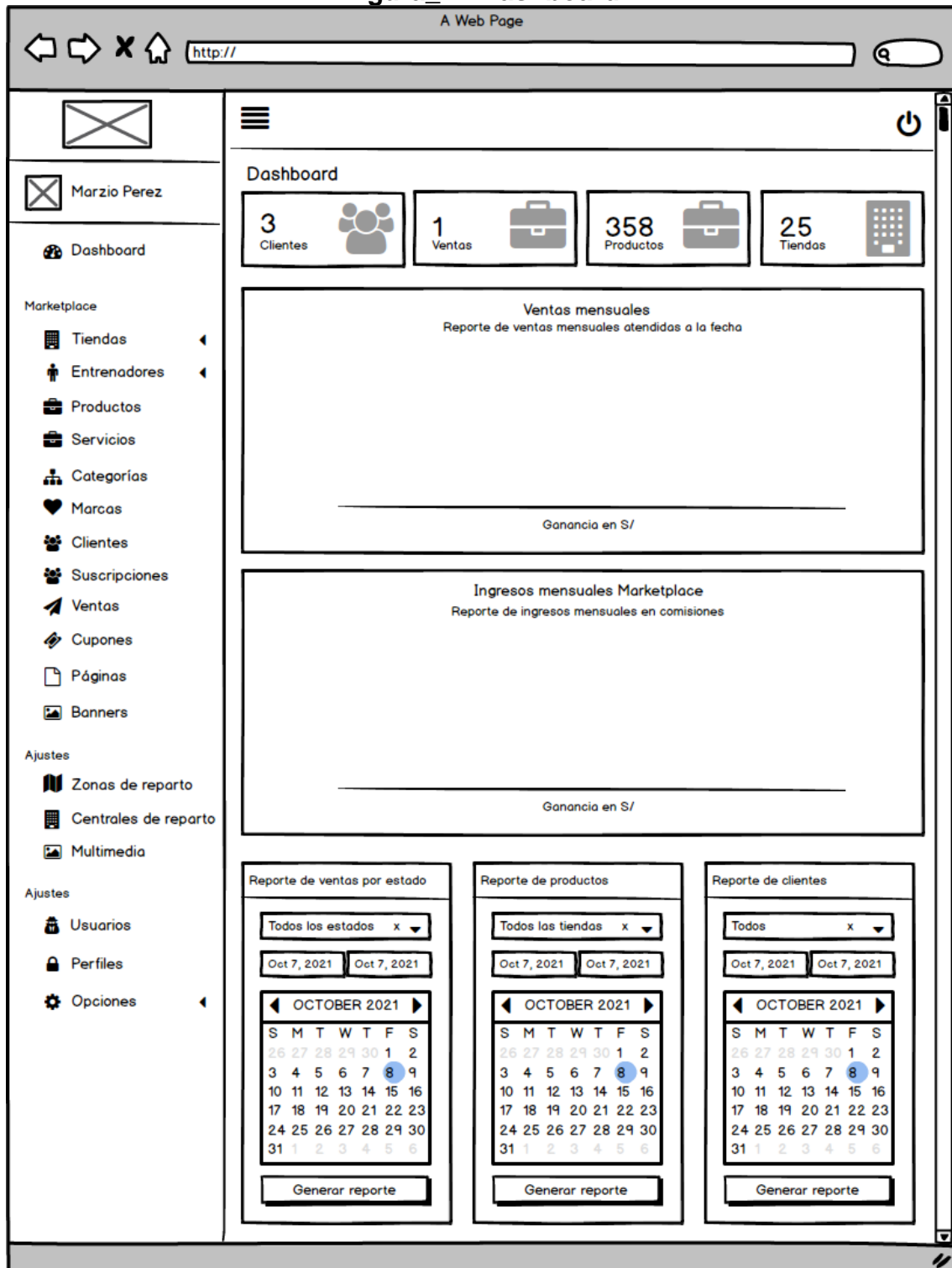
Fuente: Elaboración Propia



Figura\_ 1: Login de administradores



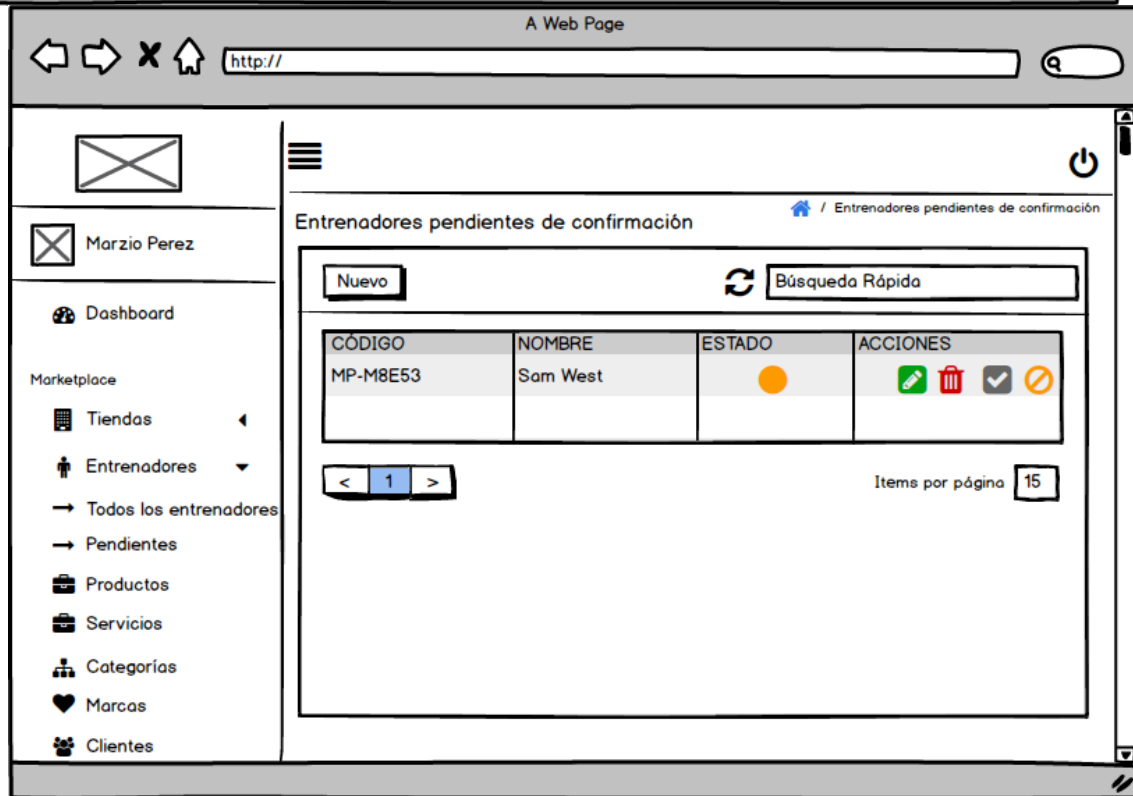
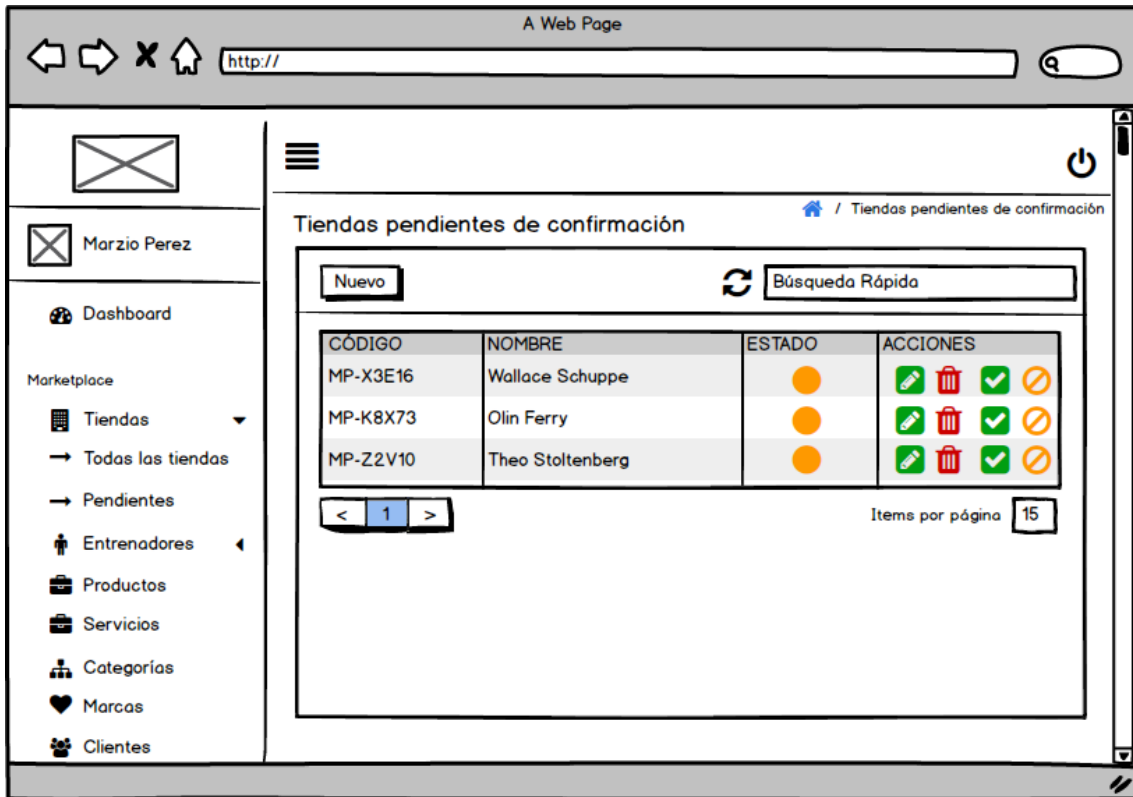
Figura\_ 2: Dashboard



Figura\_ 3: Gestión de comercios



Figura\_ 4: Aprobar nuevos comercios



Figura\_ 5: Gestión de categorías

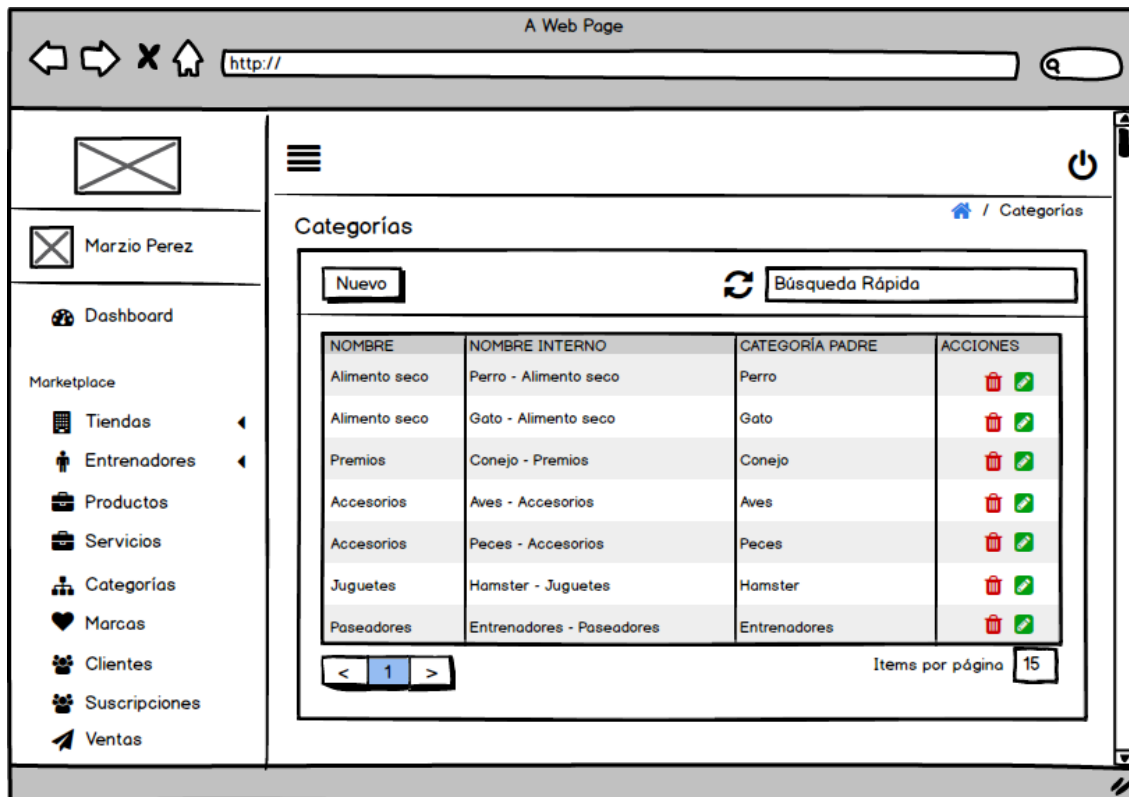
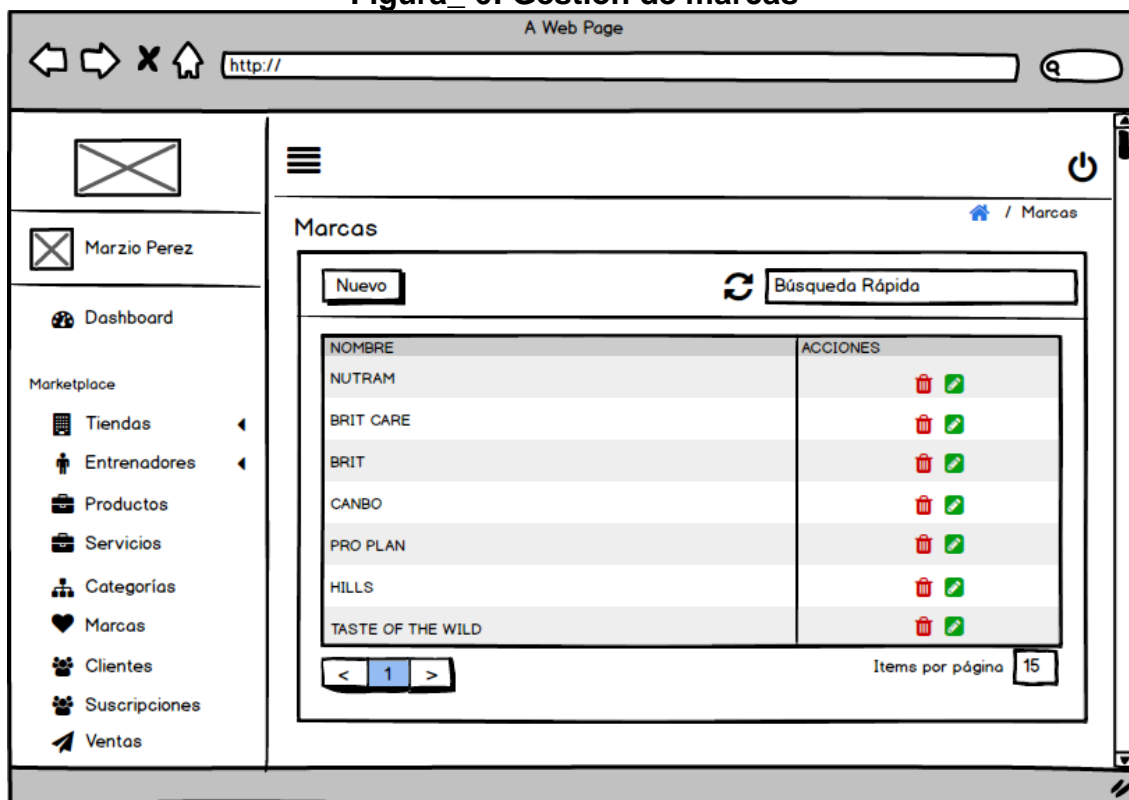


Figura 6: Gestión de marcas



Figura\_ 7: Gestión de productos

Entrenadores pendientes de confirmación

Nuevo

SKU	NOMBRE	PRECIO	MARCA	TIENDA	CATEGORÍAS	ACCIONES
MKT-2FD	Producto 27	97.61	PRO PLAN	Robyn Feest	Entrenador	
MKT-1BA	Producto 22	44.85	TASTE OF THE WILD	Lorenza Purdy	Perro	

< 1 > Items por página 15

Figura\_ 8: Gestión de servicios

Servicios

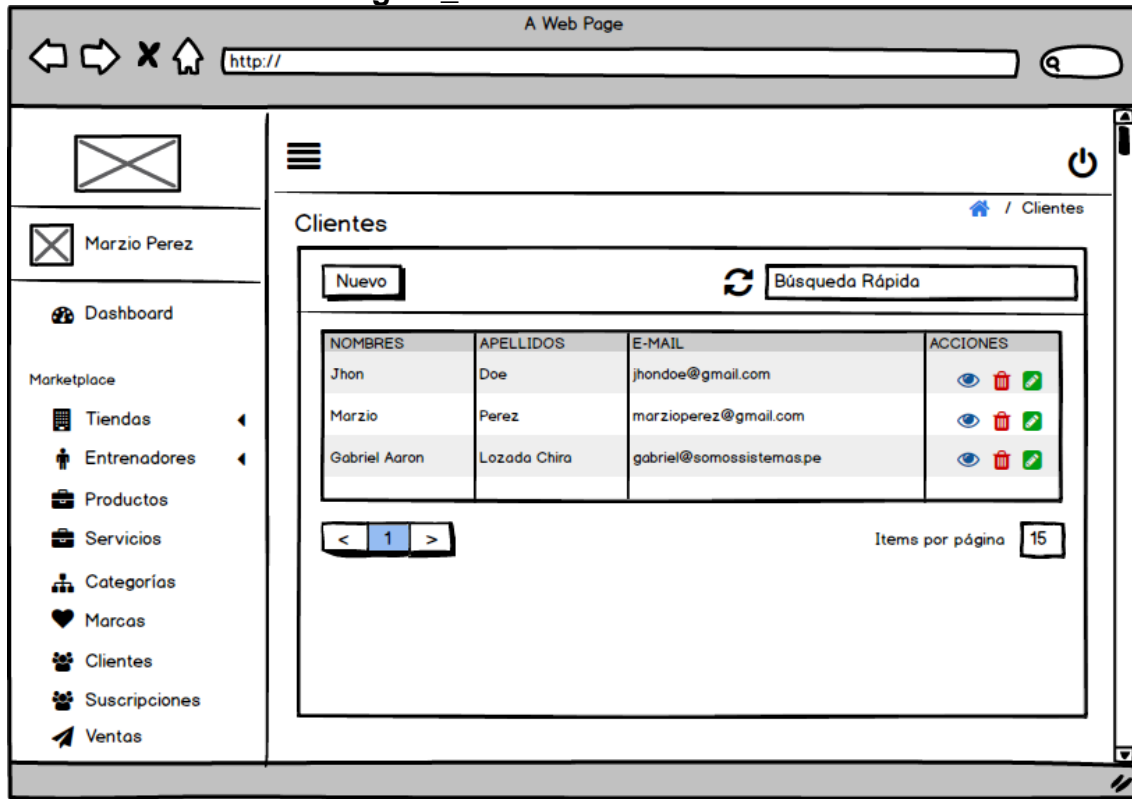
Nuevo

SKU	NOMBRE	PRECIO	ENTRENADOR	CATEGORÍAS	ACCIONES
MKT-8HK	Servicio 1	152.08	Okey Schmidt	Hamster	
MKT-5IP	Servicio 2	20.67	Okey Schmidt	Aves	
MKT-9WG	Servicio 3	32.87	Okey Schmidt	Entrenadore	

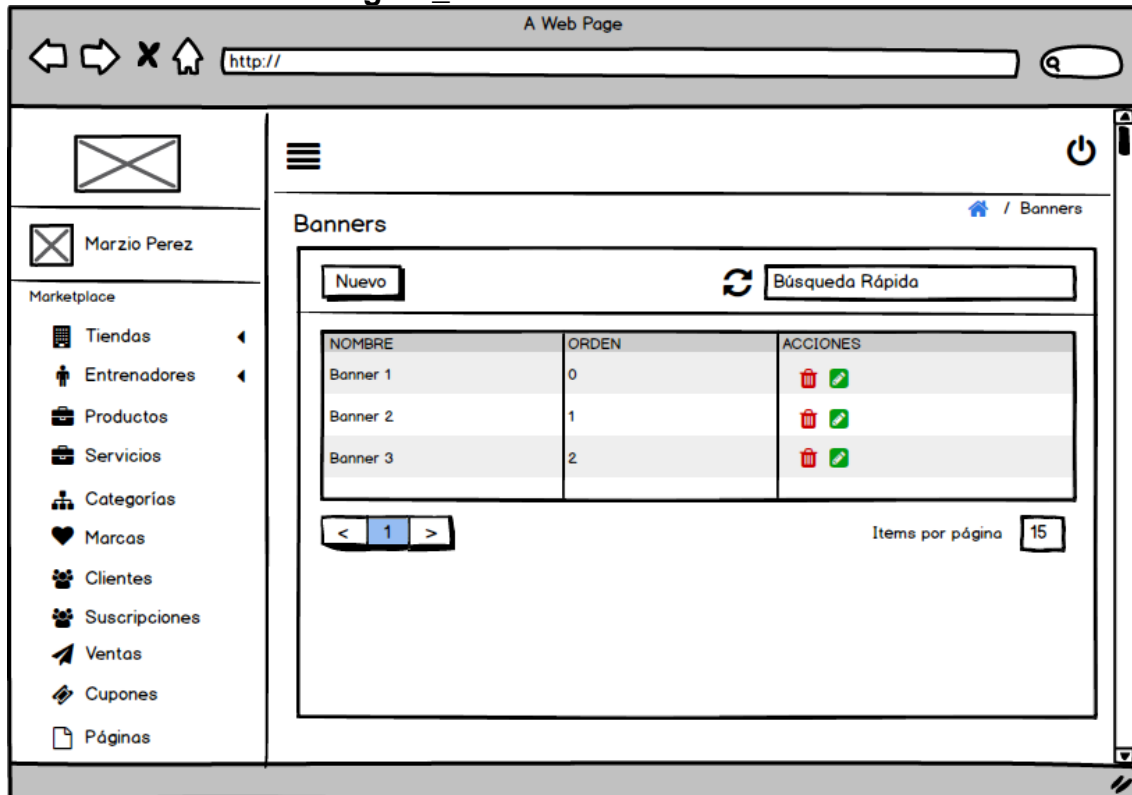
< 1 > Items por página 15



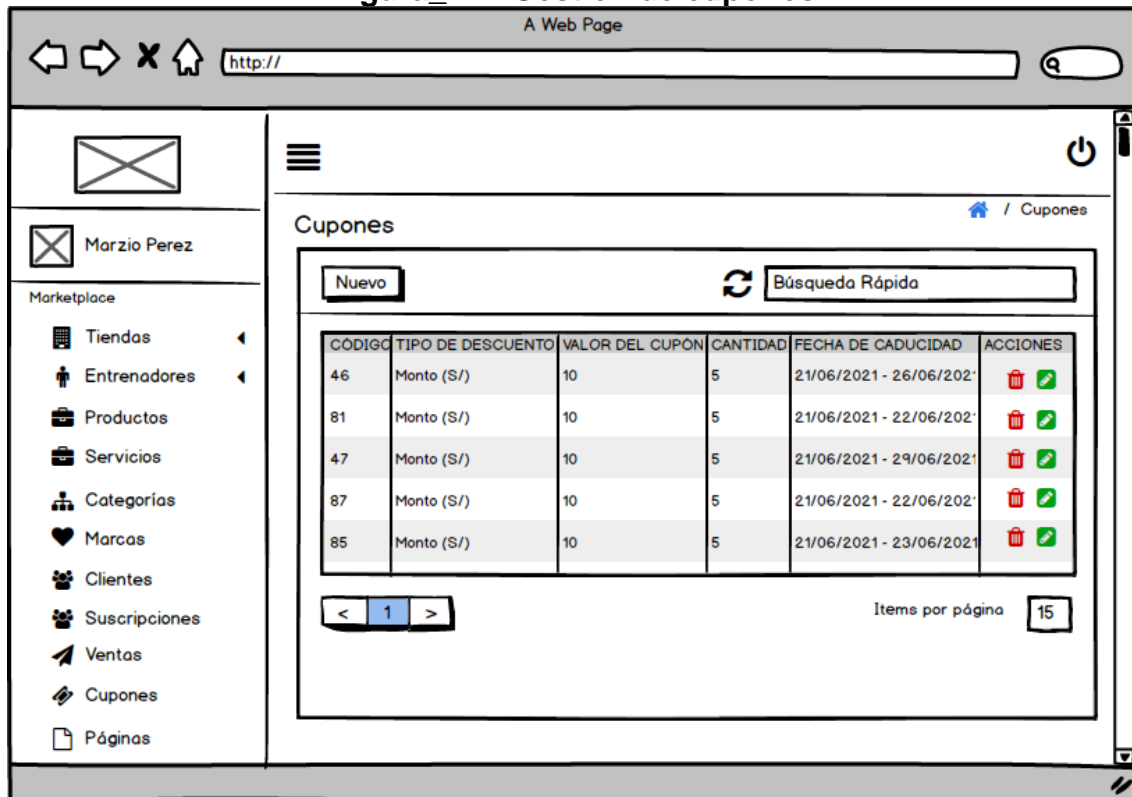
Figura\_ 9: Gestión de Clientes



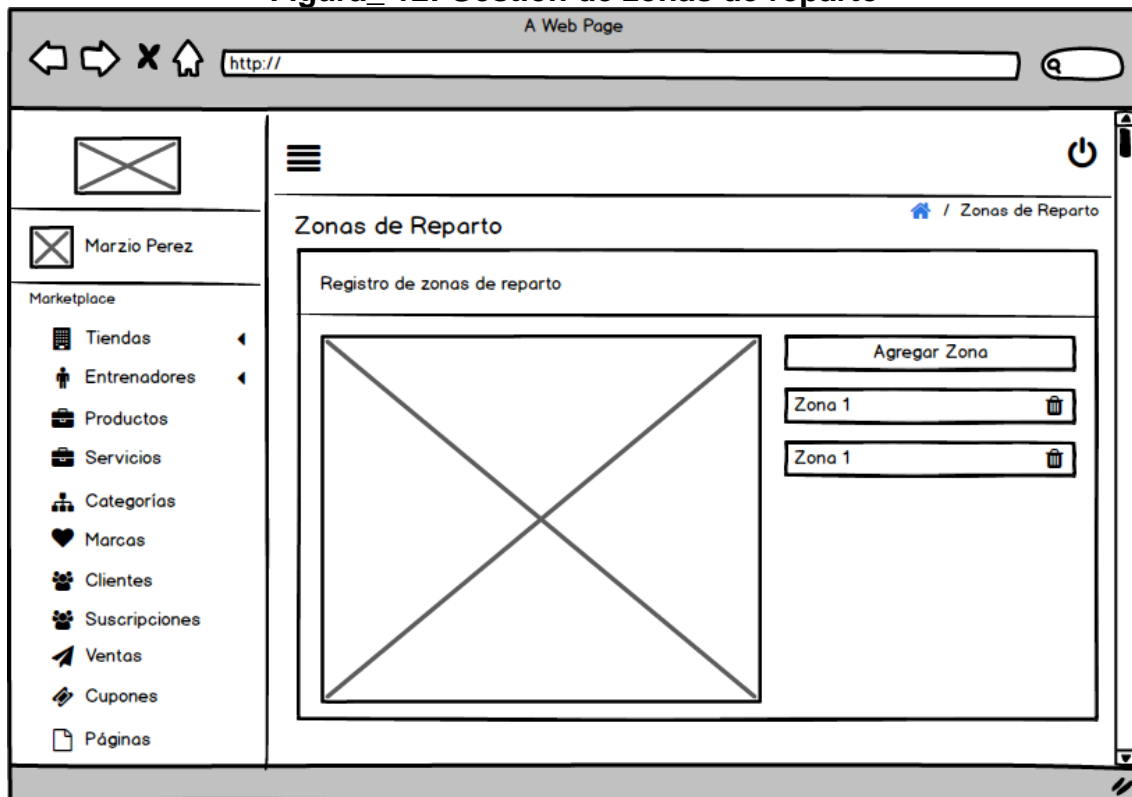
Figura\_ 10: Gestión de banners



Figura\_ 11: Gestión de cupones



Figura\_ 12: Gestión de zonas de reparto



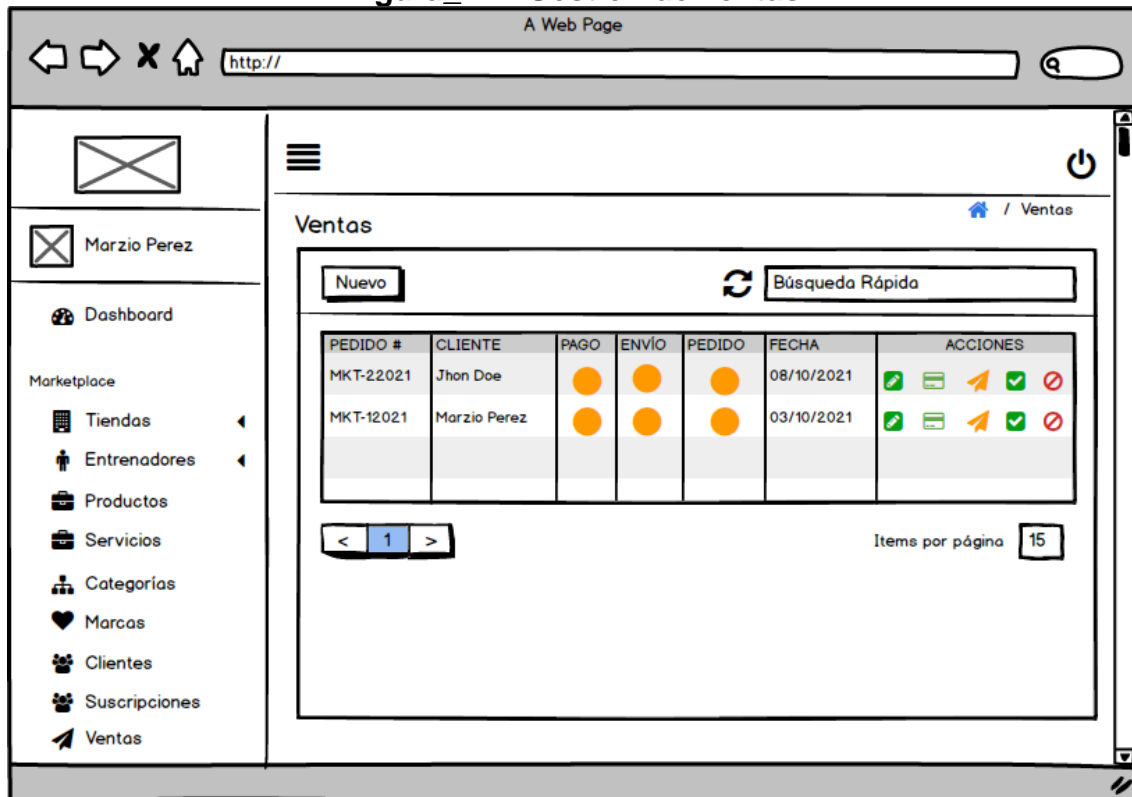
Figura\_ 13: Gestión de centrales de reparto

The screenshot shows a web browser window with the address bar containing "http://". The page title is "A Web Page". The main content area is titled "Centrales de reparto" and features a "Nuevo" button and a "Búsqueda Rápida" search bar. Below these is a table with the following data:

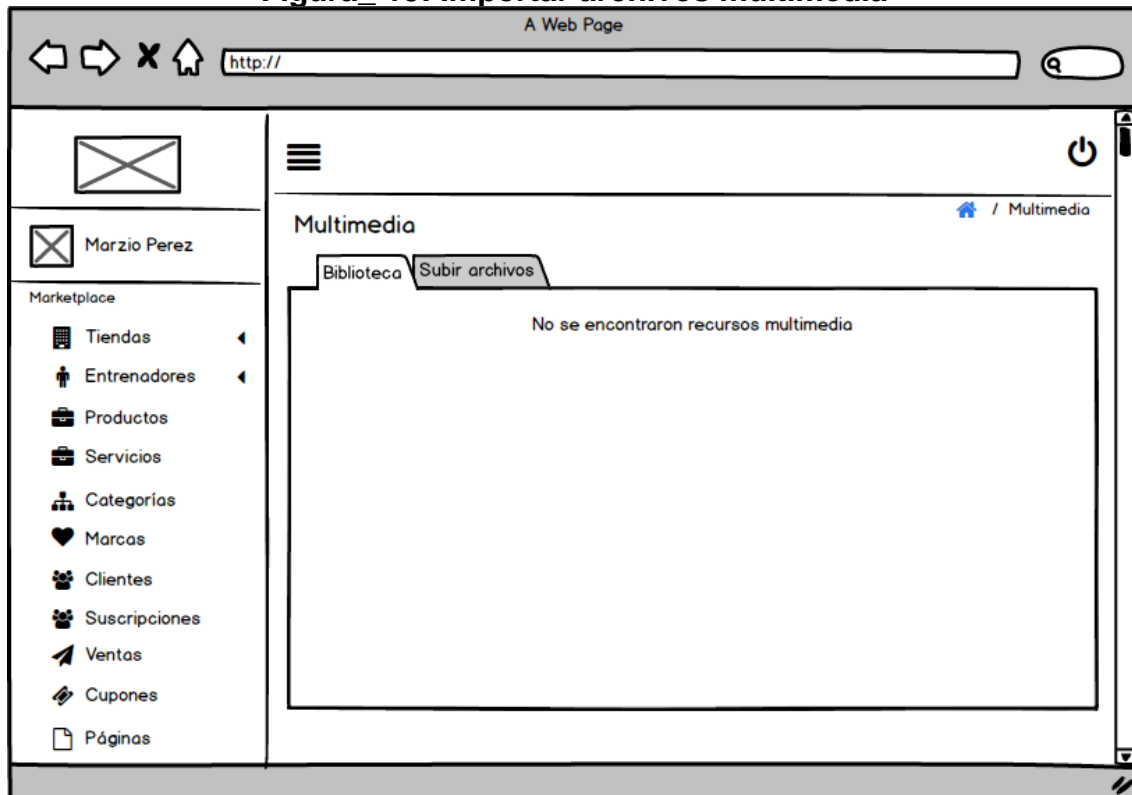
NOMBRE	E-MAIL	TELÉFONO	ZONA DE REPARTO	DIRECCIÓN	ACCIONES
Central Zona 1		5516255050	Zona 1	Dirección zona	
Central Zona 2		5516255051	Zona 2	Dirección zona	

Below the table, there is a pagination control showing "< 1 >" and a label "Items por página" with a value of "15".

Figura\_ 14: Gestión de ventas

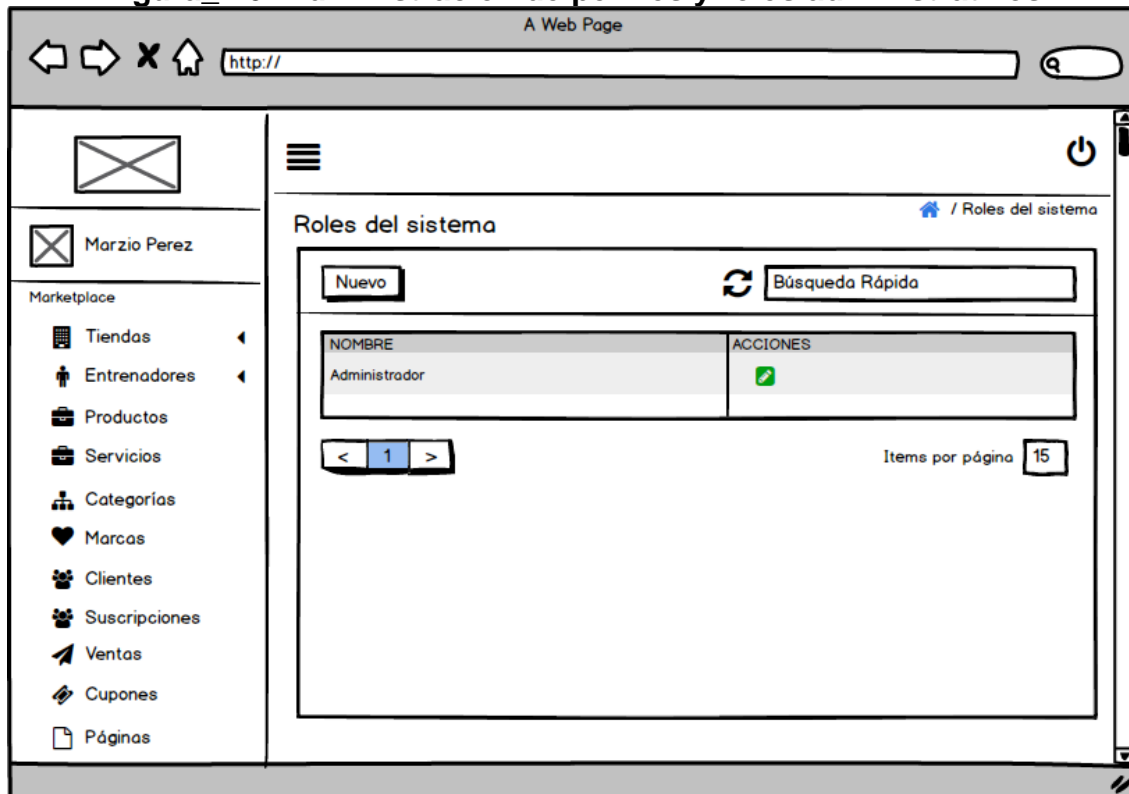


Figura\_ 15: Importar archivos multimedia





Figura\_ 16: Administración de perfiles y roles administrativos



Figura\_ 17: Módulo de punto de venta

**Punto de venta** / Tipos de envío

**Datos del cliente**

Seleccionar cliente... E-mail +51 DNI

**Datos de despacho**

Recojo en sede central  Envío por delivery

Central Zona 1 - Dirección zona

Central Zona 2 - Dirección zona

**Datos del comprobante**

Boleta  Factura

**Detalle de la compra**

Nombre	Cantidad	Precio	Precio Unit.	Importe	Quitar
Escribe el nombre del producto o servicio...					
					Subtotal 0
					Delivery 0
					Total 0

Registrar venta

## USUARIO COMERCIO

Figura\_ 18: Ingresar como comercio

**Pet Store**

Ingresa tus datos para iniciar sesión

Correo electrónico

Contraseña

Recordarme

Figura\_ 19: Administración de productos como comercio

The screenshot shows a web browser window titled 'A Web Page' with a URL bar containing 'http://'. The application interface is divided into a sidebar and a main content area. The sidebar on the left contains a user profile for 'Vendedor Angelica Rodriguez' and a menu with categories: Dashboard, Negocio (containing Productos, Sucursales, Inventario, Ventas, Multimedia), and Ajustes (containing Usuarios, Perfiles). The main content area is titled 'Productos' and features a search bar with a 'Búsqueda Rápida' button and a refresh icon. Below the search bar are two buttons: 'Nuevo' and 'Carga masiva de productos'. A table displays product information with columns for SKU, NOMBRE, PRECIO, MARCA, CATEGORIAS, and ACCIONES. The table contains three rows of data. Below the table is a pagination control showing page 1 of 1 and a dropdown for 'Items por página' set to 15.

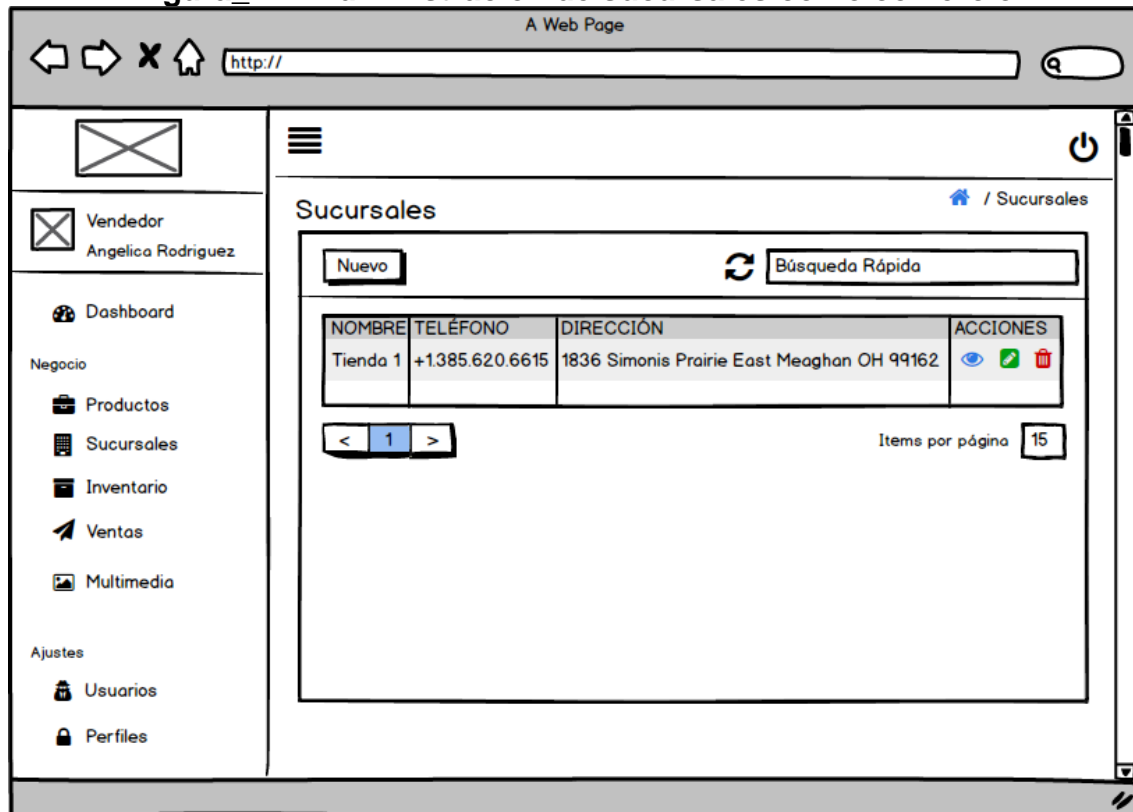
SKU	NOMBRE	PRECIO	MARCA	CATEGORIAS	ACCIONES
MKT-9PL	Producto 1	180.24	PURINA	Hamster	
MKT-7DA	Producto 4	62.76	EARTHBOEN	Perro	
MKT-3SJ	Producto 5	194.01	CANBO	Conejo	

Figura\_ 20: Administración de ventas como comercio

The screenshot shows a web browser window titled 'A Web Page' with a URL bar containing 'http://'. The application interface is similar to Figure 19, with a sidebar for user and business management. The main content area is titled 'Ventas' and features a search bar with a 'Búsqueda Rápida' button and a refresh icon. Below the search bar is a table displaying sales order information with columns for PEDIDO #, CLIENTE, PAGO, ENVÍO, PEDIDO, FECHA, and ACCIONES. The table contains two rows of data. Below the table is a pagination control showing page 1 of 1 and a dropdown for 'Items por página' set to 15.

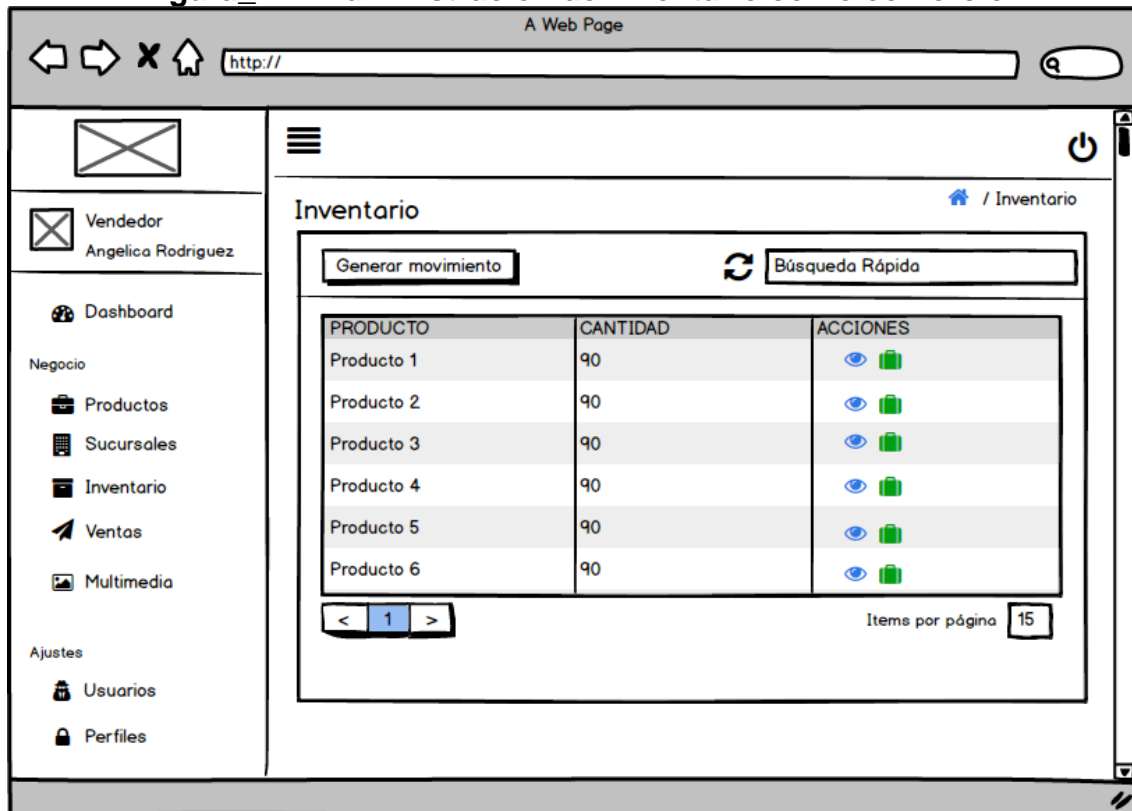
PEDIDO #	CLIENTE	PAGO	ENVÍO	PEDIDO	FECHA	ACCIONES
MKTS-22021	Jhon Doe				08/10/2021	
MKTS-12021	Marzio Perez				03/10/2021	

Figura\_ 21: Administración de sucursales como comercio



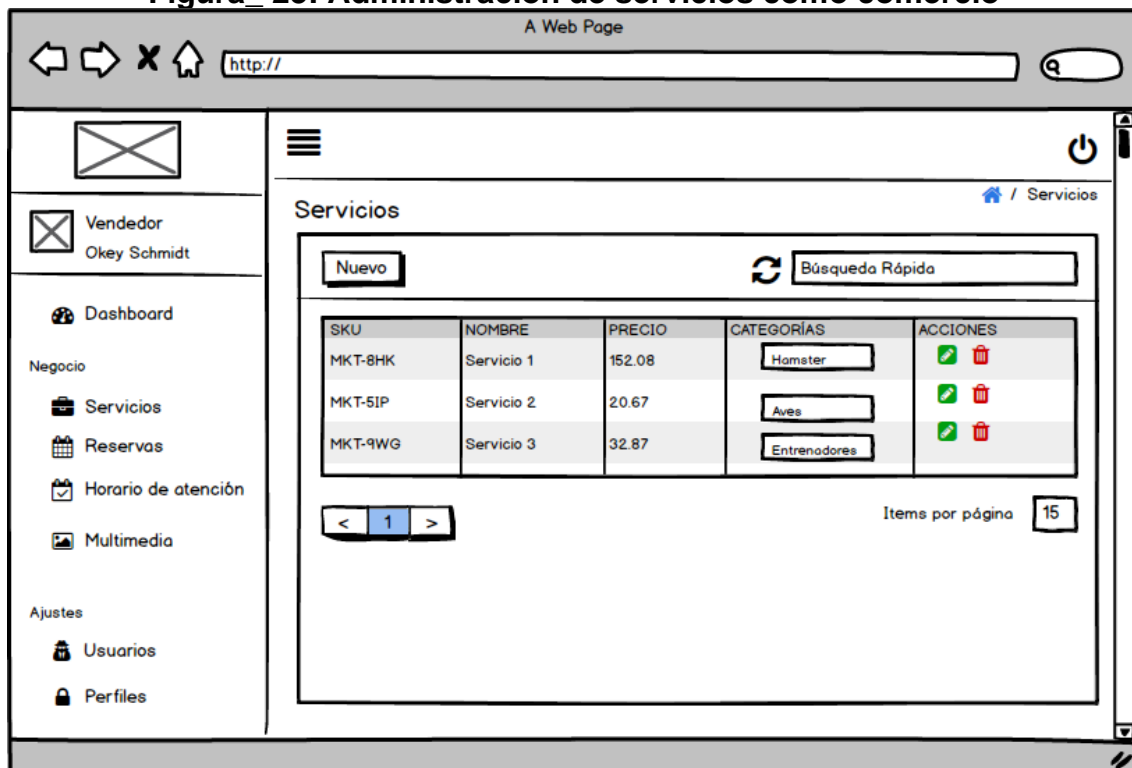


Figura\_ 22: Administración de inventario como comercio



USUARIO ENTRENADOR

Figura\_ 23: Administración de servicios como comercio



Figura\_24: Administración de reservas generadas como comercio

A Web Page

http://

Vendedor  
Okey Schmidt

Dashboard

Negocio

- Servicios
- Reservas
- Horario de atención
- Multimedia

Ajustes

- Usuarios
- Perfiles

Reservas

Todas las reservas Pendiente Atendida Cancelada

Octubre de 2021

OCTOBER 2021						
S	M	T	W	T	F	S
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Todas las reservas Pendiente Atendida Cancelada

Figura\_ 25: Administración de horarios de atención

A Web Page

http://

Vendedor  
Okey Schmidt

Dashboard

Negocio

- Servicios
- Reservas
- Horario de atención
- Multimedia

Ajustes

- Usuarios
- Perfiles

### Horarios de atención

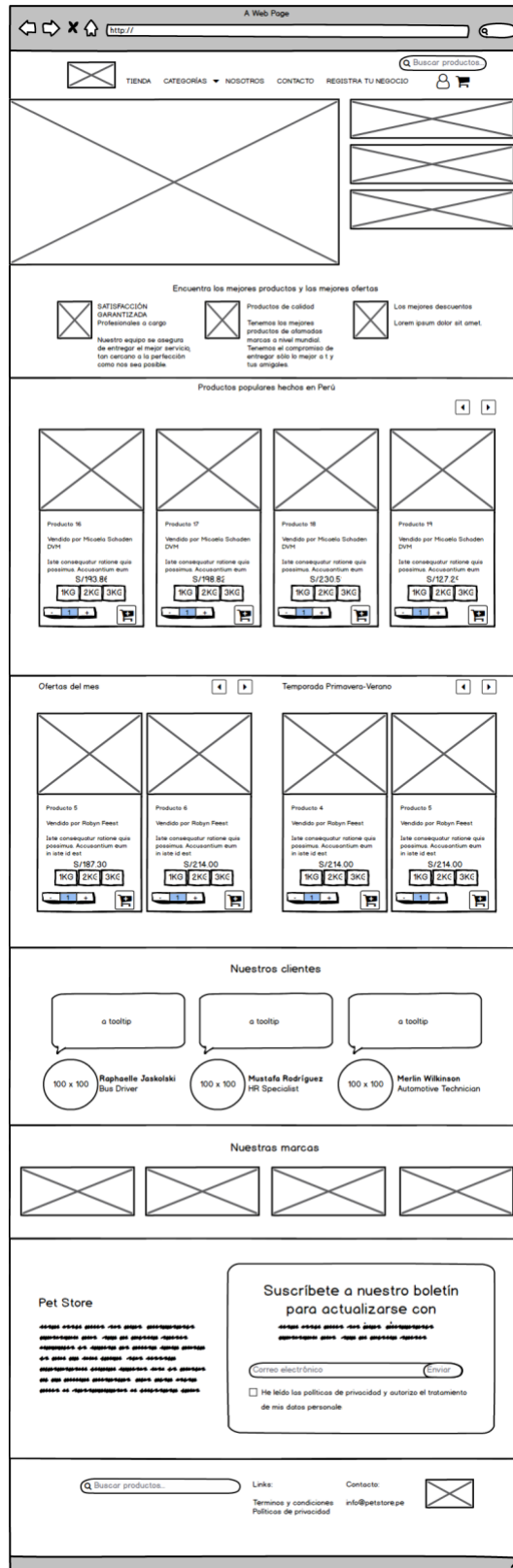
Ingresar el rango de horarios de atención para cada día, esta información se le mostrará al cliente al momento de reservar una cita de entrenamiento o paseo. Recuerda que esta información debe estar actualizada según tu disponibilidad de tiempo, ten cuenta tu horario de refrigerio para evitar problemas con el cliente. Ten en cuenta que los rangos deben ser de menor a mayor.

Lunes	+	09:00	13:00	x	09:00	13:00	x
Martes	+	09:00	13:00	x	09:00	13:00	x
Miércoles	+	09:00	13:00	x	09:00	13:00	x
Jueves	+	09:00	13:00	x	09:00	13:00	x
Viernes	+	09:00	13:00	x	09:00	13:00	x
Sábado	+	09:00	13:00	x	09:00	13:00	x
Domingo	+	09:00	13:00	x	09:00	13:00	x

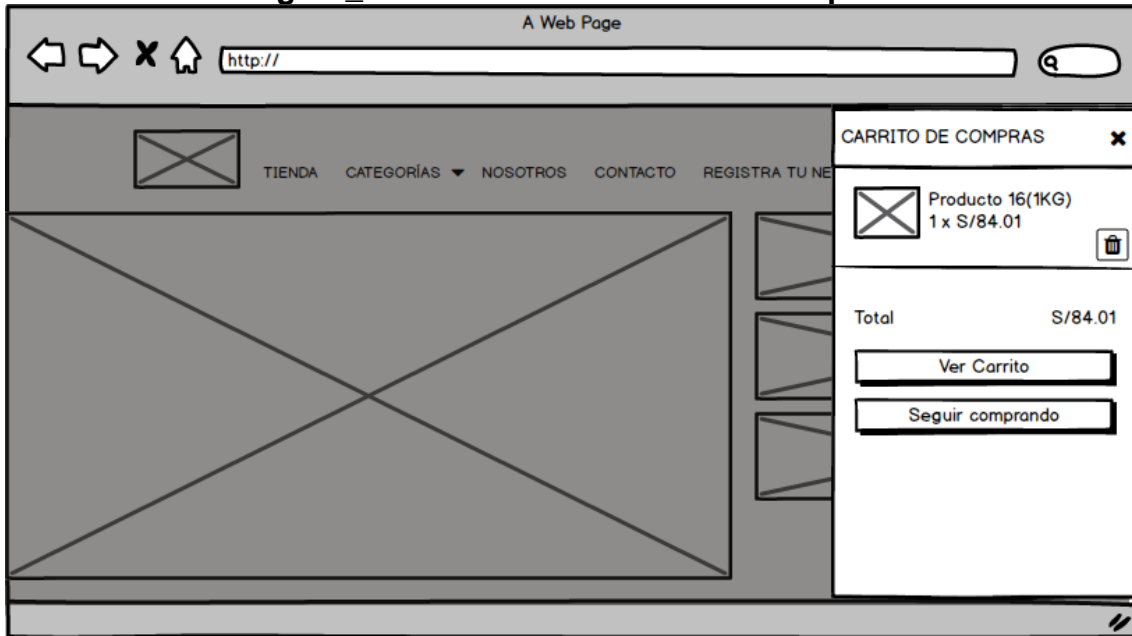
Guardar información de horarios

USUARIO CLIENTE

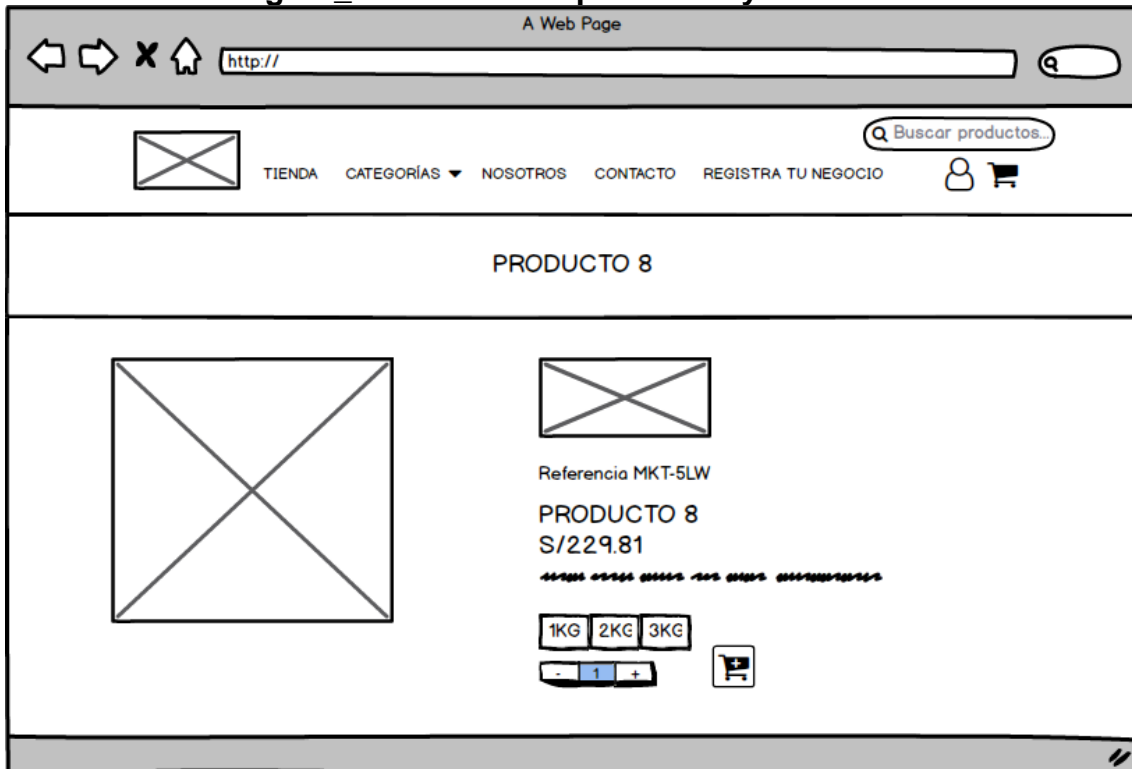
Figura\_ 26: Interacción página de inicio



Figura\_ 27: Interacción carrito de compras



Figura\_ 28: Detalle de productos y servicios



Figura\_ 29: Formulario de registro de cliente

A Web Page

http://

Buscar productos...

TIENDA CATEGORÍAS ▼ NOSOTROS CONTACTO REGISTRA TU NEGOCIO

Ingresar **Regístrate**

Regístrate en nuestro sitio

Nombres Apellidos

E-mail Contraseña

DNI

**REGISTRARME**

Encuentra los mejores productos y las mejores ofertas

SATISFACCIÓN GARANTIZADA Profesionales a cargo

Productos de calidad Tenemos los mejores productos de afamadas

Los mejores descuentos Lorem ipsum dolor sit amet.

Figura\_ 30: Formulario de ingreso de cliente

A Web Page

http://

Buscar productos...

TIENDA CATEGORÍAS ▼ NOSOTROS CONTACTO REGISTRA TU NEGOCIO

**Ingresar** Regístrate

Inicia sesión con tu cuenta

Nombres

E-mail

**INGRESAR**

¿Olvidaste su contraseña?

Encuentra los mejores productos y las mejores ofertas

SATISFACCIÓN GARANTIZADA Profesionales a cargo

Productos de calidad Tenemos los mejores productos de afamadas

Los mejores descuentos Lorem ipsum dolor sit amet.

Figura\_ 31: Perfil de cliente

Actualiza tus datos

Nombres: Jhon

Apellidos: Doe

DNI: 47804233

Celular: 95037541

E-mail: jhondoe@gmail.com

Contraseña: Contraseña

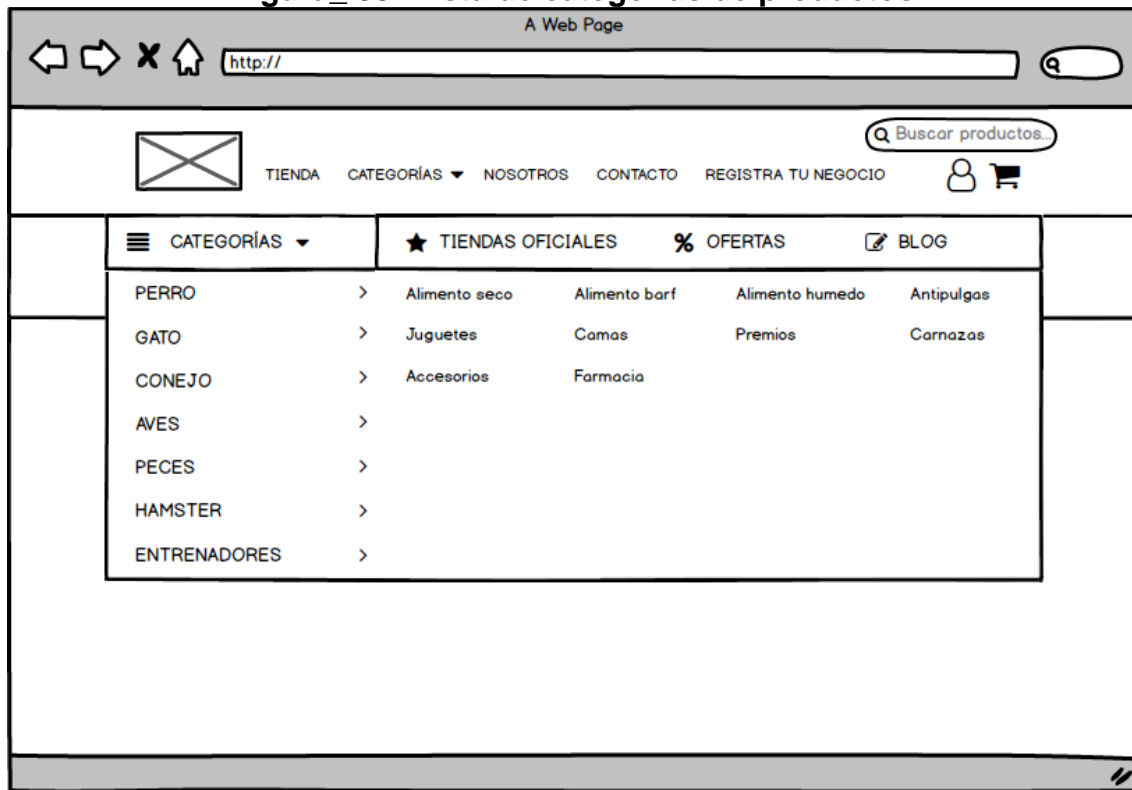
GUARDAR

Figura\_ 32: Historial de pedidos

MIS PEDIDOS

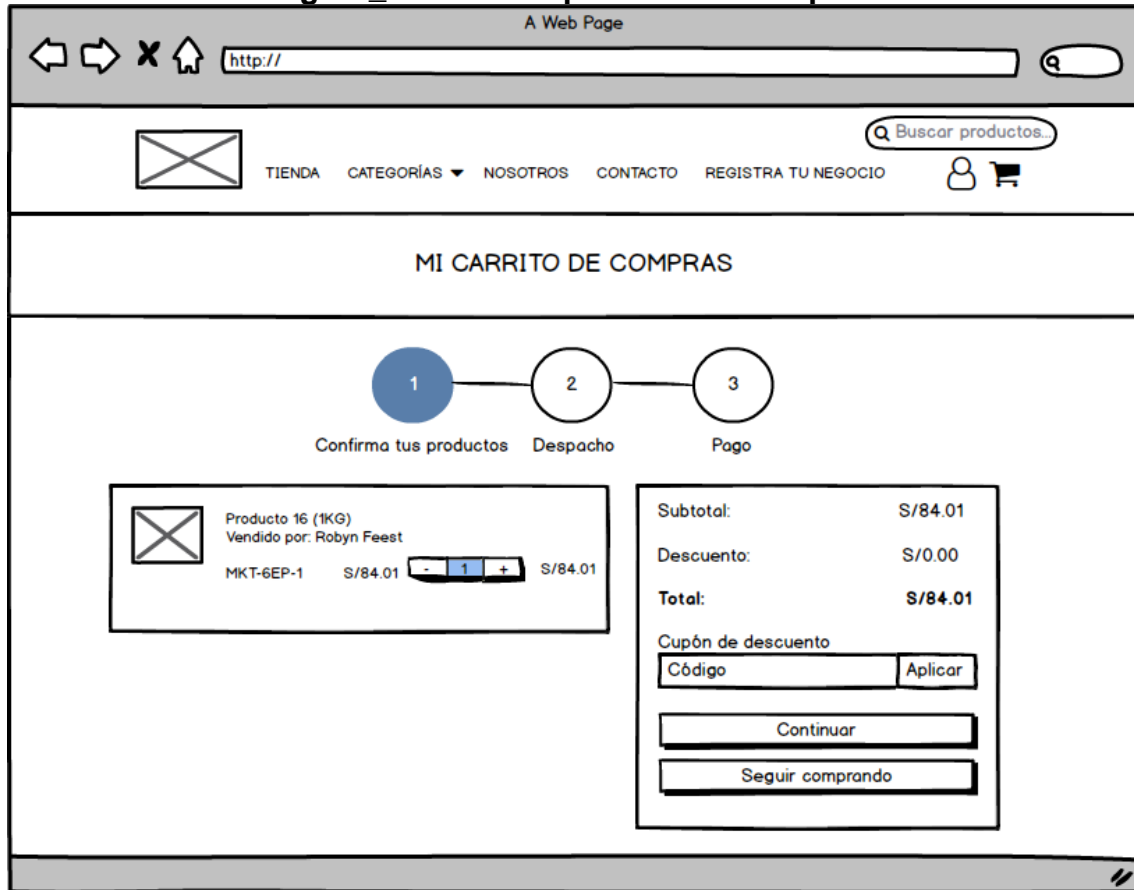
PEDIDO	FECHA	TOTAL	ESTADO	VER
MKT-22021	07/10/2021	S/454.74	Pendiente	

Figura\_ 33: Vista de categorías de productos

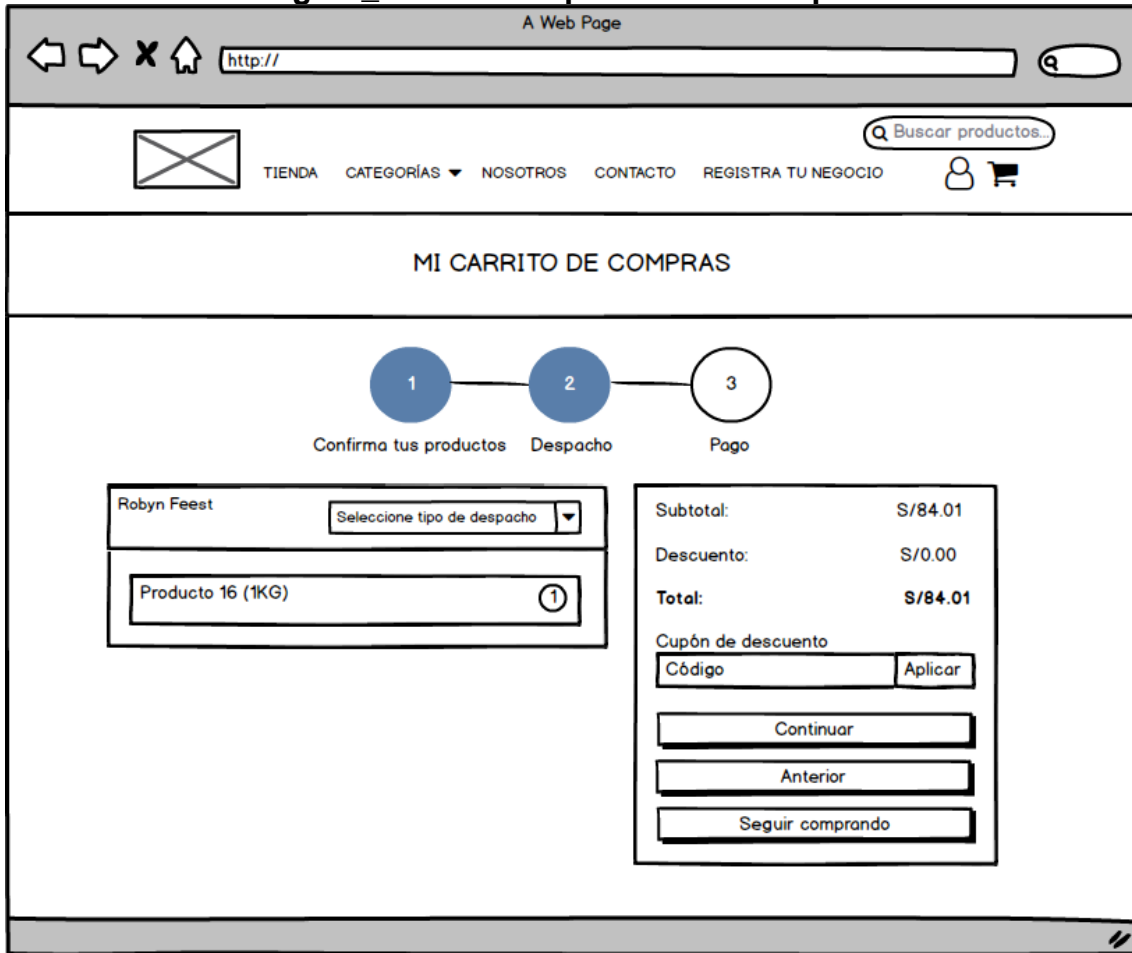




Figura\_ 34: Paso 1 proceso de compras



Figura\_ 35: Paso 2 proceso de compras



Figura\_ 36: Paso 3 proceso de compras

A Web Page

http://

Buscar productos...

TIENDA CATEGORÍAS ▼ NOSOTROS CONTACTO REGISTRA TU NEGOCIO

### MI CARRITO DE COMPRAS

1 Confirma tus productos 2 Despacho 3 Pago

**DATOS DE PARA EL COMPROBANTE**

Boleta  Factura

Jhon Jhon Jhon

**SELECCIONA TU FORMA DE PAGO**

Transferencia  Yape  Tarjeta

**Siga los siguientes pasos:**

1. Puede realizar el pago por medio de transferencia bancaria o en ventanilla a los siguientes números de cuenta a nombre de PET STORE SAC:

**Continental:**  
Soles: 0011-0191-0100042320

**BCP:**  
Soles: 193-2558169-0-89

2. Enviar la constancia de pago o transferencia al mail: [ventas@petstore.pe](mailto:ventas@petstore.pe) indicando:

- Nombres y Apellidos.
- RUC o DNI.
- Adjuntar imagen del voucher o constancia de transferencia.

Pet Store responderá dentro de las próximas 24 horas confirmando su pedido.

Acepto los [términos y condiciones](#)

Subtotal: S/84.01  
Descuento: S/0.00  
Total: S/84.01

Cupón de descuento

Código Aplicar

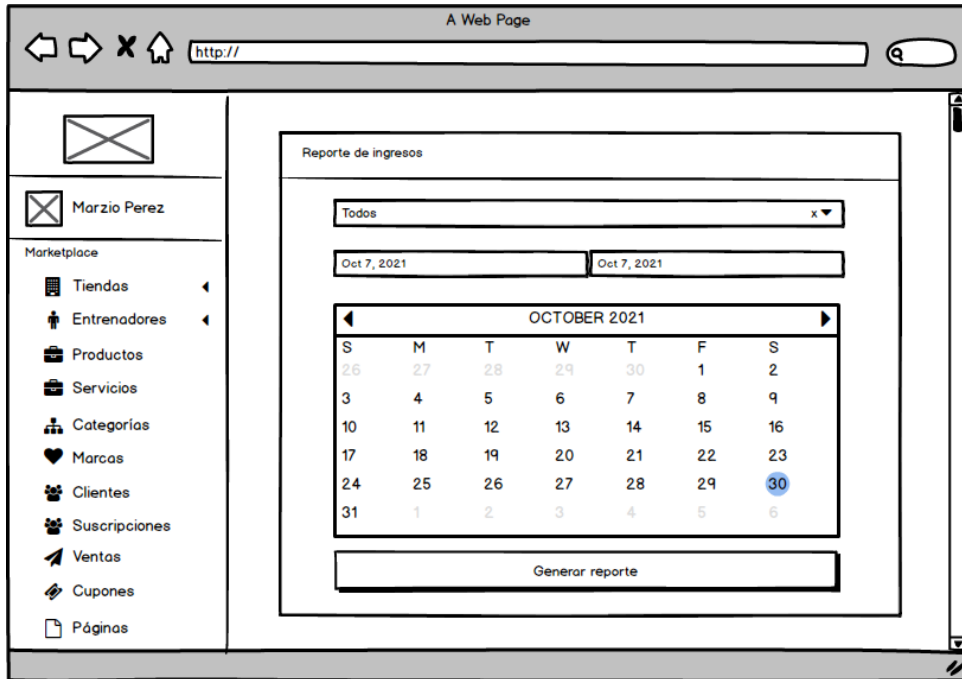
Continuar

Anterior

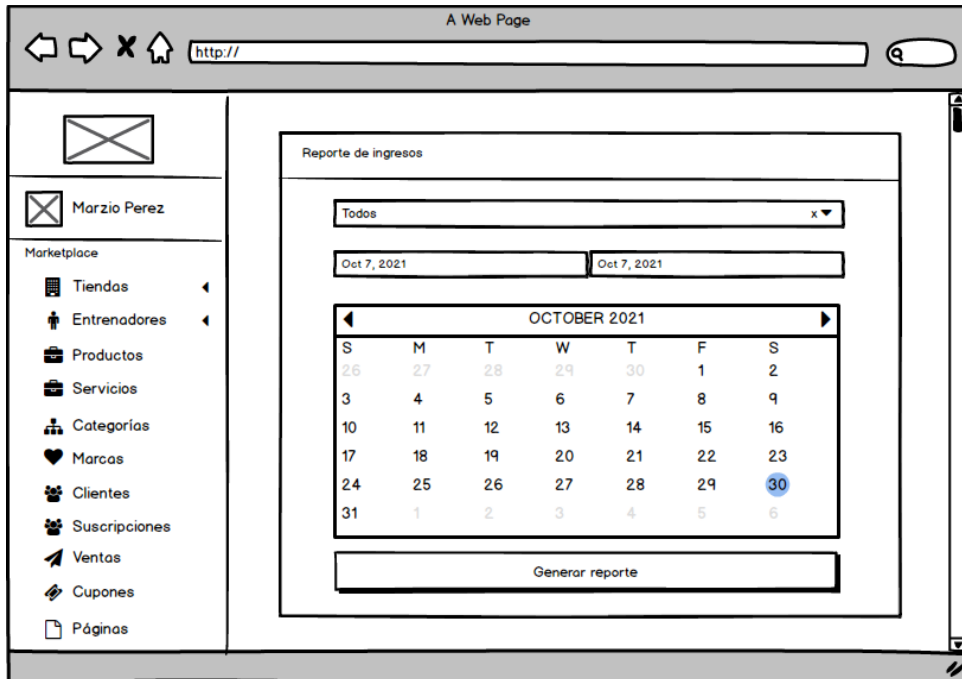
Seguir comprando

# USUARIO ADMINISTRADOR

## Figura\_ 37: Indicador Ingresos



## Figura\_ 38: Indicador Calidad de pedidos generados



## PLANIFICACIÓN DEL SPRINT N°1

Siendo las 4 pm del día 21 de mayo del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.

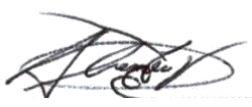
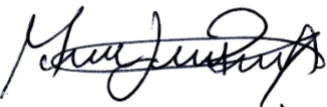


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El gerente de la Empresa AC Brands & Consulting E.I.R.L., efectúa la presentación de los requerimientos, de igual manera manifiesta los requerimientos que tienen mayor prioridad.

Realizado el análisis de los requerimientos presentados por el gerente de AC Brands & Consulting E.I.R.L., el señor Juan Galarza aclara dudas existentes y luego se compromete al cumplimiento de todos los requerimientos propuestas en el Sprint 1.

Los participantes de la reunión manifiestan su aprobación con respecto a la presentación del Sprint 1, así mismo especificando que la fecha de entrega será el día 5 de junio del 2021.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## EJECUCIÓN DEL SPRINT 1

Lo primero que se debe realizar antes del diseño, es conocer y comprender el funcionamiento del sistema, en otras palabras, analizar lo que realmente se necesita, lo cual será elaborado según las historias de usuario.

**RF1: El sistema debe permitir el inicio de sesión a los administradores, por medio de un usuario correo y una contraseña**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 39: Prototipo de Login de administradores**



The image shows a wireframe of a login page for 'Pet Store' within a browser window. The browser title is 'A Web Page' and the address bar contains 'http://'. The page features a logo with an envelope icon and the text 'Pet Store'. Below the logo is a login form with the heading 'Ingresa tus datos para iniciar sesión'. The form includes two input fields: 'Correo electrónico' with an envelope icon and 'Contraseña' with a lock icon. There is a checkbox labeled 'Recordarme' and a button labeled 'Ingresar'.

## **DESARROLLO**

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura 40: Controlador de Login de administradores

```
AdminLoginController.php 2, M X
app > Http > Controllers > Backoffice > Auth > AdminLoginController.php > AdminLoginController
1 <?php
2
3 namespace App\Http\Controllers\Backoffice\Auth;
4
5 use App\Http\Controllers\Controller;
6 use Illuminate\Foundation\Auth\AuthenticatesUsers;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Auth;
9
10 class AdminLoginController extends Controller {
11
12     use AuthenticatesUsers;
13     protected $redirectTo = "/admin/dashboard";
14
15     public function showLoginForm() {
16         if(Auth::guard('admin')->check()){
17             return redirect()->to(route('admin.dashboard'));
18         }
19         return view('back-office.auth.admin-login');
20     }
21
22     public function login(Request $request) {
23         $this->validator($request);
24
25         if(Auth::guard('admin')->attempt($request->only('email','password'),$request->filled('remember'))){
26             return redirect()->intended(route('admin.dashboard'));
27         }
28         return $this->loginFailed();
29     }
30
31     public function logout() {
32         Auth::guard('admin')->logout();
33         return redirect()->route('admin.login');
34     }
35
36     private function validator(Request $request) {
37         $rules = [
38             'email' => 'required|email|exists:admins|min:5|max:191',
39             'password' => 'required|string|min:4|max:255',
40         ];
41
42         $messages = [
43             'email.exists' => 'These credentials do not match our records.',
44         ];
45     }
46 }
```

Figura\_ 41: Modelo de Login de administradores

```
app > Models > Admin.php > Admin
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Foundation\Auth\User as Authenticatable;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8  use Illuminate\Notifications\Notifiable;
9
10 class Admin extends Authenticatable {
11
12     use HasFactory;
13     use Notifiable;
14     use SoftDeletes;
15
16     protected $dates = ['deleted_at'];
17
18     protected $guard = 'admin';
19
20     protected $fillable = [
21         'name',
22         'type',
23         'email',
24         'password',
25         'deleted'
26     ];
27
28     protected $appends = ["roles_text"];
29
30     protected $hidden = [
31         'password', 'remember_token',
32     ];
33
34     public function roles(){
35         return $this->hasMany('App\Models\RoleAdmin', 'admin_id', 'id');
36     }
37
38     public function getRolesTextAttribute(){
39         $roles = $this->roles;
40         $roles_array = [];
41         foreach ($roles as $role){
42             $roles_array[] = $role["role_id"];
43         }
44         return implode(",", $roles_array);
45     }
46 }
47
```



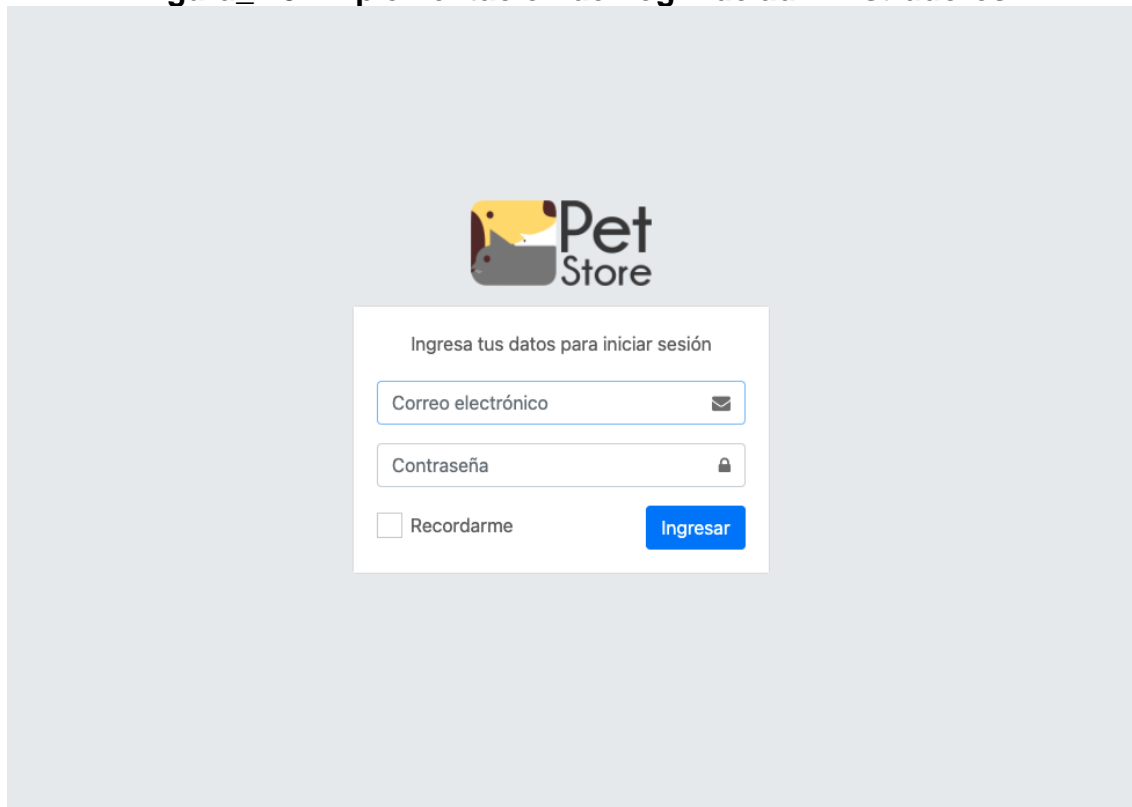
Figura\_ 42: Vista Login de administradores

```
admin-login.blade.php x
resources > views > back-office > auth > admin-login.blade.php > ...
1 @extends('layouts.back-office.login')
2
3 @section('content')
4
5 <div class="card">
6   <div class="card-body login-card-body">
7     <p class="login-box-msg">Ingresa tus datos para iniciar sesión</p>
8     <form method="POST" action="{{ route('admin.process-login') }}">
9       @csrf
10      <div class="input-group mb-3">
11        <input type="email" name="email" class="form-control @error('email') is-invalid @enderror" placeholder="Correo electr
12        <div class="input-group-append">
13          <div class="input-group-text">
14            <span class="fa fa-envelope"></span>
15          </div>
16        </div>
17      </div>
18
19      <div class="input-group mb-3">
20        <input type="password" name="password" class="form-control @error('password') is-invalid @enderror" placeholder="Cont
21        <div class="input-group-append">
22          <div class="input-group-text">
23            <span class="fa fa-lock"></span>
24          </div>
25        </div>
26      </div>
27
28      <div class="row">
29        <div class="col-8">
30          <div class="icheck-primary">
31            <input type="checkbox" id="remember" name="remember" {{ old('remember') ? 'checked' : '' }}>
32            <label for="remember">Recordarme</label>
33          </div>
34        </div>
35        <div class="col-4">
36          <button type="submit" class="btn btn-primary btn-block">Ingresar</button>
37        </div>
38      </div>
39    </form>
40  </div>
41 </div>
42 @endsection
43
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 43: Implementación de Login de administradores**



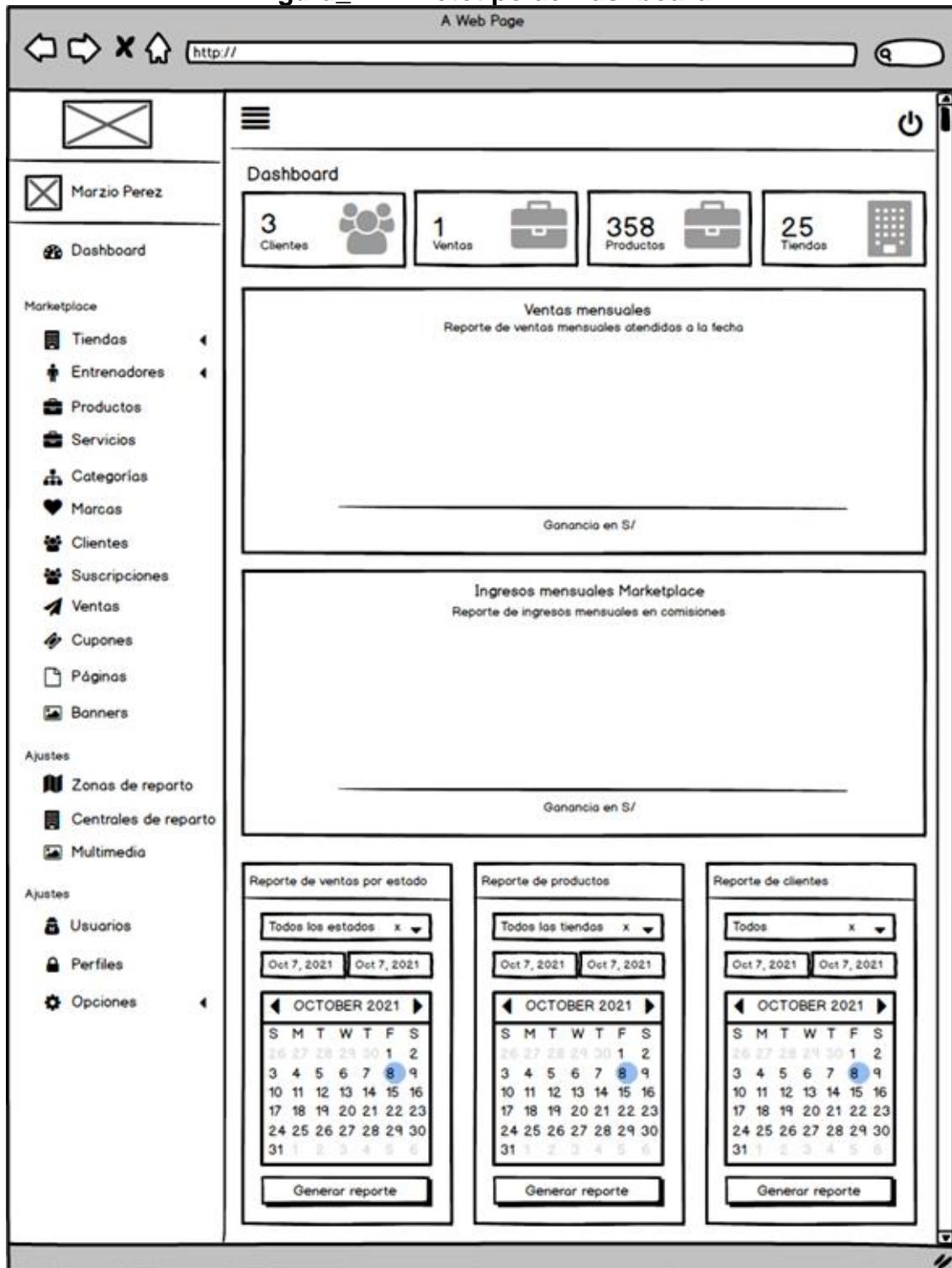
**RF2: El sistema debe permitir la vista de un resumen de reportes para cada usuario, denominado dashboard**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

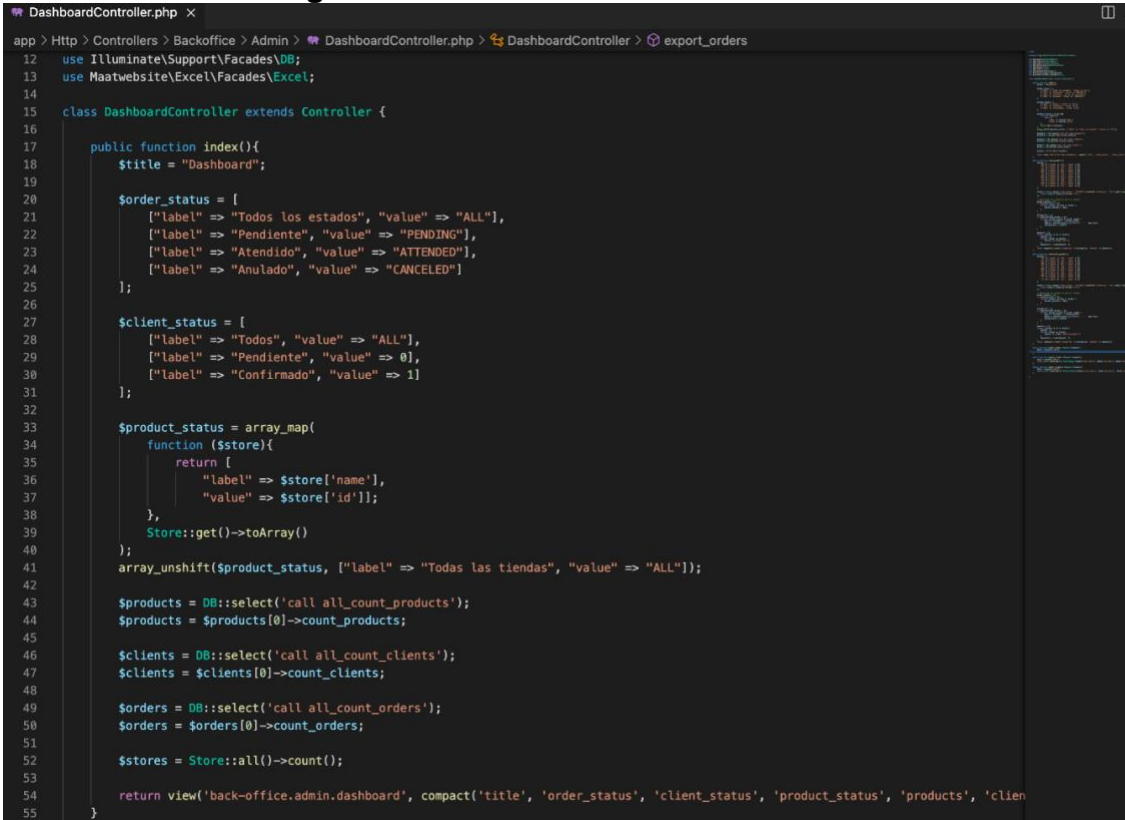
Figura\_ 44: Prototipo de Dashboard



DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

**Figura\_ 45: Controlador de Dashboard**



```
DashboardController.php
app > Http > Controllers > Backoffice > Admin > DashboardController.php > DashboardController > export_orders
12 use Illuminate\Support\Facades\DB;
13 use Maatwebsite\Excel\Facades\Excel;
14
15 class DashboardController extends Controller {
16
17     public function index(){
18         $title = "Dashboard";
19
20         $order_status = [
21             ["label" => "Todos los estados", "value" => "ALL"],
22             ["label" => "Pendiente", "value" => "PENDING"],
23             ["label" => "Atendido", "value" => "ATTENDED"],
24             ["label" => "Anulado", "value" => "CANCELED"]
25         ];
26
27         $client_status = [
28             ["label" => "Todos", "value" => "ALL"],
29             ["label" => "Pendiente", "value" => 0],
30             ["label" => "Confirmado", "value" => 1]
31         ];
32
33         $product_status = array_map(
34             function ($store){
35                 return [
36                     "label" => $store['name'],
37                     "value" => $store['id'];
38                 ],
39             Store::get()->toArray()
40         );
41         array_unshift($product_status, ["label" => "Todas las tiendas", "value" => "ALL"]);
42
43         $products = DB::select('call all_count_products');
44         $products = $products[0]->count_products;
45
46         $clients = DB::select('call all_count_clients');
47         $clients = $clients[0]->count_clients;
48
49         $orders = DB::select('call all_count_orders');
50         $orders = $orders[0]->count_orders;
51
52         $stores = Store::all()->count();
53
54         return view('back-office.admin.dashboard', compact('title', 'order_status', 'client_status', 'product_status', 'products', 'clien
55     }
```

**Figura\_ 46: Modelo de Dashboard**

Figura\_ 47: Vista de Dashboard

```
dashboard.blade.php x
resources > views > back-office > admin > dashboard.blade.php > div.row.mt-3 > div.col-4
1 @extends('layouts.back-office.admin', ['title' => $title])
2
3 @section('content')
4
5 <div class="row">
6 <div class="col-lg-3 col-6">
7 <div class="small-box bg-info">
8 <div class="inner">
9 <h3 class="text-white">{{ $clients }}</h3>
10 <p class="text-white">Clientes</p>
11 </div>
12 <div class="icon">
13 <i class="fa fa-users"></i>
14 </div>
15 </div>
16 </div>
17 <div class="col-lg-3 col-6">
18 <div class="small-box bg-success">
19 <div class="inner">
20 <h3 class="text-white">{{ $orders }}</h3>
21 <p class="text-white">Ventas</p>
22 </div>
23 <div class="icon">
24 <i class="fa fa-briefcase"></i>
25 </div>
26 </div>
27 </div>
28 <div class="col-lg-3 col-6">
29 <div class="small-box bg-warning">
30 <div class="inner">
31 <h3 class="text-white">{{ $products }}</h3>
32 <p class="text-white">Productos</p>
33 </div>
34 <div class="icon">
35 <i class="fa fa-briefcase"></i>
36 </div>
37 </div>
38 </div>
39 <div class="col-lg-3 col-6">
40 <div class="small-box bg-danger">
41 <div class="inner">
42 <h3 class="text-white">{{ $stores }}</h3>
43 <p class="text-white">Tiendas</p>
44 </div>
45 <div class="icon">
46 <i class="fa fa-building"></i>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

Figura\_ 48: Implementación de Dashboard



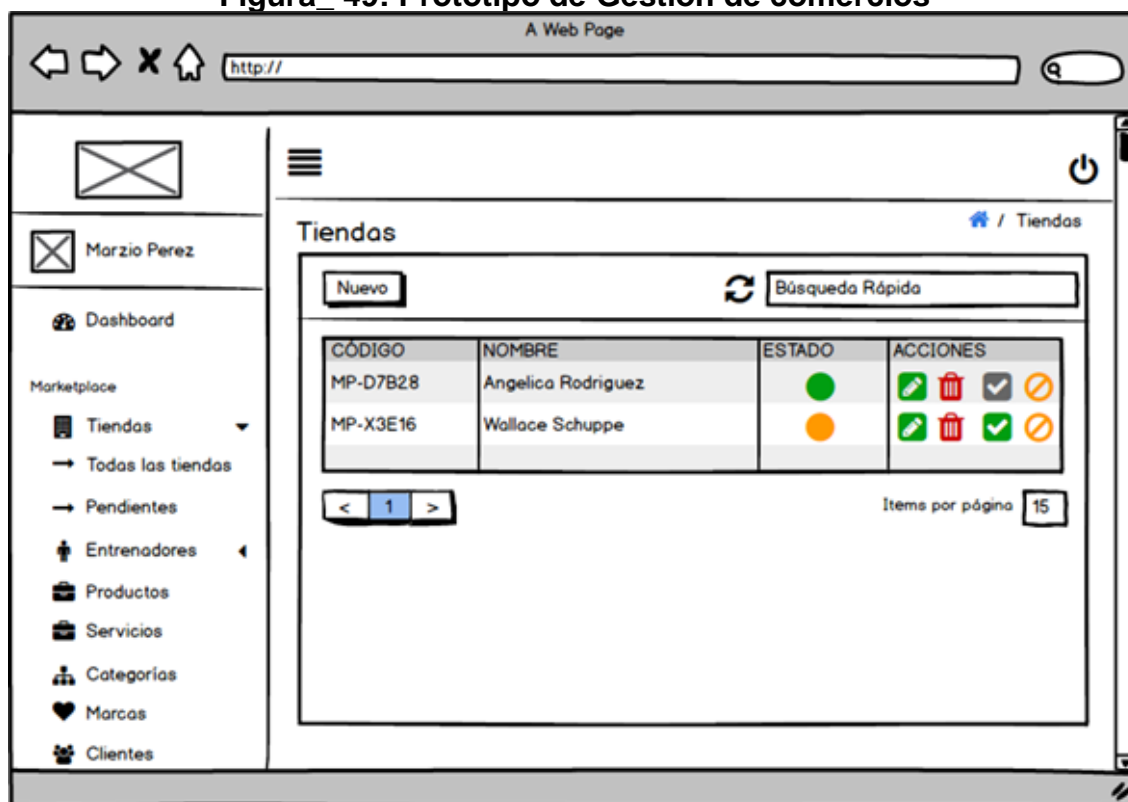
RF3: El sistema debe permitir registrar, modificar, buscar, consultar y eliminar los comercios que se afilian a la plataforma. En donde se deben tener en cuenta los campos: código, nombre, correo, teléfono, celular, dirección, banco, número de cuenta, número de cuenta cci, comisión, y opciones de envío.

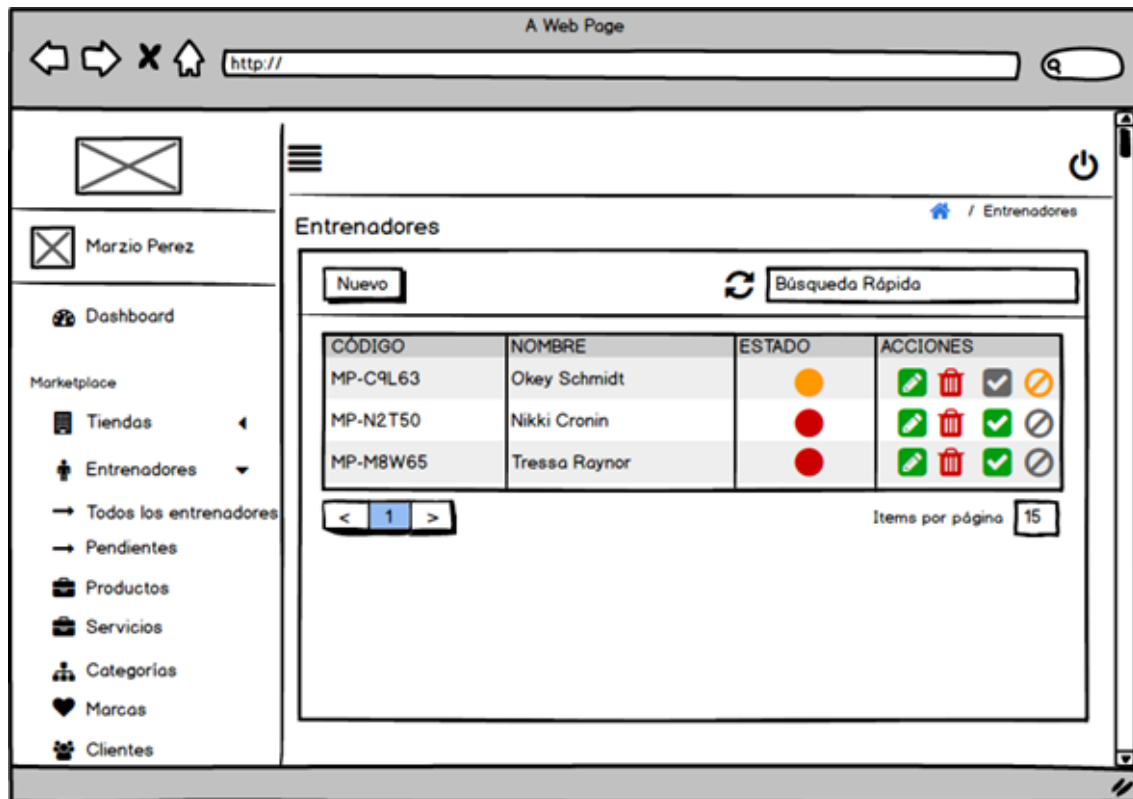
## DISEÑO

### PROTOTIPO

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 49: Prototipo de Gestión de comercios





## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 50: Controlador de Gestión de comercios

```
24
25 class StoreController extends Controller {
26
27     public function index() {
28         $title = "Tiendas";
29         $list_url = route('admin.stores.lists');
30
31         $delivery_types = array_map(
32             function($item){
33                 return [
34                     "id" => $item['id'],
35                     "value" => $item['name']
36                 ];
37             }, DeliveryType::where('status', 'ACTIVE')->get()->toArray()
38         );
39         return view('back-office.admin.store', compact('title', 'list_url', 'delivery_types'));
40     }
41
42     public function pending() {
43         $title = "Tiendas pendientes de confirmación";
44         $list_url = route('admin.stores.pending.lists');
45         $delivery_types = array_map(
46             function($item){
47                 return [
48                     "id" => $item['id'],
49                     "value" => $item['name']
50                 ];
51             }, DeliveryType::where('status', 'ACTIVE')->get()->toArray()
52         );
53         return view('back-office.admin.store', compact('title', 'list_url', 'delivery_types'));
54     }
55
56     public function lists(){
57         $model = Store::with('delivery_types')->where("is_trainer", false)->get();
58         return response()->json($model);
59     }
60
61     public function pending_lists(){
62         $model = Store::where("is_trainer", false)->where('status', "PENDING")->get();
63         return response()->json($model);
64     }
65 }
```



Figura\_ 51: Modelo de Gestión de comercios

```
app > Models > Store.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class Store extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'code',
18         'name',
19         'address',
20         'phone',
21         'cellphone',
22         'email',
23
24         'bank_name',
25         'bank_number',
26         'bank_cci_number',
27         // Porcentaje de comisión
28         'comission_rate',
29
30         'is_trainer',
31         'slug_route_id',
32         // PENDING | APPROVED | DENIED
33         'status'
34     ];
35
36     protected $appends = ["route", "delivery_types_array"];
```

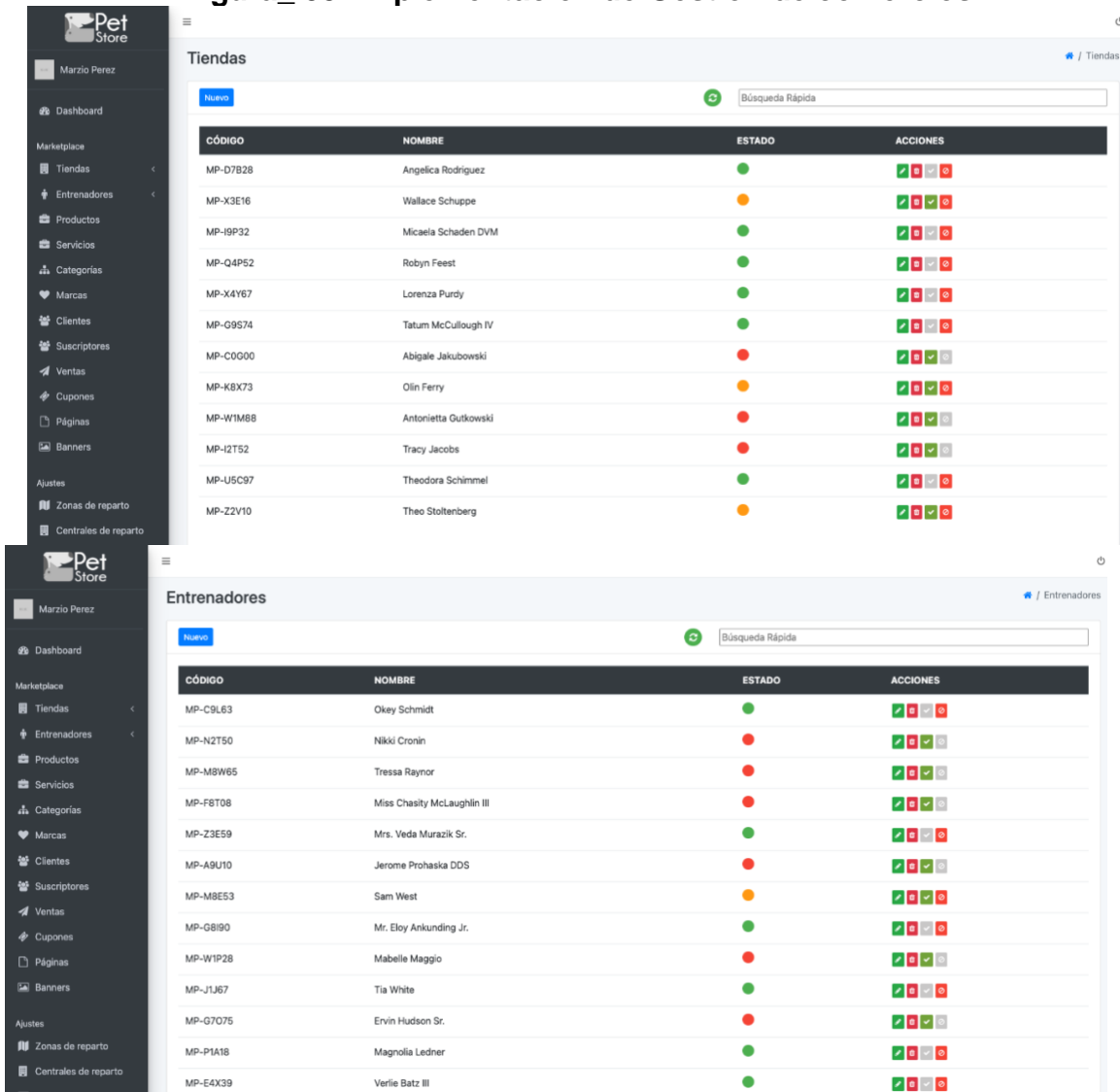
Figura\_ 52: Vista de Gestión de comercios

```
resources > views > back-office > admin > store.blade.php > ...
1 @extends('layouts.back-office.admin', ['title' => $title])
2
3 @section('content')
4
5 @php
6 use App\Utils\ReactCrudField;
7 use App\Utils\ReactCrudSettingsBuilder;
8
9 $builder = new ReactCrudSettingsBuilder();
10
11 $code = new ReactCrudField('code');
12 $code->title('Código')->disabled(true)->width(2);
13 $builder->addField($code);
14
15 $name = new ReactCrudField('name');
16 $name->title('Nombre')->required(true)->width(4);
17 $builder->addField($name);
18
19 $email = new ReactCrudField('email');
20 $email->title('E-mail')->show(false)->required(true)->width(3);
21 $builder->addField($email);
22
23 $phone = new ReactCrudField('phone');
24 $phone->title('Teléfono')->show(false)->required(true)->width(3);
25 $builder->addField($phone);
26
27 $cellphone = new ReactCrudField('cellphone');
28 $cellphone->title('Celular')->show(false)->required(true)->width(3);
29 $builder->addField($cellphone);
30
31 $address = new ReactCrudField('address');
32 $address->title('Dirección')->show(false)->required(true)->width(9);
33 $builder->addField($address);
34
35 $bank_name = new ReactCrudField('bank_name');
36 $bank_name->title('Banco')->show(false)->required(true)->width(2);
37 $builder->addField($bank_name);
38
39 $bank_number = new ReactCrudField('bank_number');
40 $bank_number->title('Número de cuenta')->show(false)->required(true)->width(4);
41 $builder->addField($bank_number);
42
43 $bank_cci_number = new ReactCrudField('bank_cci_number');
44 $bank_cci_number->title('Número de cuenta CCI')->show(false)->required(true)->width(4);
45 $builder->addField($bank_cci_number);
46
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 53: Implementación de Gestión de comercios**



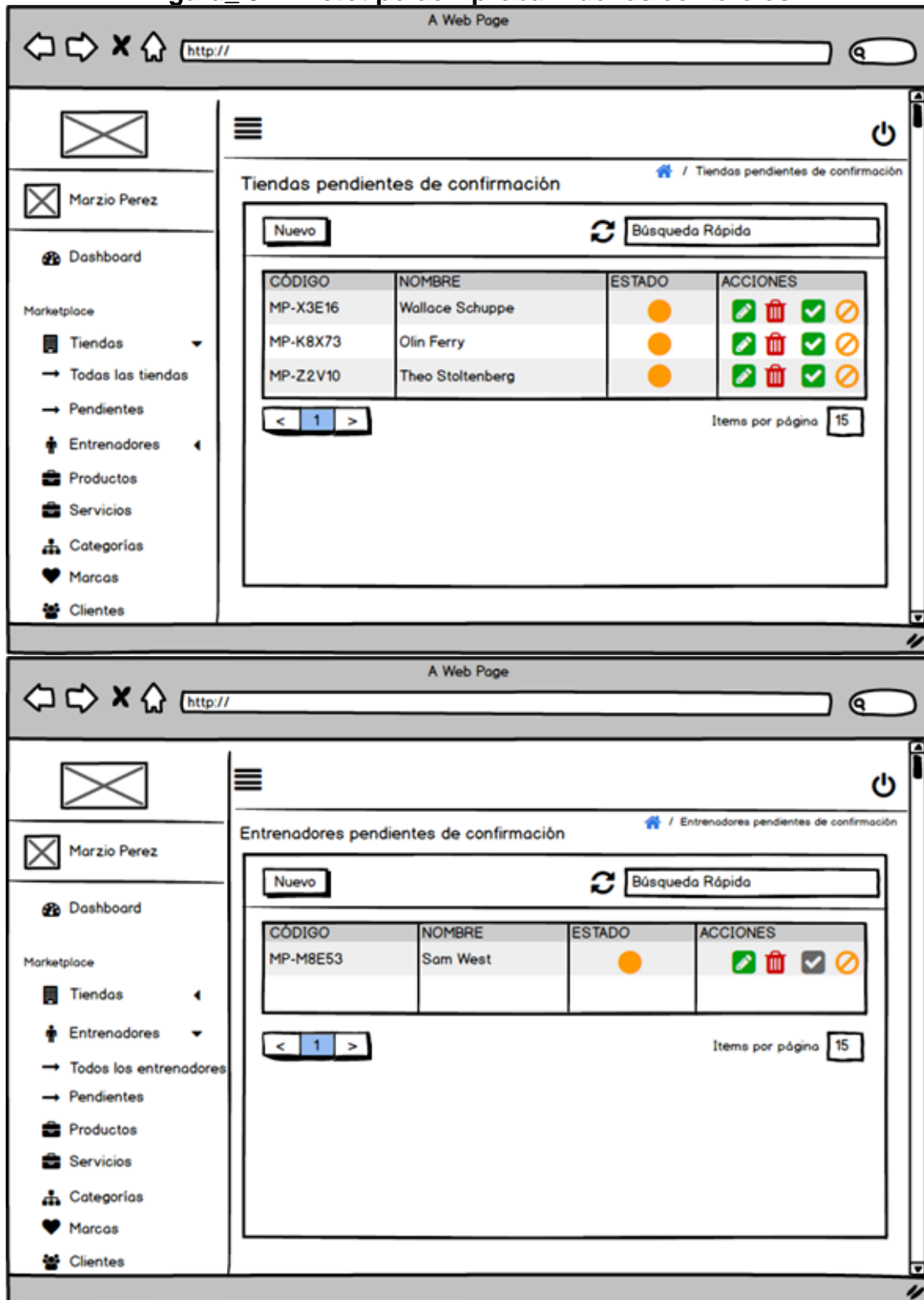
**RF4: El sistema permite la aprobación de nuevos comercios, y debe enviar sus credenciales por correo**

## DISEÑO

### PROTOTIPO

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 54: Prototipo de Aprobar nuevos comercios



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

**Figura\_ 55: Controlador de Aprobar nuevos comercios**

```
StoreController.php x
app > Http > Controllers > Backoffice > Admin > StoreController.php > ...
237 public function confirm(Request $request) {
238     $data = $request->all();
239     $model = Store::find($data['id']);
240
241     if($model["status"] === "PENDING") {
242         $admin = Role::create([
243             'name' => "Administrador",
244             'model' => "STORE",
245             'store_id' => $data["id"],
246             'deleted' => false
247         ]);
248
249         $modules = Module::with('menus')->where('store_type', "STORE")->orWhereNull('store_type')->where('store_id', $data["id"]);
250         if (count($modules) > 0) {
251             foreach ($modules as $module) {
252                 $role_admin_module = RoleModule::create([
253                     'role_id' => $admin->id,
254                     'module_id' => $module["id"]
255                 ]);
256
257                 if (count($module["menus"]) > 0){
258                     foreach ($module["menus"] as $menu){
259                         RoleModuleMenu::create([
260                             'module_menu_id' => $menu["id"],
261                             'role_module_id' => $role_admin_module->id
262                         ]);
263                     }
264                 }
265             }
266         }
267
268         if($admin) {
269             $seller = Seller::create([
270                 'name' => "Vendedor",
271                 'email' => $model["email"],
272                 'password' => bcrypt($model["code"]),
273                 'deleted' => false,
274                 'store_id' => $model->id
275             ]);
276
277             RoleSeller::create([
278                 'role_id' => $admin->id,
279                 'seller_id' => $seller->id
280             ]);

```

Figura\_ 56: Modelo de Aprobar nuevos comercios

```
app > Models > Store.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class Store extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'code',
18         'name',
19         'address',
20         'phone',
21         'cellphone',
22         'email',
23
24         'bank_name',
25         'bank_number',
26         'bank_cci_number',
27         // Porcentaje de comisión
28         'comission_rate',
29
30         'is_trainer',
31         'slug_route_id',
32         // PENDING | APPROVED | DENIED
33         'status'
34     ];
35
36     protected $appends = ["route", "delivery_types_array"];
```

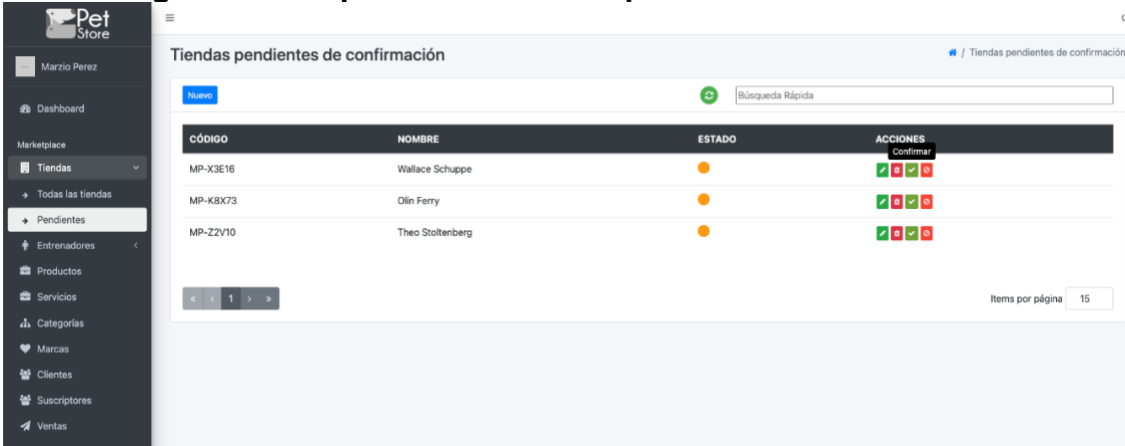
Figura\_ 57: Vista de Aprobar nuevos comercios

```
resources > views > back-office > admin > store.blade.php > ...
1 @extends('layouts.back-office.admin', ['title' => $title])
2
3 @section('content')
4
5 @php
6 use App\Utils\ReactCrudField;
7 use App\Utils\ReactCrudSettingsBuilder;
8
9 $builder = new ReactCrudSettingsBuilder();
10
11 $code = new ReactCrudField('code');
12 $code->title('Código')->disabled(true)->width(2);
13 $builder->addField($code);
14
15 $name = new ReactCrudField('name');
16 $name->title('Nombre')->required(true)->width(4);
17 $builder->addField($name);
18
19 $email = new ReactCrudField('email');
20 $email->title('E-mail')->show(false)->required(true)->width(3);
21 $builder->addField($email);
22
23 $phone = new ReactCrudField('phone');
24 $phone->title('Teléfono')->show(false)->required(true)->width(3);
25 $builder->addField($phone);
26
27 $cellphone = new ReactCrudField('cellphone');
28 $cellphone->title('Celular')->show(false)->required(true)->width(3);
29 $builder->addField($cellphone);
30
31 $address = new ReactCrudField('address');
32 $address->title('Dirección')->show(false)->required(true)->width(9);
33 $builder->addField($address);
34
35 $bank_name = new ReactCrudField('bank_name');
36 $bank_name->title('Banco')->show(false)->required(true)->width(2);
37 $builder->addField($bank_name);
38
39 $bank_number = new ReactCrudField('bank_number');
40 $bank_number->title('Número de cuenta')->show(false)->required(true)->width(4);
41 $builder->addField($bank_number);
42
43 $bank_cci_number = new ReactCrudField('bank_cci_number');
44 $bank_cci_number->title('Número de cuenta CCI')->show(false)->required(true)->width(4);
45 $builder->addField($bank_cci_number);
46
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 58: Implementación de Aprobar nuevos comercios**



The screenshot shows a web application interface for 'Pet Store'. The user is logged in as 'Marzio Perez'. The main content area is titled 'Tiendas pendientes de confirmación' (Stores pending confirmation). It features a search bar with the text 'Búsqueda Rápida' and a 'Nuevo' button. Below the search bar is a table with the following data:

CÓDIGO	NOMBRE	ESTADO	ACCIONES
MP-X3E16	Wallace Schuppe	●	Confirmar
MP-KBX73	Olin Ferry	●	Confirmar
MP-ZZV10	Theo Stoltenberg	●	Confirmar

At the bottom of the table, there are navigation arrows and a pagination control showing 'Items por página 15'.

**RF5: El sistema debe permitir registrar, modificar, buscar, consultar y eliminar las categorías que servirán para el registro de productos, estos además pueden tener una estructura de padre e hijo.**

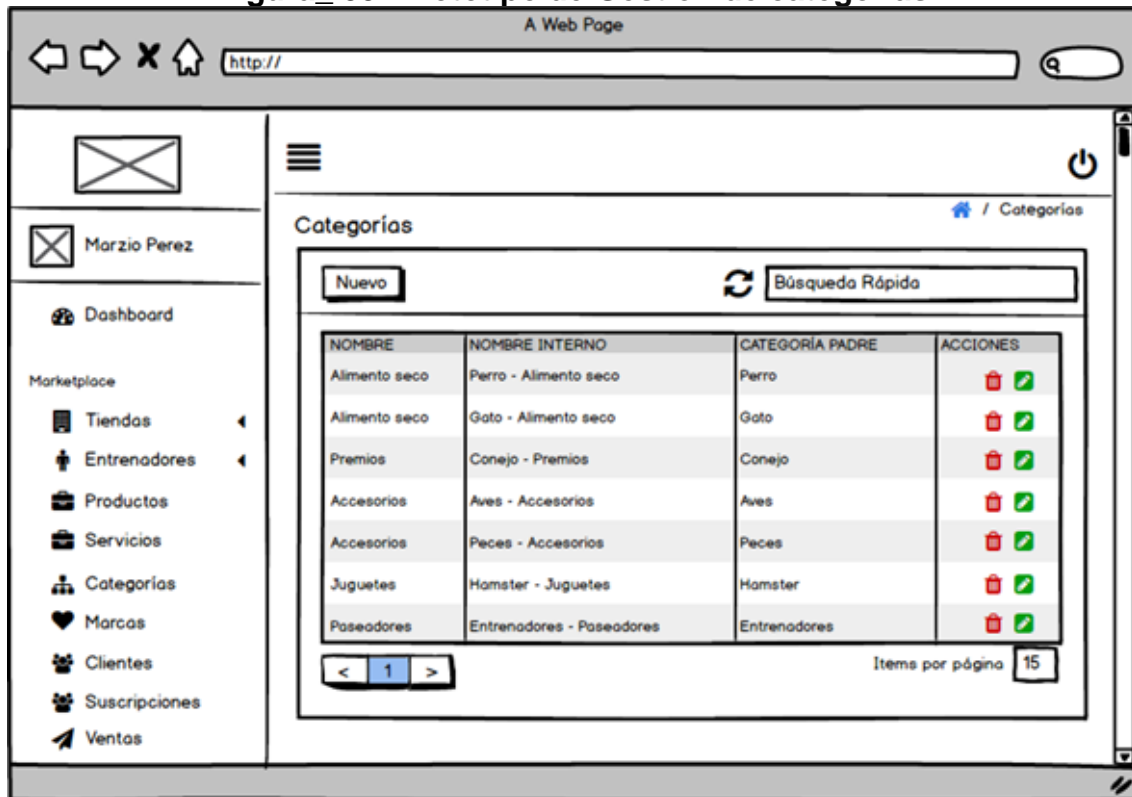
## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.



Figura\_ 59: Prototipo de Gestión de categorías



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura 60: Controlador de Gestión de categorías

```
app > Http > Controllers > Backoffice > Admin > CategoryController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Backoffice\Admin;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Category;
7  use App\Utils\SlugUtil;
8  use Illuminate\Http\Request;
9
10 class CategoryController extends Controller {
11
12     public function index() {
13         $title = "Categorías";
14         $categories = array_map(
15             function($client){
16                 return [
17                     "id" => $client["id"],
18                     "value" => $client["name"]
19                 ];
20             },
21             Category::where("type", "products")->get()->toArray()
22         );
23         return view('back-office.admin.category', compact('title', 'categories'));
24     }
25
26     public function lists(){
27         $model = Category::where("type", "products")->get();
28         return response()->json($model);
29     }
30
31     public function create(Request $request){
32         $data = $request->all();
33         $parent_slug_id = null;
34         if($data["category_id"] != ""){
35             $data["is_child"] = true;
36             $parent_category = Category::find($data["category_id"]);
37             $parent_slug_id = $parent_category["slug_route_id"];
38         }
39
40         $slug_util = new SlugUtil();
41         $slug = $slug_util->createSlug($data["name"], $parent_slug_id);
42
43         $data["slug_route_id"] = $slug->id;
44         $data["type"] = "products";
45         $model = Category::create($data);
46         $slug_util->setModel($slug->id, 'category', $model->id);
47         $model = Category::find($model->id);

```

Figura\_ 61: Modelo de Gestión de categorías

```
Category.php x
app > Models > Category.php > Category
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class Category extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'internal_name',
20         'is_child',
21         // articles - products
22         "type",
23         'category_id',
24         'featured_image_id',
25         "slug_route_id"
26     ];
27
28     protected $appends = ["image_url", "route"];
29
30     public function getRouteAttribute(){
31         if($this->slug){
32             if ($this->attributes["slug_route_id"]) {
33                 $explode_slug = explode("/", $this->slug->slug);
34                 if(count($explode_slug) === 1) {
35                     return route('page.route', ["base" => $explode_slug[0]]);
36                 }
37
38                 if(count($explode_slug) === 2) {
39                     return route('page.route', ["base" => $explode_slug[0], "child" => $explode_slug[1] ]);
40                 }
41             }
42         }
43         return null;
44     }
45 }
```

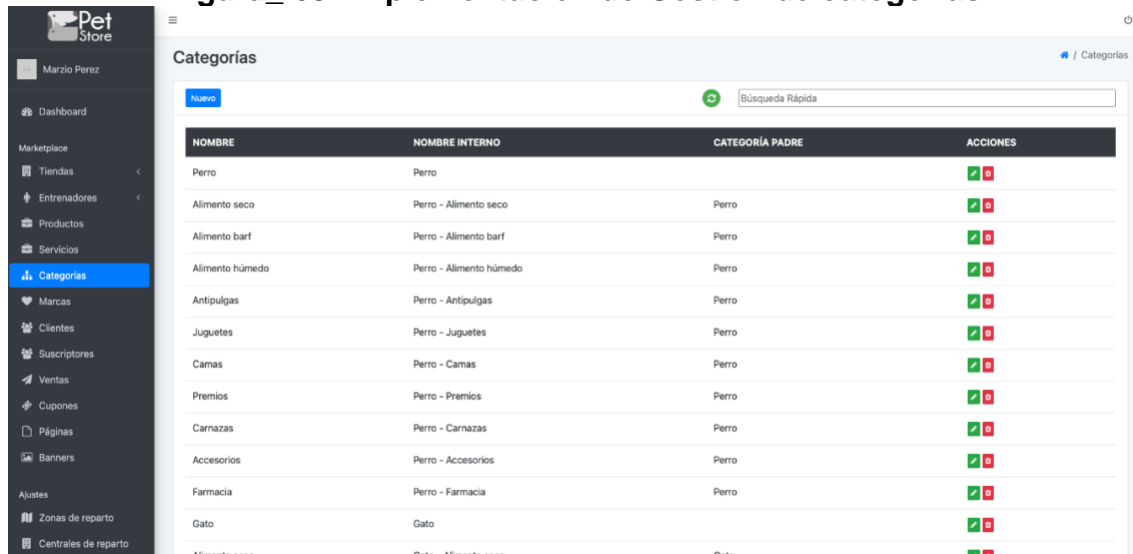
Figura\_ 62: Vista de Gestión de categorías

```
category.blade.php x
resources > views > back-office > admin > category.blade.php > ...
1 @extends('layouts.back-office.admin', ["title" => $title])
2
3 @section('content')
4
5     @php
6         use App\Utils\ReactCrudField;
7         use App\Utils\ReactCrudSettingsBuilder;
8
9         $builder = new ReactCrudSettingsBuilder();
10
11         $nameField = new ReactCrudField('name');
12         $nameField->title('Nombre')->required(true)->width(4);
13         $builder->addField($nameField);
14
15         $internal_name = new ReactCrudField('internal_name');
16         $internal_name->title('Nombre interno')->required(true)->width(4);
17         $builder->addField($internal_name);
18
19         $category_idField = new \App\Utils\ReactCrudField('category_id');
20         $category_idField->title('Categoría padre')->show(true)->type('map', $categories)->width(4);
21         $builder->addField($category_idField);
22
23         $featured_image_idField = new ReactCrudField('featured_image_id');
24         $featured_image_idField->title('Imagen destacada')->show(false)->type('featured_image');
25         $builder->addField($featured_image_idField);
26
27         $actions = [];
28         $actions["custom"] = [];
29         $actions['create-custom'] = [
30             'url' => route('admin.categories.create'),
31             'tabs' => [
32                 [
33                     "name" => "Imagen destacada",
34                     "type" => 'featured_image',
35                     'multimedia-lists-url' => route('admin.multimedia.lists')
36                 ],
37                 [
38                     "name" => "Cargar imágenes",
39                     "type" => 'upload_resources',
40                     'multimedia-upload-url' => route('admin.multimedia.upload')
41                 ],
42                 [
43                     "name" => "Meta datos",
44                     "type" => 'media_tags'
45                 ]
46             ]
47         ]
48     
```

























## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 63: Implementación de Gestión de categorías**



The screenshot displays a web application interface for 'Pet Store'. On the left is a dark sidebar menu with options like 'Dashboard', 'Tiendas', 'Productos', 'Servicios', 'Categorías' (highlighted), 'Marcas', 'Clientes', 'Suscriptores', 'Ventas', 'Cupones', 'Páginas', 'Banners', 'Ajustes', 'Zonas de reparto', and 'Centrales de reparto'. The main content area is titled 'Categorías' and features a search bar labeled 'Búsqueda Rápida'. Below the search bar is a table with the following data:

NOMBRE	NOMBRE INTERNO	CATEGORÍA PADRE	ACCIONES
Perro	Perro		 
Alimento seco	Perro - Alimento seco	Perro	 
Alimento barf	Perro - Alimento barf	Perro	 
Alimento húmedo	Perro - Alimento húmedo	Perro	 
Antipulgas	Perro - Antipulgas	Perro	 
Juguetes	Perro - Juguetes	Perro	 
Camas	Perro - Camas	Perro	 
Premios	Perro - Premios	Perro	 
Carnazas	Perro - Carnazas	Perro	 
Accesorios	Perro - Accesorios	Perro	 
Farmacia	Perro - Farmacia	Perro	 
Gato	Gato		 

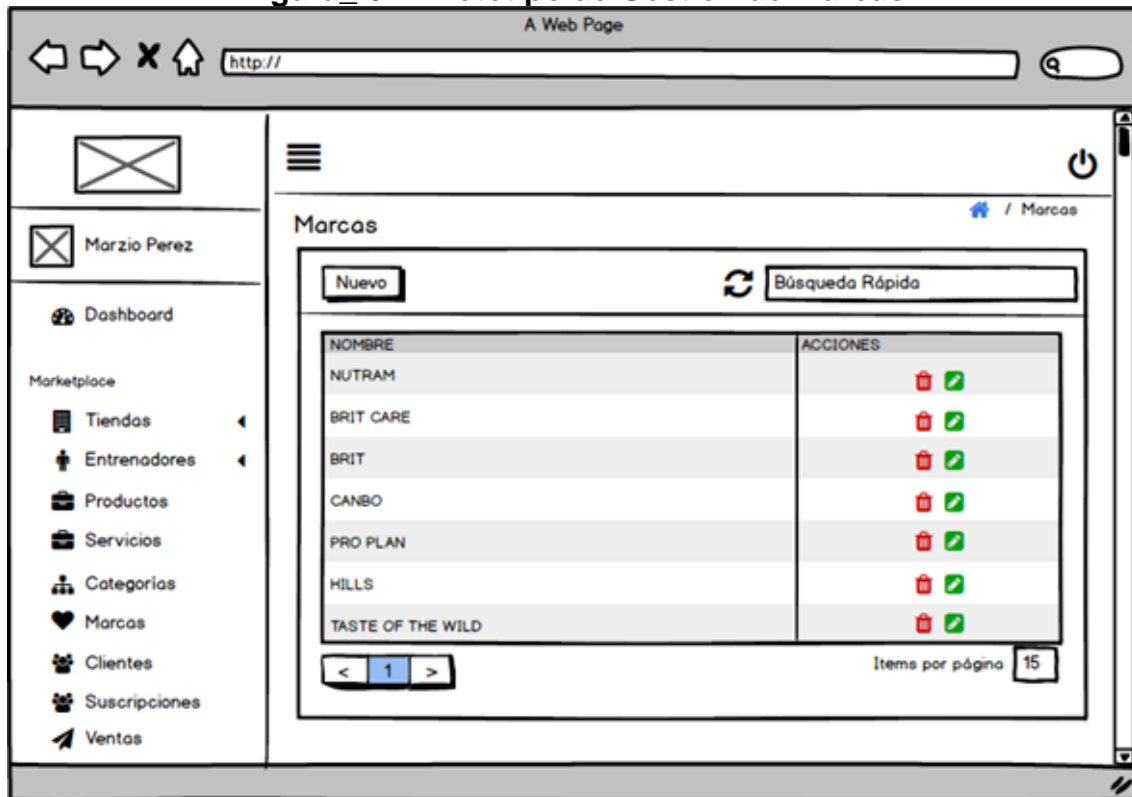
**RF6: El sistema debe permitir registrar, modificar, buscar, consultar y eliminar las marcas para el registro posterior de los productos**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 64: Prototipo de Gestión de marcas



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 65: Controlador de Gestión de marcas

```
BrandController.php x
app > Http > Controllers > Backoffice > Admin > BrandController.php > ...
1 <?php
2
3 namespace App\Http\Controllers\Backoffice\Admin;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Brand;
7 use App\Utils\SlugUtil;
8 use Exception;
9 use Illuminate\Http\Request;
10
11 class BrandController extends Controller {
12
13     public function index() {
14         $title = "Marcas";
15         return view('back-office.admin.brand', compact('title'));
16     }
17
18     public function lists(){
19         $model = Brand::get();
20         return response()->json($model);
21     }
22
23     public function create(Request $request){
24         try{
25             $data = $request->all();
26             $slug_util = new SlugUtil();
27             $slug = $slug_util->createSlug($data["name"]);
28             $data["slug_route_id"] = $slug->id;
29             $model = Brand::create($data);
30             $slug_util->setModel($slug->id, 'brand', $model->id);
31             $model = Brand::find($model->id);
32         }catch(Exception $e) {
33             return response()->json(['status'=>'error', 'message' => "No se pudo actualizar el registro."]);
34         }
35         return response()->json(['status'=>"ok", 'data' => $model]);
36     }
37
38     public function update(Request $request) {
39         $data = $request->all();
40         $brand = Brand::find($data['id']);
41         $brand->update($data);
42         $model = Brand::find($data['id']);
43         return response()->json(["status" => "ok", "data" => $model]);
44     }
45
46     public function delete(Request $request){
47         $data = $request->all();
48     }
49 }
```

Ln. 1, col. 1 Espacios: 4 UTF-8 LF PHP

Figura\_ 66: Modelo de Gestión de marcas

```
Brand.php ×
app > Models > Brand.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class Brand extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'featured_image_id',
20         'slug_route_id'
21     ];
22
23     protected $appends = ["image_url", "route"];
24
25     public function getRouteAttribute(){
26         if($this->slug){
27             if ($this->attributes["slug_route_id"]) {
28                 $explode_slug = explode("/", $this->slug->slug);
29                 if(count($explode_slug) > 0) {
30                     return route('page.route', ["base" => $explode_slug[0]]);
31                 }
32             }
33         }
34         return null;
35     }
36
37     public function getImageUrlAttribute(){
38         $featured_image = $this->attributes["featured_image_id"];
39         if($featured_image){
40             return $this->featured_image["file_url"];
41         }
42         return "https://placeholder.jp/400x200.png";
43     }
44 }
```



Figura\_ 67: Vista de Gestión de marcas

```
brand.blade.php x
resources > views > back-office > admin > brand.blade.php > ...
1 @extends('layouts.back-office.admin', ['title' => $title])
2
3 @section('content')
4
5 @php
6     use App\Utils\ReactCrudField;
7     use App\Utils\ReactCrudSettingsBuilder;
8
9     $builder = new ReactCrudSettingsBuilder();
10
11     $nameField = new ReactCrudField('name');
12     $nameField->title('Nombre')->required(true)->width(12);
13     $builder->addField($nameField);
14
15     $featured_image_idField = new ReactCrudField('featured_image_id');
16     $featured_image_idField->title('Imagen destacada')->show(false)->type('featured_image');
17     $builder->addField($featured_image_idField);
18
19     $actions = [];
20     $actions["custom"] = [];
21     $actions['create-custom'] = [
22         'url' => route('admin.brands.create'),
23         'tabs' => [
24             [
25                 "name" => "Imagen destacada",
26                 "type" => 'featured_image',
27                 "field" => 'featured_image_id',
28                 'multimedia-lists-url' => route('admin.multimedia.lists')
29             ],
30             [
31                 "name" => "Cargar imágenes",
32                 "type" => 'upload_resources',
33                 'multimedia-upload-url' => route('admin.multimedia.upload')
34             ]
35         ]
36     ];
37     $actions['update-custom'] = [
38         'url' => route('admin.brands.update'),
39         'tabs' => [
40             [
41                 "name" => "Imagen destacada",
42                 "type" => 'featured_image',
43                 'multimedia-lists-url' => route('admin.multimedia.lists')
44             ],
45             [
46                 "name" => "Cargar imágenes",
47                 "type" => 'upload_resources'
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el product owner para que el equipo de trabajo pueda desarrollarlo.

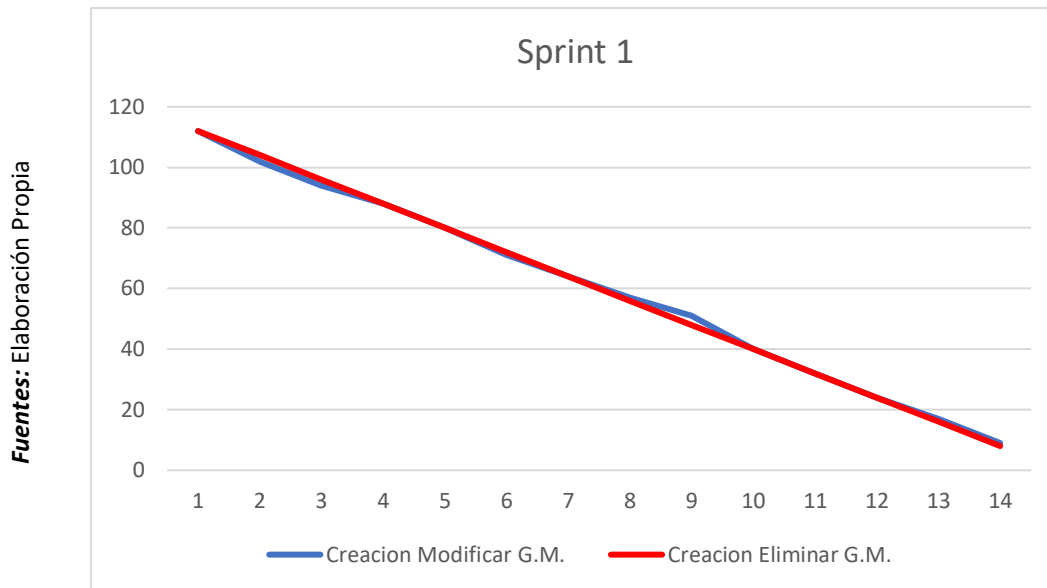
Figura\_ 68: Implementación de Gestión de marcas

The screenshot shows a web application interface for 'Pet Store'. On the left is a dark sidebar menu with options like Dashboard, Tiendas, Entrenadores, Productos, Servicios, Categorías, Marcas (highlighted), Clientes, Suscriptores, Ventas, Cupones, Páginas, Banners, Ajustes, Zonas de reparto, and Centrales de reparto. The main content area is titled 'Marcas' and features a search bar with a 'Nuevo' button and a 'Búsqueda Rápida' input. Below the search bar is a table with two columns: 'NOMBRE' and 'ACCIONES'. The table contains 14 rows of brand names, each with a green checkmark and a red square icon in the 'ACCIONES' column.

NOMBRE	ACCIONES
NUTRAM	✓ □
BRIT CARE	✓ □
BRIT	✓ □
CANBO	✓ □
PRO PLAN	✓ □
HILLS	✓ □
TASTE OF THE WILD	✓ □
PRO PAC	✓ □
EARTH BORN	✓ □
PURINA	✓ □
ROYAL CANIN	✓ □
CARNILOVE	✓ □
VIRBAC	✓ □

## BURNDOWN DEL SPRINT N° 1

Figura\_ 69: Burndown del Sprint 1



### Burndown Sprint 1

En la figura vista, se muestra el proceso de desarrollo de cada sprint la cual está representada a través de un gráfico, como podemos observar tenemos dos líneas la cual la línea roja representa al tiempo ideal que se define para el desarrollo del sprint, de igual manera la línea azul representa el tiempo real que tomo el desarrollo del sprint. Ya teniendo bien definido las representaciones tenemos que concluir que si la línea azul se encuentra por debajo de la roja significa que se presentó un adelanto, de lo contrario si la línea azul está por encima es decir que hubo un retraso.

## ACTA DE REUNION DEL SPRINT N° 1

Siendo las 3 pm del día 21 de mayo del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.


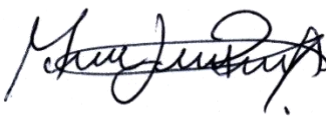


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El encargado de la Empresa AC Brands & Consulting E.I.R.L., Isaac Abensur Vargas concluye la reunión con los Sres. Marzio Perez y Nicolás Segovia aclarando los últimos puntos sobre el sprint.

Para determinar las formas de trabajo y los requerimientos respectivos para esta interacción, también la fecha de entrega y firmar el acta para dar por terminado la reunión.

Los participantes manifestaron su aprobación respecto a todos los puntos definidos en la reunión, que fueron de ayuda a la planificación del Sprint, manifestando su compromiso para la entrega de los resultados en las fechas pactadas.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## ACTA DE ENTREGA DEL SPRINT N° 1


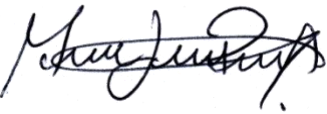


Siendo la 1 pm del día 5 de junio del 2021 se reúne en la empresa AC Brands & Consulting E.I.R.L.

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

Los Sres. Marzio Perez y Nicolás Segovia, especifican cada uno de los requerimientos desarrollados, también presentan las interfaces elaboradas y brindadas por el producto Owner.

Luego de ser verificadas las presentaciones realizadas por los Srs. Marzio Perez y Nicolás Segovia, se decidió de manera unánime, dar por aprobado el término del Sprint respectivo, del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Los presentes manifestarán su aprobación al informe presentado por los Sres. Marzio Perez y Nicolás Segovia del Sprint N° 1 terminado del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 1

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	AC Brands & Consulting E.I.R.L.
<b>Proyecto</b>	“Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Información de la reunión:

<b>Lugar</b>	AC Brands & Consulting E.I.R.L.
<b>Fecha</b>	05/06/2021
<b>Número de iteración / Sprint</b>	Sprint 1
<b>Personas Convocadas a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza
<b>Persona que asistieron a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
<p>Se consiguió ejecutar correctamente el inicio de sesión y las validaciones de los privilegios.</p> <p>Se desarrolló correctamente el tablero de reportes</p> <p>Se desarrolló exitosamente la gestión de los comercios y la aprobación de los nuevos.</p> <p>Se logró desarrollar la gestión de categorías.</p> <p>Se desarrolló la gestión de las marcas.</p>	<p>Se reveló una falla en el momento de realizar las validaciones de los privilegios, pero se solucionó.</p> <p>Se pudo observar fallas en los registros, pero se pudo solucionar.</p>

## PLANIFICACIÓN DEL SPRINT N°2

Siendo las 4 pm del día 5 de junio del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.

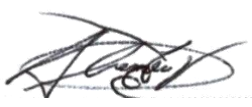
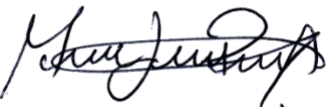


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El gerente de la Empresa AC Brands & Consulting E.I.R.L., efectúa la presentación de los requerimientos, de igual manera manifiesta los requerimientos que tienen mayor prioridad.

Realizado el análisis de los requerimientos presentados por el gerente de AC Brands & Consulting E.I.R.L., el señor Juan Galarza aclara dudas existentes y luego se compromete al cumplimiento de todos los requerimientos propuestas en el Sprint 2.

Los participantes de la reunión manifiestan su aprobación con respecto a la presentación del Sprint 2, así mismo especificando que la fecha de entrega será el día 25 de junio del 2021.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## EJECUCIÓN DEL SPRINT 2

Lo primero que se debe realizar antes del diseño, es conocer y comprender el funcionamiento del sistema, en otras palabras, analizar lo que realmente se necesita, lo cual será elaborado según las historias de usuario.

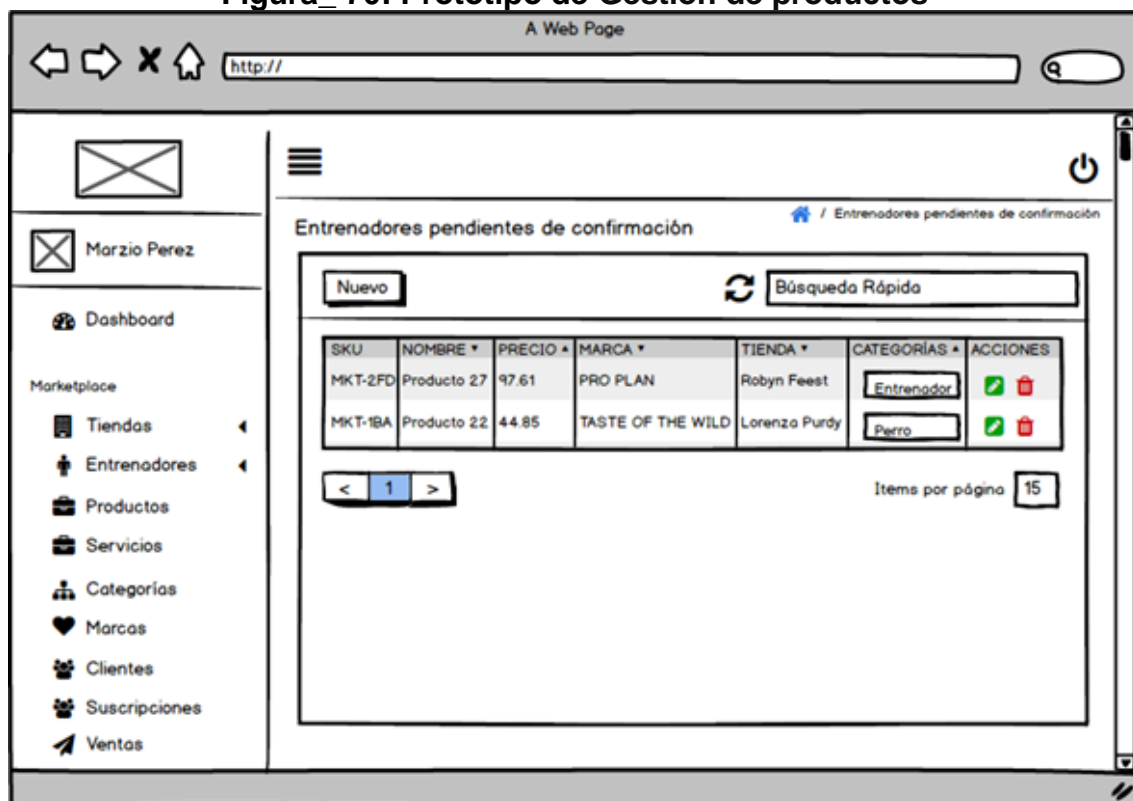
**RF7: El sistema debe permitir registrar, modificar, buscar, consultar y eliminar los productos que se ofrecen en la plataforma. Con los campos: Sku, nombre, precio, precio de oferta, marca, tienda, producto destacado, categorías, descripción, foto, imagen destacada, presentaciones y características.**

## DISEÑO

### PROTOTIPO

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 70: Prototipo de Gestión de productos**



## DESARROLLO



A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

**Figura\_ 71: Controlador de Gestión de productos**

```
ProductController.php M X
app > Http > Controllers > Backoffice > Admin > ProductController.php > ...
15 use App\Utils\SlugUtil;
16 use Exception;
17 use Illuminate\Http\Request;
18 use Illuminate\Support\Facades\DB;
19
20 class ProductController extends Controller {
21
22     public function index() {
23         $title = "Productos";
24
25         $categories = array_map(
26             function($item){
27                 return [
28                     "id" => $item['id'],
29                     "value" => $item['internal_name']
30                 ];
31             }, Category::where('type', 'products')->get()->toArray()
32         );
33
34         $brands = array_map(
35             function($item){
36                 return [
37                     "id" => $item['id'],
38                     "value" => $item['name']
39                 ];
40             }, Brand::all()->toArray()
41         );
42
43         $stores = array_map(
44             function($item){
45                 return [
46                     "id" => $item['id'],
47                     "value" => $item['name']
48                 ];
49             }, Store::where("is_trainer", false)->get()->toArray()
50         );
51         return view('back-office.admin.products.index', compact('title', 'categories', 'brands', 'stores'));
52     }
53
54     public function lists(){
55         $model = DB::select('call all_products');
56         return response()->json($model);
57     }
58
59     public function variations(Request $request) {
60         $data = $request->all();
61         $products = Product::whereIn('product_id', $data['product_id'])->get();
62     }
63 }
```

Figura\_ 72: Modelo de Gestión de productos

```
Product.php X
app > Models > Product.php > Product > $fillable
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Product extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'sku_code',
20         'description',
21         'summary',
22         'featured_image_id',
23
24         'categories',
25
26         'is_featured',
27
28         'price',
29         'iva',
30         'price_with_iva',
31
32         'offer_price',
33         'offer_iva',
34         'offer_price_with_iva',
35
36         'is_service',
37         'service_attentions_per_hour',
38         'rate',
39
40         'brand_id',
41         'brand_name',
42
43         'slug_route_id',
44         'url',
45
46         'store_id',
```

Figura\_ 73: Vista de Gestión de productos

```
index.blade.php
~/Desarrollo/Proyectos/marketplace-pets/
resources/views/back-office/admin/products/
index.blade.php
@extends('back-office.admin', ["title" => $title])
2
3 @section('content')
4
5 @php
6     use App\Utils\ReactCrudField;
7     use App\Utils\ReactCrudSettingsBuilder;
8
9     $builder = new ReactCrudSettingsBuilder();
10
11     $sku_code = new ReactCrudField('sku_code');
12     $sku_code->title('SKU')->required(true)->width(2);
13     $builder->addField($sku_code);
14
15     $name = new ReactCrudField('name');
16     $name->title('Nombre')->required(true)->width(10);
17     $builder->addField($name);
18
19     $price = new ReactCrudField('price');
20     $price->title('Precio')->required(true)->width(2);
21     $builder->addField($price);
22
23     $offer_price = new ReactCrudField('offer_price');
24     $offer_price->title('Precio de oferta')->show(false)->width(2);
25     $builder->addField($offer_price);
26
27     $brand_id = new ReactCrudField('brand_id');
28     $brand_id->title('Marca')->type('map', $brands)->width(3);
29     $builder->addField($brand_id);
30
31     $store_id = new ReactCrudField('store_id');
32     $store_id->title('Tienda')->type('map', $stores)->required(true)->width(3);
33     $builder->addField($store_id);
34
35     $is_featured = new ReactCrudField('is_featured');
36     $is_featured->title('Producto destacado')->type('map', [
37         ['id' => 1, 'value' => 'Si'],
38         ['id' => 0, 'value' => 'No']
39     ]->default(0)->show(false)->width(2);
40     $builder->addField($is_featured);
41
42     $categories_field = new ReactCrudField('categories');
43     $categories_field->title('Categorías')->type('json', $categories)->width(12);
44     $builder->addField($categories_field);
45
46     $description = new \App\Utils\ReactCrudField('description');
47     $description->title('Descripción')->type('editor')->show(false)->width(12);
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 74: Implementación de Gestión de productos**

The screenshot shows a web application interface for 'Pet Store' with a sidebar menu and a main content area titled 'Productos'. The sidebar includes options like Dashboard, Tiendas, Entrenadores, Productos (highlighted), Servicios, Categorías, Marcas, Clientes, Suscriptores, Ventas, Cupones, Páginas, Banners, and Ajustes. The main area contains a table with 15 product entries. Each row includes a SKU, a name (e.g., 'Producto 1'), a price, a brand (e.g., 'PURINA'), a store name ('Angelica Rodriguez'), a category (e.g., 'Hemorre'), and action icons (a red 'X' and a green checkmark).

SKU	NOMBRE	PRECIO	MARCA	TIENDA	CATEGORÍAS	ACCIONES
MKT-9PL	Producto 1	180.24	PURINA	Angelica Rodriguez	Hemorre	<span>X</span> <span>✓</span>
MKT-7DA	Producto 4	62.76	EARTH BORN	Angelica Rodriguez	Parrs	<span>X</span> <span>✓</span>
MKT-3SJ	Producto 5	194.01	CANBO	Angelica Rodriguez	Conato	<span>X</span> <span>✓</span>
MKT-7KD	Producto 6	21.45	EARTH BORN	Angelica Rodriguez	Aves	<span>X</span> <span>✓</span>
MKT-0DD	Producto 7	146.74	BRIT CARE	Angelica Rodriguez	Conato	<span>X</span> <span>✓</span>
MKT-5LW	Producto 8	179.81	NUTRAM	Angelica Rodriguez	Entrenadores	<span>X</span> <span>✓</span>
MKT-8TV	Producto 9	139.15	BRIT CARE	Angelica Rodriguez	Conato	<span>X</span> <span>✓</span>
MKT-3FX	Producto 10	94.27	CARNILOVE	Angelica Rodriguez	Conato	<span>X</span> <span>✓</span>
MKT-4FN	Producto 11	118.2	CARNILOVE	Angelica Rodriguez	Hemorre	<span>X</span> <span>✓</span>
MKT-7HX	Producto 12	99.88	PRO PAC	Angelica Rodriguez	Aves	<span>X</span> <span>✓</span>
MKT-6AG	Producto 13	188.1	CANBO	Angelica Rodriguez	Conato	<span>X</span> <span>✓</span>
MKT-7LG	Producto 14	99.91	CARNILOVE	Angelica Rodriguez	Entrenadores	<span>X</span> <span>✓</span>
MKT-9HJ	Producto 15	127.64	EARTH BORN	Angelica Rodriguez	Gato	<span>X</span> <span>✓</span>

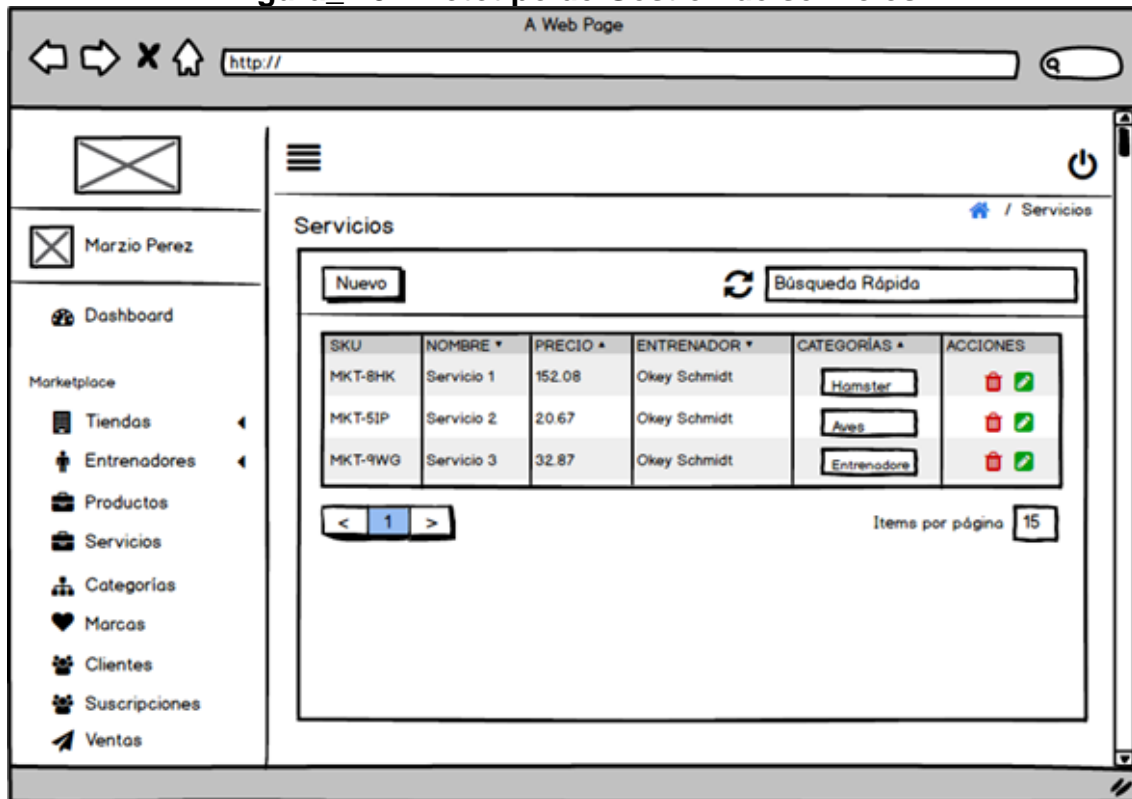
**RF8: El sistema debe permitir registrar, modificar, buscar, consultar y eliminar los servicios que se ofrecen en la plataforma. Con los siguientes campos: sku, nombre, precio, entrenador, categorías, fotos, imagen destacada, presentaciones, características, cargar imágenes, metadatos**

**DISEÑO**

**PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 75: Prototipo de Gestión de servicios



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 76: Controlador de Gestión de servicios

```
ServiceController.php x
app > Http > Controllers > Backoffice > Admin > ServiceController.php > ServiceController > edit
15 class ServiceController extends Controller {
16
17     public function index() {
18         $title = "Servicios";
19
20         $categories = array_map(
21             function($item){
22                 return [
23                     "id" => $item['id'],
24                     "value" => $item['internal_name']
25                 ];
26             }, Category::where('type', 'products')->get()->toArray()
27         );
28
29         $stores = array_map(
30             function($item){
31                 return [
32                     "id" => $item['id'],
33                     "value" => $item['name']
34                 ];
35             }, Store::where("is_trainer", true)->get()->toArray()
36         );
37         return view('back-office.admin.services.index', compact('title', 'categories', 'stores'));
38     }
39
40     public function lists(){
41         $model = DB::select('call all_services');
42         return response()->json($model);
43     }
44
45     public function create(Request $request){
46         try{
47             $data = $request->all();
48
49             $slug_util = new SlugUtil();
50
51             $categories = explode(",", $data["categories"]);
52             $parent_slug_id = null;
53             if (count($categories) > 0){
54                 $primary_category = Category::find($categories[0]);
55                 $parent_slug_id = $primary_category["slug_route_id"];
56             }
57             $slug = $slug_util->createSlug($data["name"], $parent_slug_id);
58
59             $data["slug_route_id"] = $slug->id;
60             $data["is_service"] = true;
61         }
```

Figura\_ 77: Modelo de Gestión de servicios

```
Product.php X
app > Models > Product.php > Product > $fillable
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Product extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'sku_code',
20         'description',
21         'summary',
22         'featured_image_id',
23
24         'categories',
25
26         'is_featured',
27
28         'price',
29         'iva',
30         'price_with_iva',
31
32         'offer_price',
33         'offer_iva',
34         'offer_price_with_iva',
35
36         'is_service',
37         'service_attentions_per_hour',
38         'rate',
39
40         'brand_id',
41         'brand_name',
42
43         'slug_route_id',
44         'url',
45
46         'store_id',
```

Figura\_ 78: Vista de Gestión de servicios

```
index.blade.php x
resources > views > back-office > admin > services > index.blade.php > ...
1 @extends('layouts.back-office.admin', ["title" => $title])
2
3 @section('content')
4
5 @php
6     use App\Utils\ReactCrudField;
7     use App\Utils\ReactCrudSettingsBuilder;
8
9     $builder = new ReactCrudSettingsBuilder();
10
11     $sku_code = new ReactCrudField('sku_code');
12     $sku_code->title('SKU')->required(true)->width(2);
13     $builder->addField($sku_code);
14
15     $name = new ReactCrudField('name');
16     $name->title('Nombre')->required(true)->width(10);
17     $builder->addField($name);
18
19     $price = new ReactCrudField('price');
20     $price->title('Precio')->required(true)->width(2);
21     $builder->addField($price);
22
23     $store_id = new ReactCrudField('store_id');
24     $store_id->title('Entrenador')->type('map', $stores)->required(true)->width(3);
25     $builder->addField($store_id);
26
27     $categories_field = new ReactCrudField('categories');
28     $categories_field->title('Categorías')->type('json', $categories)->width(12);
29     $builder->addField($categories_field);
30
31     $actions = [];
32     $actions["custom"] = [];
33     $actions["create-custom"] = [
34         'url' => route('admin.products.create'),
35         'tabs' => [
36             [
37                 "name" => "Fotos",
38                 "type" => 'images',
39                 'multimedia-lists-url' => route('admin.multimedia.lists')
40             ],
41             [
42                 "name" => "Imagen destacada",
43                 "type" => 'featured_image',
44                 'multimedia-lists-url' => route('admin.multimedia.lists')
45             ],
46             [
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.



Figura\_ 79: Implementación de Gestión de servicios

SKU	NOMBRE	PRECIO	ENTRENADOR	CATEGORÍAS	ACCIONES
MKT-8HK	Servicio 1	152.08	Okey Schmidt	Humano	[X] [✓]
MKT-5IP	Servicio 2	20.67	Okey Schmidt	Aves	[X] [✓]
MKT-9WG	Servicio 3	32.87	Okey Schmidt	Entrenadores	[X] [✓]
MKT-7PF	Servicio 4	96.87	Okey Schmidt	Conejo	[X] [✓]
MKT-2DZ	Servicio 5	95.35	Okey Schmidt	Entrenadores	[X] [✓]
MKT-1SN	Servicio 6	188.99	Okey Schmidt	Gato	[X] [✓]
MKT-1JH	Servicio 7	97.99	Okey Schmidt	Perros	[X] [✓]
MKT-3QQ	Servicio 8	26.74	Okey Schmidt	Entrenadores	[X] [✓]
MKT-2PB	Servicio 9	113.37	Okey Schmidt	Entrenadores	[X] [✓]
MKT-9KJ	Servicio 10	142.8	Okey Schmidt	Humano	[X] [✓]
MKT-9KC	Servicio 11	45.69	Okey Schmidt	Aves	[X] [✓]
MKT-7VL	Servicio 12	136.38	Okey Schmidt	Humano	[X] [✓]

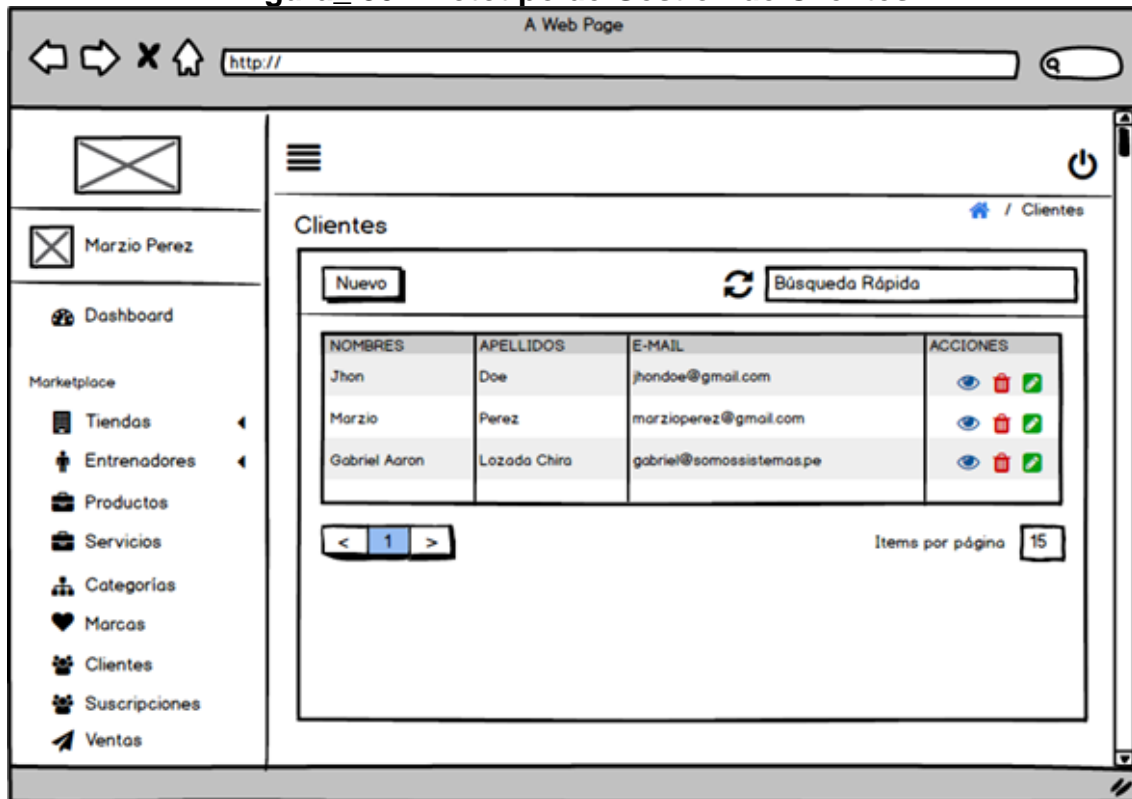
**RF9: El sistema debe permitir registrar, modificar, buscar, consultar y eliminar los clientes, con los campos: nombres, apellidos, tipo de documento, número de documento, teléfono, dirección, email y contraseña.**

## DISEÑO

### PROTOTIPO

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 80: Prototipo de Gestión de Clientes



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 81: Controlador de Gestión de Clientes

```
ClientController.php x
app > Http > Controllers > Backoffice > Admin > ClientController.php > ClientController > lists
1  ?php
2
3  namespace App\Http\Controllers\Backoffice\Admin;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Client;
7  use App\Models\User;
8  use Exception;
9  use Illuminate\Http\Request;
10
11 class ClientController extends Controller {
12
13     public function index() {
14         $title = "Clientes";
15         return view('back-office.admin.client', compact('title'));
16     }
17
18     public function lists(){
19         $model = Client::get();
20         return response()->json($model);
21     }
22
23     public function create(Request $request){
24         $data = $request->all();
25         $exist_email = User::where("email", $data["email"])->get()->first();
26         if($exist_email) {
27             return response()->json(["status" => "error", "message" => "El correo ingresado ya existe."]);
28         }else {
29
30             $user = User::create([
31                 'first_name' => $data["first_name"],
32                 'last_name' => $data["last_name"],
33                 'email' => $data["email"],
34                 'password' => $data["password"]
35             ]);
36             unset($data["first_name"]);
37             unset($data["last_name"]);
38             unset($data["email"]);
39             unset($data["password"]);
40
41             $data["user_id"] = $user->id;
42             $data["verified"] = true;
43             $model = Client::create($data);
44             $model = Client::find($model->id);
45             return response()->json(['status'=>"ok", 'data' => $model]);
46         }
47     }

```

Lín. 19, col. 21 Espacios: 4 UTF-8 LF PHP

Figura\_ 82: Modelo de Gestión de Clientes

```
app > Models > Client.php > Client
1  ?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class Client extends Model {
10
11     use HasFactory;
12
13     use SoftDeletes;
14
15     protected $dates = ["deleted_at"];
16
17     protected $fillable = [
18         'id',
19         'cellphone',
20         // DNI | CEDULA | PASAPORTE
21         'document_type',
22         'document_number',
23         'address',
24         "user_id",
25         // N: NATURAL - J: JURIDICO
26         'type',
27         'ruc',
28         'business_name',
29         // PENDING | BLOCKED | CONFIRM
30         'status',
31         'verified',
32         'verify_token'
33     ];
34
```

Figura\_ 83: Vista de Gestión de Clientes

```

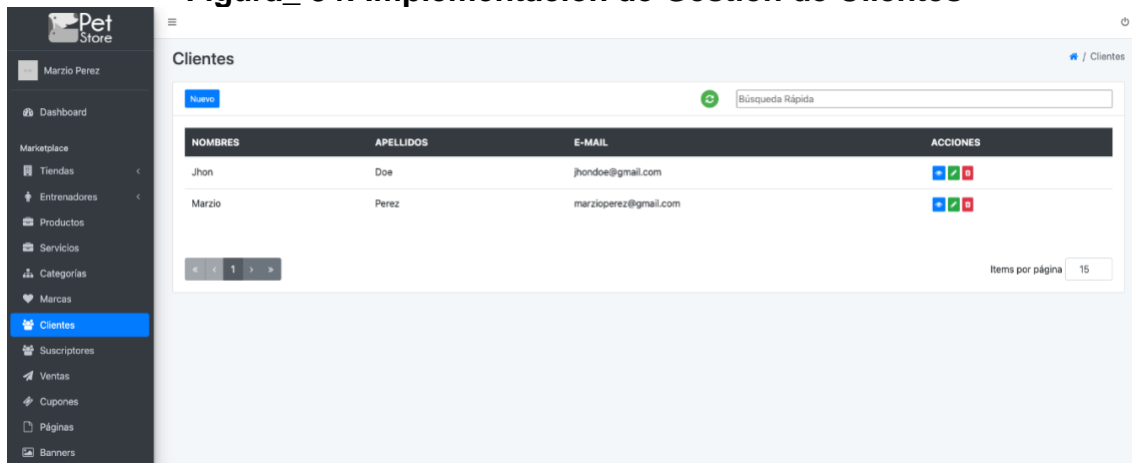
client.blade.php
resources > views > back-office > admin > client.blade.php > ...
1  @extends('layouts.back-office.admin', ["title" => $title])
2
3  @section('content')
4
5  @php
6      use App\Utils\ReactCrudField;
7      use App\Utils\ReactCrudSettingsBuilder;
8
9      $builder = new ReactCrudSettingsBuilder();
10
11      $first_name = new ReactCrudField('first_name');
12      $first_name->title('Nombres')->required(true)->width(3);
13      $builder->addField($first_name);
14
15      $last_name = new ReactCrudField('last_name');
16      $last_name->title('Apellidos')->required(true)->width(3);
17      $builder->addField($last_name);
18
19      $document_type = new ReactCrudField('document_type');
20      $document_type->title('Tipo de documento')->type('map', [
21          ['id' => "DNI", 'value' => 'D.N.I.'],
22          ['id' => "CEDULA", 'value' => 'C.E.'],
23          ['id' => "PASAPORTE", "value" => "Pasaporte"],
24          ['id' => "OTRO", "value" => "Otro"]
25      ]->required(true)->show(false)->width(3);
26      $builder->addField($document_type);
27
28      $document_number = new ReactCrudField('document_number');
29      $document_number->title('Número de documento')->required(true)->show(false)->width(3);
30      $builder->addField($document_number);
31
32      $phone = new ReactCrudField('cellphone');
33      $phone->title('Teléfono')->required(false)->show(false)->width(3);
34      $builder->addField($phone);
35
36      $address = new ReactCrudField('address');
37      $address->title('Dirección')->show(false)->width(9);
38      $builder->addField($address);
39
40      $address = new ReactCrudField('email');
41      $address->title('E-mail')->required(true)->width(6);
42      $builder->addField($address);
43
44      $address = new ReactCrudField('password');
45      $address->title('Contraseña')->type('password')->show(false)->width(6);
46      $builder->addField($address);
47

```

**IMPLEMENTACIÓN**

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

Figura\_ 84: Implementación de Gestión de Clientes



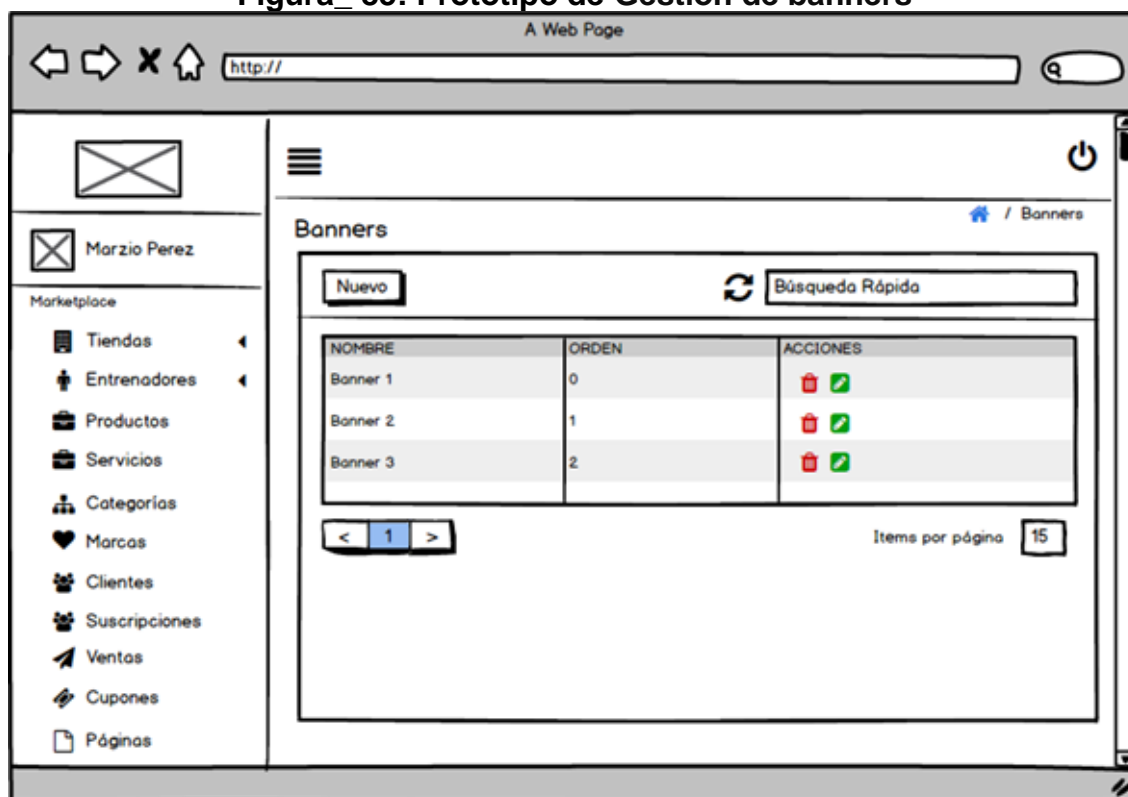
**RF10:** El sistema debe permitir registrar, modificar, buscar, consultar y eliminar banners informativos, para las diferentes promociones, con los campos: nombre, url, orden, URL de Imagen 1900 x 800, URL de Imagen 1400 x 700, URL de Imagen 800 x 700, URL de Imagen 600 x 700

## DISEÑO

### PROTOTIPO

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 85: Prototipo de Gestión de banners**



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

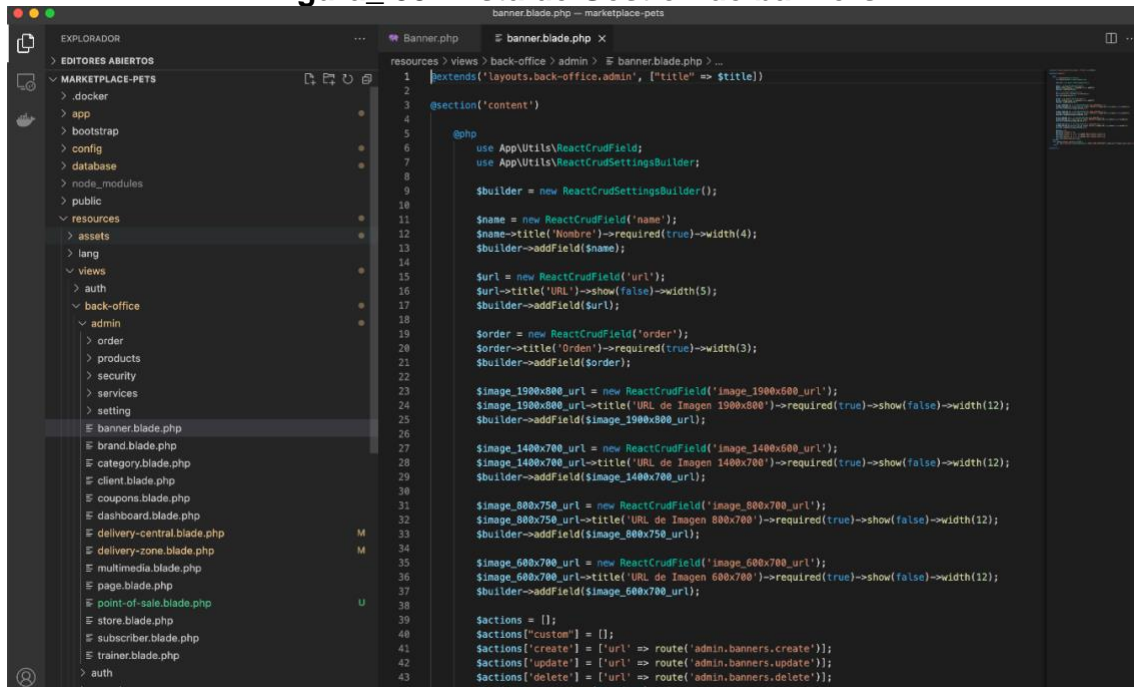
Figura\_ 86: Controlador de Gestión de banners

```
BannerController.php — marketplace-pets
BannerController.php x
app > Http > Controllers > Backoffice > Admin > BannerController.php > ...
1 <?php
2
3 namespace App\Http\Controllers\Backoffice\Admin;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Banner;
7 use Exception;
8 use Illuminate\Http\Request;
9
10 class BannerController extends Controller {
11
12     public function index() {
13         $title = "Banners";
14         return view('back-office.admin.banner', compact('title'));
15     }
16
17     public function lists(){
18         $model = Banner::get();
19         return response()->json($model);
20     }
21
22     public function create(Request $request){
23         try{
24             $data = $request->all();
25             $model = Banner::create($data);
26             $model = Banner::find($model->id);
27         }catch(Exception $e) {
28             return response()->json(['status'=>'error', 'message' => "No se pudo actualizar el registro.")]
29         }
30         return response()->json(['status'=>"ok", 'data' => $model]);
31     }
32
33     public function update(Request $request) {
34         $data = $request->all();
35         $brand = Banner::find($data['id']);
36         $brand->update($data);
37         $model = Banner::find($data['id']);
38         return response()->json(["status" => "ok", "data" => $model]);
39     }
40
41     public function delete(Request $request){
42         $data = $request->all();
43         $model = Banner::find($data['id']);
44         if($model->delete()) {
45             return response()->json(['status'=>'ok', 'data'=>$model]);
46         }
47     }
48 }
```

Figura\_ 87: Modelo de Gestión de banners

```
Banner.php — marketplace-pets
Banner.php x
app > Models > Banner.php > ...
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Banner extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ['deleted_at'];
15     protected $fillable = [
16         'id',
17         'name',
18         "image_1900x600_url",
19         "image_1400x600_url",
20         "image_800x700_url",
21         "image_600x700_url",
22         "url",
23         "order"
24     ];
25
26 }
27 }
```

Figura\_ 88: Vista de Gestión de banners

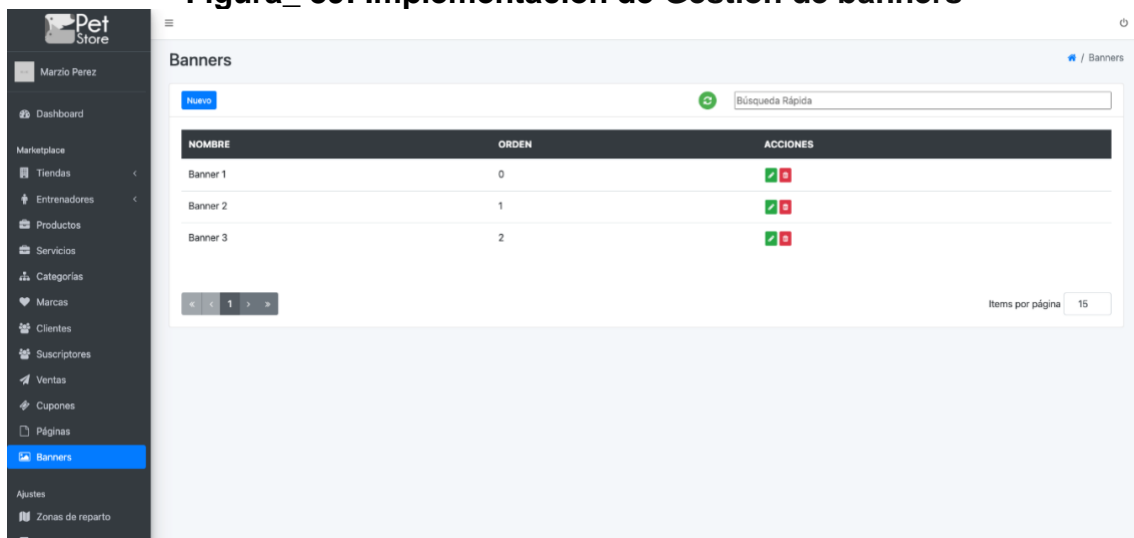


```
resources > views > back-office > admin > banner.blade.php > ...
1 @extends('layouts.back-office.admin', ['title' => $title])
2
3 @section('content')
4
5     @php
6         use App\Utils\ReactCrudField;
7         use App\Utils\ReactCrudSettingsBuilder;
8
9         $builder = new ReactCrudSettingsBuilder();
10
11         $name = new ReactCrudField('name');
12         $name->title('Nombre')->required(true)->width(4);
13         $builder->addField($name);
14
15         $url = new ReactCrudField('url');
16         $url->title('URL')->show(false)->width(5);
17         $builder->addField($url);
18
19         $order = new ReactCrudField('order');
20         $order->title('Orden')->required(true)->width(3);
21         $builder->addField($order);
22
23         $image_1900x800_url = new ReactCrudField('image_1900x800_url');
24         $image_1900x800_url->title('URL de Imagen 1900x800')->required(true)->show(false)->width(12);
25         $builder->addField($image_1900x800_url);
26
27         $image_1400x700_url = new ReactCrudField('image_1400x700_url');
28         $image_1400x700_url->title('URL de Imagen 1400x700')->required(true)->show(false)->width(12);
29         $builder->addField($image_1400x700_url);
30
31         $image_800x750_url = new ReactCrudField('image_800x750_url');
32         $image_800x750_url->title('URL de Imagen 800x750')->required(true)->show(false)->width(12);
33         $builder->addField($image_800x750_url);
34
35         $image_600x700_url = new ReactCrudField('image_600x700_url');
36         $image_600x700_url->title('URL de Imagen 600x700')->required(true)->show(false)->width(12);
37         $builder->addField($image_600x700_url);
38
39         $actions = [];
40         $actions["custom"] = [];
41         $actions["create"] = ['url' => route('admin.banners.create')];
42         $actions["update"] = ['url' => route('admin.banners.update')];
43         $actions["delete"] = ['url' => route('admin.banners.delete')];
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

Figura\_ 89: Implementación de Gestión de banners





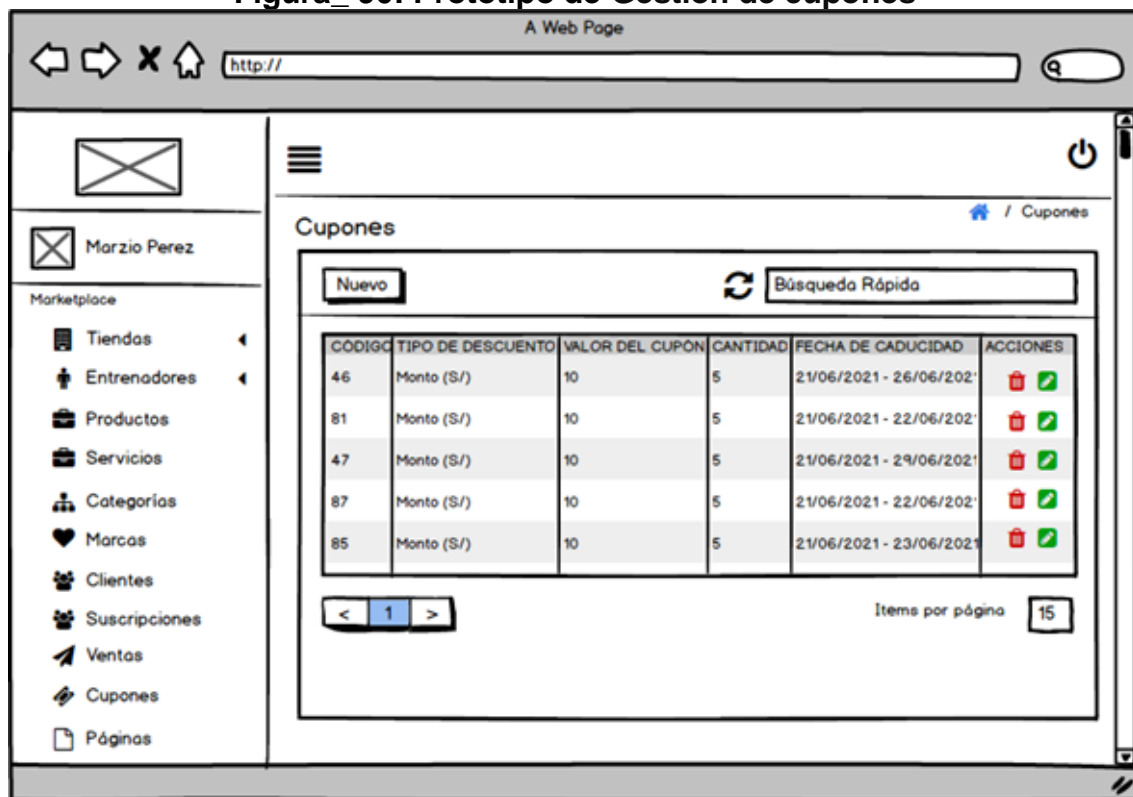
**RF11: El sistema permite la gestión de cupones para generar descuentos por parte de los clientes, de manera pública o privada y además permitiendo limitar el tiempo de su uso.**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 90: Prototipo de Gestión de cupones**



## **DESARROLLO**

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 91: Controlador de Gestión de cupones

```
CouponController.php x
app > Http > Controllers > Backoffice > Admin > CouponController.php > ...
1 <?php
2
3 namespace App\Http\Controllers\Backoffice\Admin;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Client;
7 use App\Models\Coupon;
8 use App\Models\Store;
9 use Exception;
10 use Illuminate\Http\Request;
11
12 class CouponController extends Controller {
13
14     public function index() {
15         $title = "Cupones";
16         $stores = array_map(
17             function($store){
18                 return [
19                     "id" => $store["id"],
20                     "value" => $store["name"]
21                 ];
22             },
23             Store::all()->toArray()
24         );
25         return view('back-office.admin.coupons', compact('title', 'stores'));
26     }
27
28     public function lists(){
29         $model = Coupon::with('usages')->get();
30         return response()->json($model);
31     }
32
33     public function create(Request $request){
34         try{
35             $data = $request->all();
36             $data["quantity"] = ($data["is_public"] ? $data["quantity"] : 1);
37             $range_date = json_decode($data["range_date"]);
38             unset($data["range_date"]);
39             $data['start_date'] = $range_date->startDate;
40             $data['end_date'] = $range_date->endDate;
41             $model = Coupon::create($data);
42             $model = Coupon::find($model->id);
43         }catch(Exception $e) {
44             return response()->json(['status'=>'error', 'message' => "No se pudo actualizar el registro."]);
45         }
46         return response()->json(['status'=>'ok', 'data' => $model]);
47     }
48 }
```

Figura\_ 92: Modelo de Gestión de cupones

```
Coupon.php x
app > Models > Coupon.php > ...
1 |-?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Coupon extends Model {
10     use HasFactory;
11     use SoftDeletes;
12
13     protected $dates = ["deleted_at"];
14
15     protected $guard = 'admin';
16
17     protected $fillable = [
18         'id',
19         'code',
20         'store_ids',
21         // percentage | amount
22         'type',
23         'value',
24         'start_date',
25         'end_date',
26         'quantity'
27     ];
28
29     protected $appends = ["usages_count", "range_date"];
30
31     public function usages(){
32         return $this->hasMany('App\Models\CouponUsage', 'coupon_id', 'id');
33     }
34
35     public function getUsagesCountAttribute(){
36         $usages = $this->usages;
37         return count($usages);
38     }
39
40     public function getRangeDateAttribute(){
41         return [
42             "start_date" => $this->attributes["start_date"],
43             "end_date" => $this->attributes["end_date"]
44         ];
45     }
46 }
47
Lin. 1, col. 1 Espacios: 4 UTF-8 LF PHP
```

Figura\_ 93: Vista de Gestión de cupones

```
resources > views > back-office > admin > coupons.blade.php > ...
1 | @extends('layouts.back-office.admin', ["title" => $title])
2
3 | @section('content')
4
5 | @php
6 |     use App\Utils\ReactCrudField;
7 |     use App\Utils\ReactCrudSettingsBuilder;
8
9 |     $builder = new ReactCrudSettingsBuilder();
10
11 |     $codeField = new ReactCrudField('code');
12 |     $codeField->title('Código')->required(true)->width(3);
13 |     $builder->addField($codeField);
14
15 |     $typeField = new \App\Utils\ReactCrudField('type');
16 |     $typeField->title('Tipo de descuento')->type('map', [
17 |         ['id' => 'amount', 'value' => 'Monto (' . env('CURRENCY_SYMBOL') . ')'],
18 |         ['id' => 'percentage', 'value' => 'Porcentaje (%)']
19 |     ]->default("amount")->width(3);
20 |     $builder->addField($typeField);
21
22 |     $valueField = new ReactCrudField('value');
23 |     $valueField->title('Valor del cupón')->required(true)->width(3);
24 |     $builder->addField($valueField);
25
26 |     $quantityField = new ReactCrudField('quantity');
27 |     $quantityField->title('Cantidad')->required(true)->width(3);
28 |     $builder->addField($quantityField);
29
30 |     $store_ids = new \App\Utils\ReactCrudField('store_ids');
31 |     $store_ids->title('Tiendas')->type('json', $stores)->width(12)->show(false);
32 |     $builder->addField($store_ids);
33
34 |     $range_dateField = new \App\Utils\ReactCrudField('range_date');
35 |     $range_dateField->title('Fecha de caducidad')->type('date_range')->width(12);
36 |     $builder->addField($range_dateField);
37
38 |     $actions = [];
39 |     $actions["custom"] = [];
40 |     $actions['create'] = ['url' => route('admin.coupons.create')];
41 |     $actions['update'] = ['url' => route('admin.coupons.update')];
42 |     $actions['delete'] = ['url' => route('admin.coupons.delete')];
43 |     $builder->setActions($actions);
44
45 | @endphp
46 | <div class="content custom-scrollbar">
47 |     <div id="{{\App\Utils\ReactComponents::LARAVEL_CRUD_COMPONENT}}" data-url="{{route('admin.coupons.lists')}}">
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 94: Implementación de Gestión de cupones**

CÓDIGO	TIPO DE DESCUENTO	VALOR DEL CUPÓN	CANTIDAD	FECHA DE CADUCIDAD	ACCIONES
46	Monto (\$)	10	5	21/06/2021 - 26/06/2021	<input checked="" type="checkbox"/> <input type="checkbox"/>
81	Monto (\$)	10	5	21/06/2021 - 22/06/2021	<input checked="" type="checkbox"/> <input type="checkbox"/>
47	Monto (\$)	10	5	21/06/2021 - 29/06/2021	<input checked="" type="checkbox"/> <input type="checkbox"/>
87	Monto (\$)	10	5	21/06/2021 - 22/06/2021	<input checked="" type="checkbox"/> <input type="checkbox"/>
85	Monto (\$)	10	5	21/06/2021 - 23/06/2021	<input checked="" type="checkbox"/> <input type="checkbox"/>

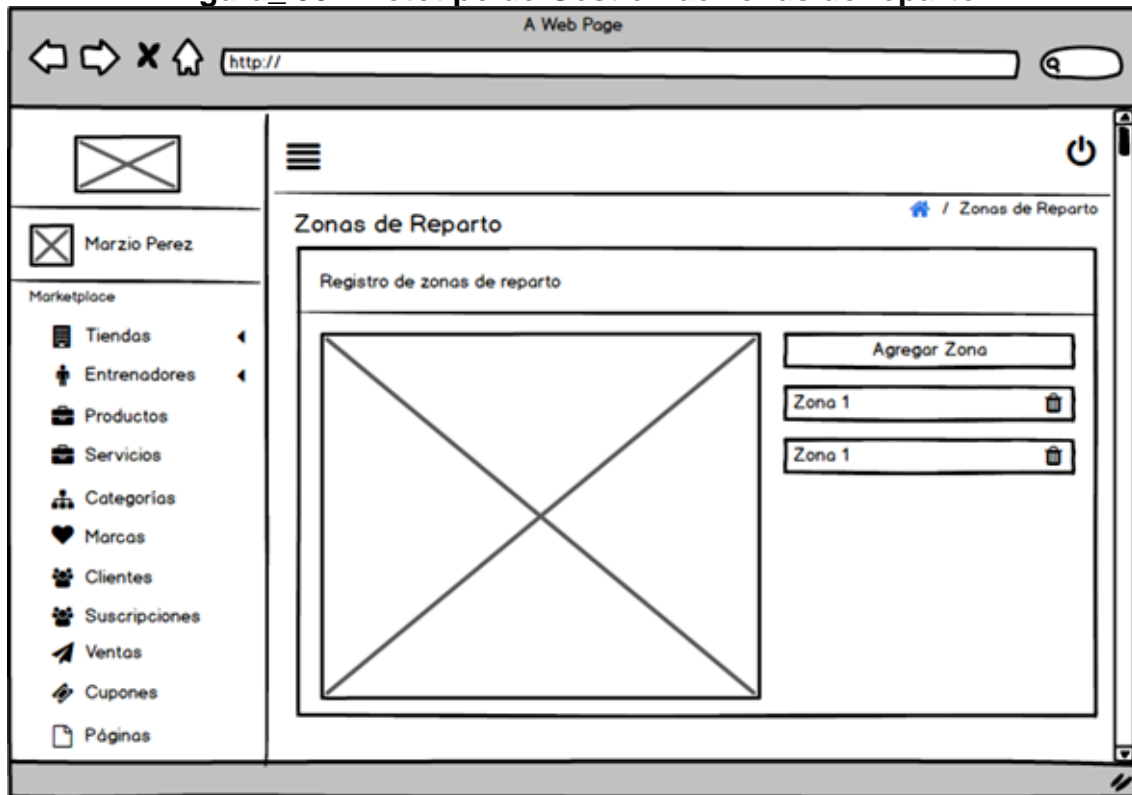
**RF12: El sistema permite el registro de zonas de reparto, con el fin de poder trazar líneas en un mapa de google y delimitar las zonas a donde se podrán realizar las entregas de los pedidos.**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 95: Prototipo de Gestión de zonas de reparto**



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 96: Controlador de Gestión de zonas de reparto

```
DeliveryZoneController.php x
app > Http > Controllers > Backoffice > Admin > DeliveryZoneController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Backoffice\Admin;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\DeliveryZone;
7  use App\Models\DeliveryZoneBound;
8  use Illuminate\Http\Request;
9
10 class DeliveryZoneController extends Controller {
11
12     public function index() {
13         $title = "Zonas de reparto";
14         $zones = DeliveryZone::with('bounds')->get();
15         return view('back-office.admin.delivery-zone', compact('title', 'zones'));
16     }
17
18     public function create(Request $request) {
19         $data = $request->all();
20         $bounds = $data["bounds"];
21         unset($data["bounds"]);
22         $delivery_zone = DeliveryZone::create($data);
23         if(count($bounds) > 0) {
24             foreach($bounds as $bound) {
25                 DeliveryZoneBound::create([
26                     "latitude" => $bound["lat"],
27                     "longitude" => $bound["lng"],
28                     "delivery_zone_id" => $delivery_zone->id
29                 ]);
30             }
31         }
32         return response()->json(["status" => "ok", "redirect_url" => route('admin.delivery.zones')]);
33     }
34
35     public function update(Request $request) {
36         $data = $request->all();
37         $delivery_zone = DeliveryZone::find($data["id"]);
38         unset($data["bounds"]);
39         $delivery_zone->update($data);
40         return response()->json(["status" => "ok", "redirect_url" => route('admin.delivery.zones')]);
41     }
42
43     public function delete(Request $request) {
```

Figura\_ 97: Modelo de Gestión de zonas de reparto

```
DeliveryZone.php x
app > Models > DeliveryZone.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class DeliveryZone extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'color',
20         'moto_km_price',
21         'moto_min_price',
22         'auto_km_price',
23         'auto_min_price'
24     ];
25
26     public function bounds(){
27         return $this->hasMany('App\Models\DeliveryZoneBound', 'delivery_zone_id', 'id');
28     }
29 }
30
```

Figura\_ 98: Vista de Gestión de zonas de reparto

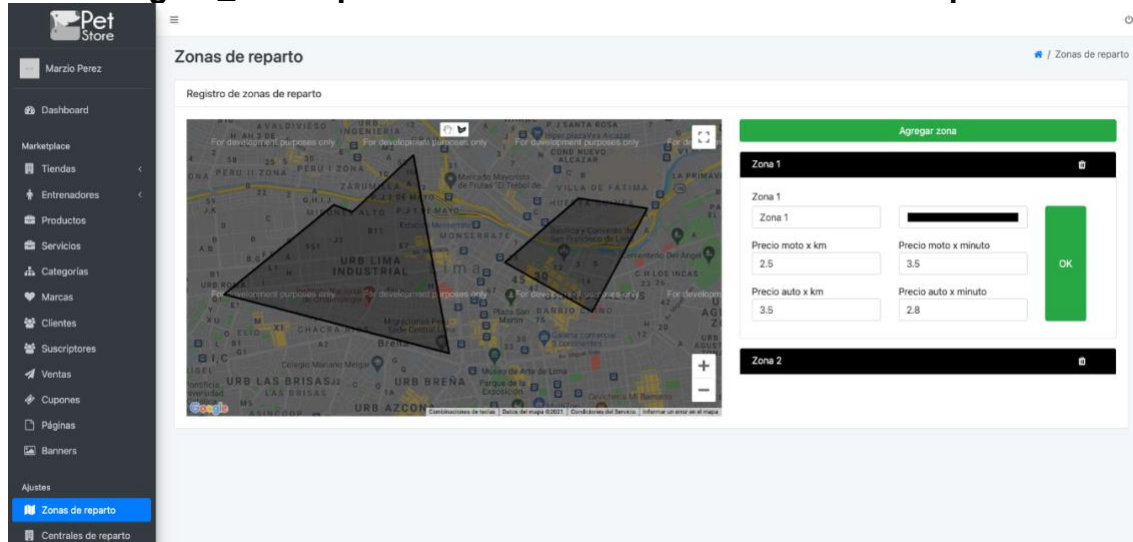
```
delivery-zone.blade.php M x
resources > views > back-office > admin > delivery-zone.blade.php > ...
1  @extends('layouts.back-office.admin', ["title" => $title])
2
3  @section('content')
4      @php
5          $google_maps_api_key = \App\Models\Setting::where("key", "GOOGLE_API_KEY")->get()->first();
6          $google_maps_api_key_value = $google_maps_api_key["value"];
7      @endphp
8      <div class="row">
9          <div class="col-12">
10             <div class="card">
11                 <div class="card-header bg-black">
12                     <h3 class="card-title">Registro de zonas de reparto</h3>
13                 </div>
14                 <div class="card-body">
15                     <div class="delivery-zones-component">
16                         create-delivery-zone-url="{{route('admin.delivery.zones.create')}}"
17                         update-delivery-zone-url="{{route('admin.delivery.zones.update')}}"
18                         delete-delivery-zone-url="{{route('admin.delivery.zones.delete')}}"
19                         zones="{{json_encode($zones)}}"
20                         google-api-key="{{ $google_maps_api_key_value }}"</div>
21                     </div>
22                 </div>
23             </div>
24         </div>
25     </div>
26 @endsection
27
```

## IMPLEMENTACIÓN



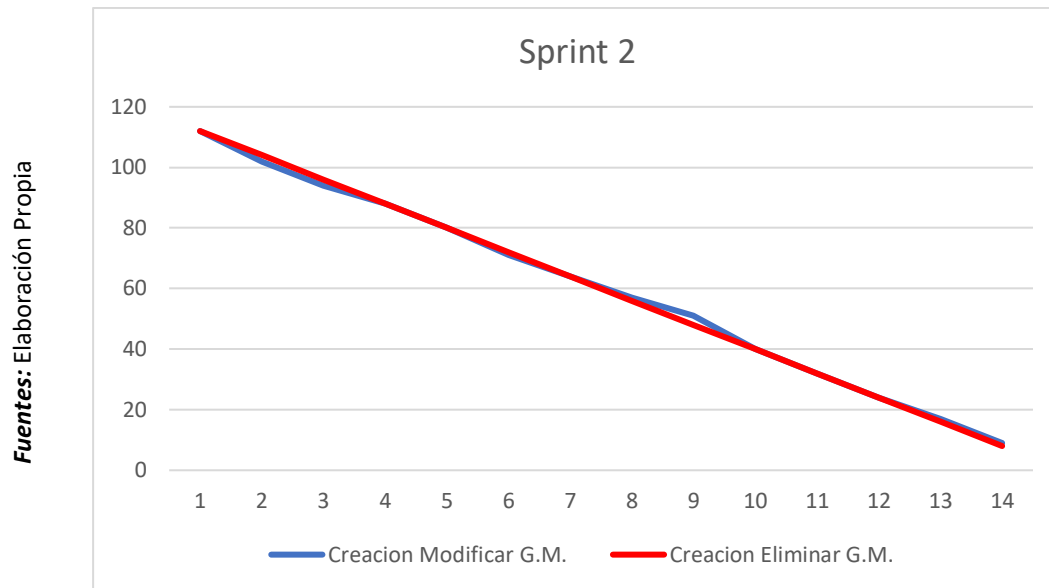
A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el product owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 99: Implementación de Gestión de zonas de reparto**



## BURNDOWN DEL SPRINT N° 2

**Figura\_ 100: Burndown del Sprint 2**



### Burndown Sprint 2

En la figura vista, se muestra el proceso de desarrollo de cada sprint la cual está representada a través de un gráfico, como podemos observar tenemos dos líneas la cual la línea roja representa al tiempo ideal que se define para el desarrollo del sprint, de igual manera la línea azul representa el tiempo real que tomo el desarrollo del sprint. Ya teniendo bien definido las representaciones tenemos que concluir que si la línea azul se encuentra por debajo de la roja significa que se presentó un adelanto, de lo contrario si la línea azul está por encima es decir que hubo un retraso.

### ACTA DE REUNION DEL SPRINT N° 2

Siendo las 3 pm del día 5 de junio del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.

Presentes:


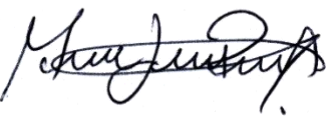


ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas

Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El encargado de la Empresa AC Brands & Consulting E.I.R.L., Isaac Abensur Vargas concluye la reunión con los Sres. Marzio Perez y Nicolás Segovia aclarando los últimos puntos sobre el sprint.

Para determinar las formas de trabajo y los requerimientos respectivos para esta interacción, también la fecha de entrega y firmar el acta para dar por terminado la reunión.

Los participantes manifestaron su aprobación respecto a todos los puntos definidos en la reunión, que fueron de ayuda a la planificación del Sprint, manifestando su compromiso para la entrega de los resultados en las fechas pactadas.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## ACTA DE ENTREGA DEL SPRINT N° 2


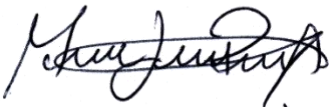


Siendo la 1 pm del día 25 de junio del 2021 se reúne en la empresa AC Brands & Consulting E.I.R.L.

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

Los Sres. Marzio Perez y Nicolás Segovia, especifican cada uno de los requerimientos desarrollados, también presentan las interfaces elaboradas y brindadas por el producto Owner.

Luego de ser verificadas las presentaciones realizadas por los Srs. Marzio Perez y Nicolás Segovia, se decidió de manera unánime, dar por aprobado el término del Sprint respectivo, del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Los presentes manifestarán su aprobación al informe presentado por los Sres. Marzio Perez y Nicolás Segovia del Sprint N° 2 terminado del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

### RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 2

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	AC Brands & Consulting E.I.R.L.
<b>Proyecto</b>	“Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Información de la reunión:

<b>Lugar</b>	AC Brands & Consulting E.I.R.L.
<b>Fecha</b>	25/06/2021

<b>Número de iteración / Sprint</b>	Sprint 2
<b>Personas Convocadas a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza
<b>Persona que asistieron a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
<p>Se desarrolló correctamente la gestión de productos y servicios.</p> <p>Se desarrolló exitosamente la gestión de clientes.</p> <p>Se logró desarrollar la gestión de banners.</p> <p>Se logró desarrollar la gestión de cupones.</p> <p>Así mismo se desarrolló la gestión de zonas de reparto.</p>	<p>Se pudo observar fallas en los registros, pero se pudo solucionar.</p>

### PLANIFICACIÓN DEL SPRINT N°3

Siendo las 4 pm del día 25 de junio del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.


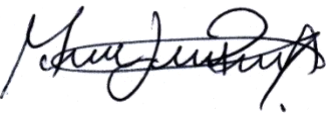


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El gerente de la Empresa AC Brands & Consulting E.I.R.L., efectúa la presentación de los requerimientos, de igual manera manifiesta los requerimientos que tienen mayor prioridad.

Realizado el análisis de los requerimientos presentados por el gerente de AC Brands & Consulting E.I.R.L., el señor Juan Galarza aclara dudas existentes y luego se compromete al cumplimiento de todos los requerimientos propuestas en el Sprint 3.

Los participantes de la reunión manifiestan su aprobación con respecto a la presentación del Sprint 3, así mismo especificando que la fecha de entrega será el día 10 de julio del 2021.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

### EJECUCIÓN DEL SPRINT 3

Lo primero que se debe realizar antes del diseño, es conocer y comprender el funcionamiento del sistema, en otras palabras, analizar lo que realmente se necesita, lo cual será elaborado según las historias de usuario.

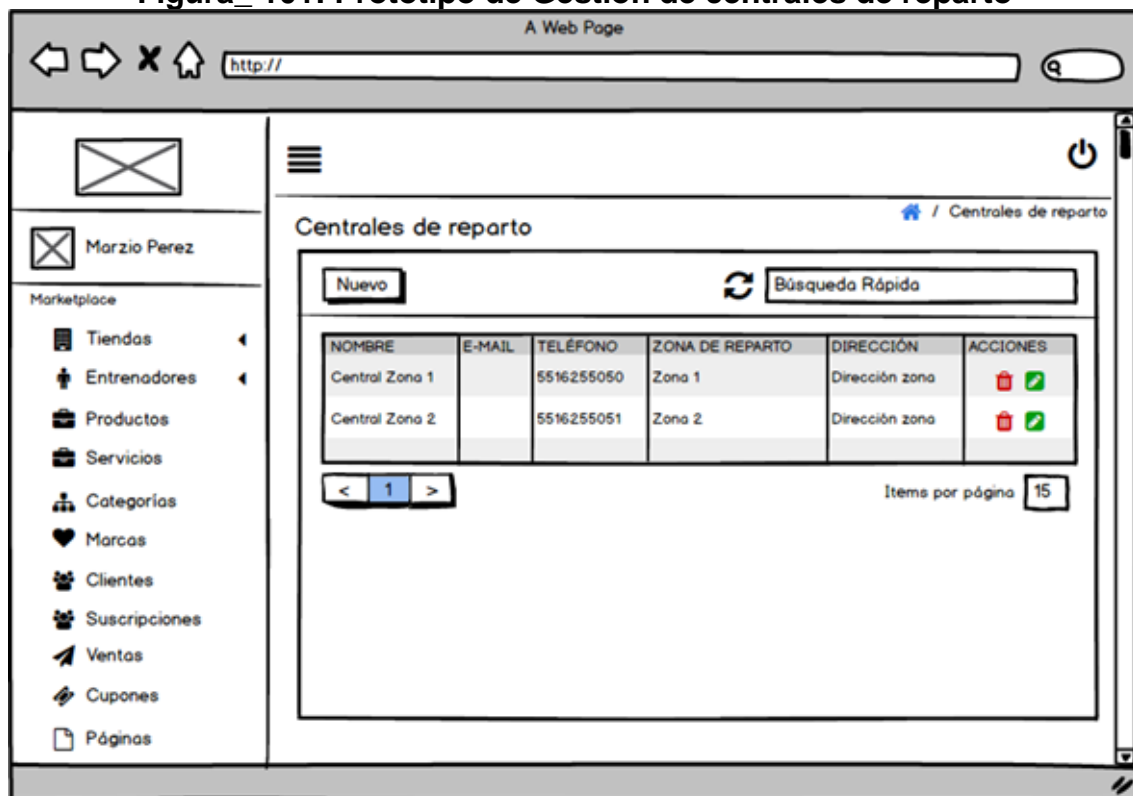
**RF13: El sistema debe permitir registrar, modificar, buscar, consultar y eliminar de las centrales de reparto, las cuales se encuentran en cada zona de reparto con el fin de que los clientes puedan venir a recoger sus productos en dicho establecimiento, con los campos: nombre, email, teléfono, zona de reparto, dirección.**

## DISEÑO

### PROTOTIPO

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 101: Prototipo de Gestión de centrales de reparto**



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

**Figura\_ 102: Controlador de Gestión de centrales de reparto**

```
DeliveryCentralController.php x
app > Http > Controllers > Backoffice > Admin > DeliveryCentralController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Backoffice\Admin;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\DeliveryCentral;
7  use App\Models\DeliveryZone;
8  use App\Utils\MarketUtil;
9  use Illuminate\Http\Request;
10
11 class DeliveryCentralController extends Controller {
12
13     public function index() {
14         $title = "Centrales de reparto";
15         $zones = array_map(
16             function($store){
17                 return [
18                     "id" => $store["id"],
19                     "value" => $store["name"]
20                 ];
21             },
22             DeliveryZone::all()->toArray()
23         );
24         return view('back-office.admin.delivery-central', compact('title', 'zones'));
25     }
26
27     public function lists(){
28         $model = DeliveryCentral::get();
29         return response()->json($model);
30     }
31
32     public function create(Request $request){
33         $data = $request->all();
34         $place_address = json_decode($data["place_info"]);
35         $market_util = new MarketUtil();
36         $get_lat_lng = $market_util->getLatLng($place_address->value->place_id);
37         if($get_lat_lng["status"] === "ok") {
38             $data["address"] = $place_address->label;
39             $data["latitude"] = $get_lat_lng["data"]->result->geometry->location->lat;
40             $data["longitude"] = $get_lat_lng["data"]->result->geometry->location->lng;
41             $data["place_info"] = json_encode($place_address);
42             $model = DeliveryCentral::create($data);
43             return response()->json(["status" => "ok", "data" => $model]);
44         }else {
45             return response()->json(["status" => "error", "message" => "No hemos podido ubicar la dirección"]);
46         }
47     }
48 }
```



Figura\_ 103: Modelo de Gestión de centrales de reparto

```
DeliveryCentral.php ×
app > Models > DeliveryCentral.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class DeliveryCentral extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'email',
20         "delivery_zone_id",
21         'address',
22         'phone',
23         "latitude",
24         "longitude",
25         "place_info"
26     ];
27 }
28
```

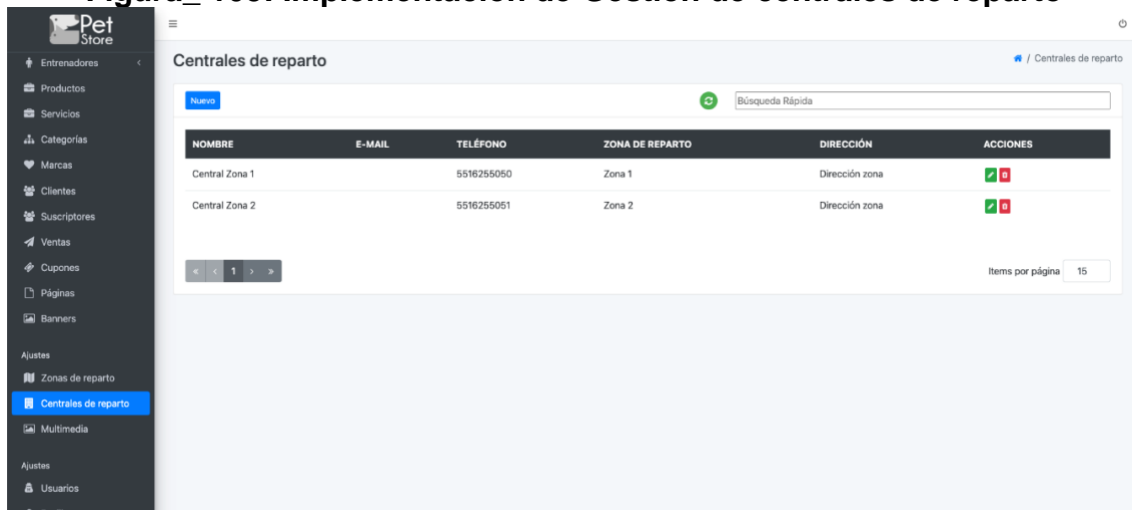
Figura\_ 104: Vista de Gestión de centrales de reparto

```
delivery-central.blade.php M X
resources > views > back-office > admin > delivery-central.blade.php > ...
1 @extends('layouts.back-office.admin', ["title" => $title])
2
3
4 @section('content')
5
6 @php
7     use App\Utils\ReactCrudField;
8     use App\Utils\ReactCrudSettingsBuilder;
9
10    $builder = new ReactCrudSettingsBuilder();
11
12    $name = new ReactCrudField('name');
13    $name->title('Nombre')->required(true)->width(4);
14    $builder->addField($name);
15
16    $email = new ReactCrudField('email');
17    $email->title('E-mail')->required(true)->width(4);
18    $builder->addField($email);
19
20    $phone = new ReactCrudField('phone');
21    $phone->title('Teléfono')->required(true)->width(4);
22    $builder->addField($phone);
23
24    $delivery_zone_id = new ReactCrudField('delivery_zone_id');
25    $delivery_zone_id->title('Zona de reparto')->type('map', $zones)->required(true)->width(12);
26    $builder->addField($delivery_zone_id);
27
28    $address = new ReactCrudField('address');
29    $address->title('Dirección')->fillable(false);
30    $builder->addField($address);
31
32    $place_info = new ReactCrudField('place_info');
33    $place_info->title('Dirección')->show(false)->type("google_autocomplete")->required(true)->width(12);
34    $builder->addField($place_info);
35
36    $actions = [];
37    $actions["custom"] = [];
38    $actions['create'] = ['url' => route('admin.delivery.centrales.create')];
39    $actions['update'] = ['url' => route('admin.delivery.centrales.update')];
40    $actions['delete'] = ['url' => route('admin.delivery.centrales.delete')];
41    $builder->setActions($actions);
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

Figura\_ 105: Implementación de Gestión de centrales de reparto



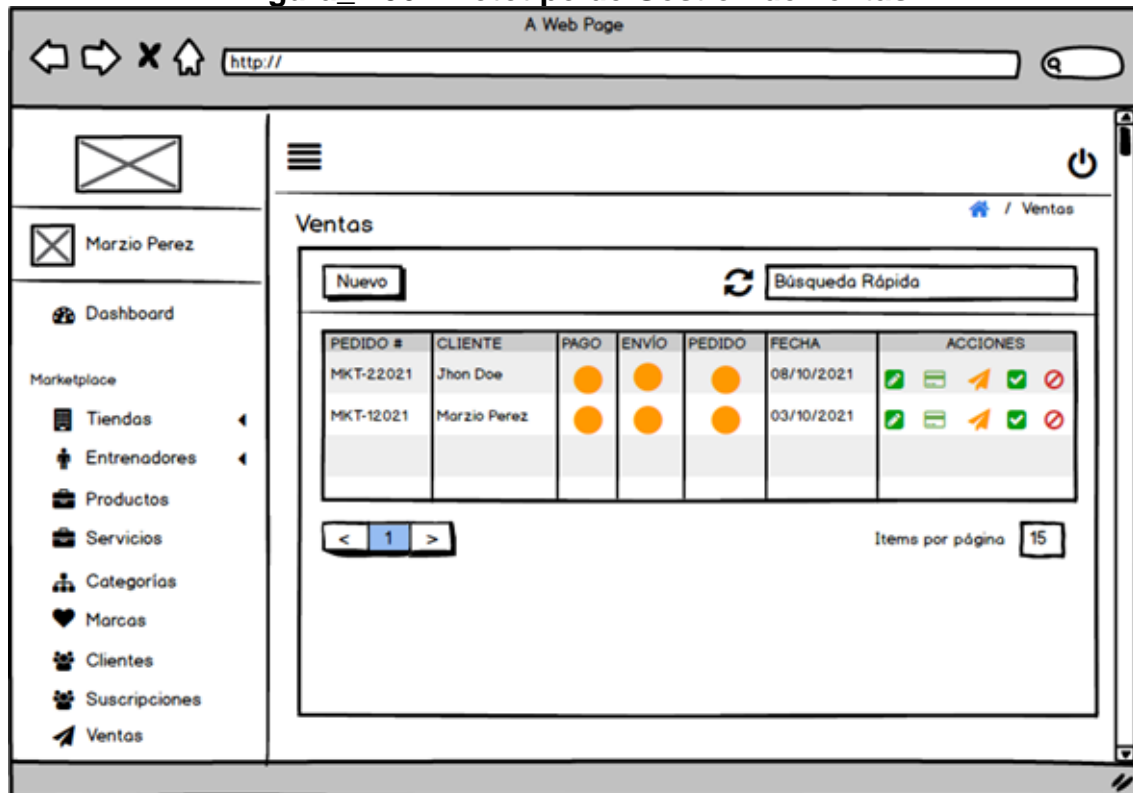
**RF14: El sistema permite visualizar las listas de las ventas y su detalle, con el fin de poder conocer el estado de las entregas y pagos que se realizan de las mismas.**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 106: Prototipo de Gestión de ventas**



## **DESARROLLO**

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 107: Controlador de Gestión de ventas

```
OrderController.php
~/Desarrollo/Proyectos/marketplace-pets/app/
Http/Controllers/Backoffice/Admin/
OrderController.php
Admin > OrderController.php > OrderController > canceled

18 use Illuminate\Http\Request;
19 use Illuminate\Support\Facades\DB;
20
21 class OrderController extends Controller {
22
23     public function index() {
24         $title = "Ventas";
25         return view('back-office.admin.order.index', compact('title'));
26     }
27
28     public function lists(){
29         $model = DB::select('call all_orders');
30         return response()->json($model);
31     }
32
33     public function detail(Request $request) {
34         $title = "Detalle de venta";
35         $data = $request->all();
36         $code = $data["order_code"];
37
38         $order = Order::with('detail')->where("order_code", $code)->get()->first();
39         return view('back-office.admin.order.detail', compact('title', 'order'));
40     }
41
42     public function attended_payment(Request $request) {
43         $data = $request->all();
44         $order = Order::find($data["id"]);
45         $order->update([
46             "payment_status" => "ATTENDED",
47             "payment_date" => date('Y-m-d H:i:s')
48         ]);
49         if($order["payment_type"] === "BANK-TRANSFER" || $order["payment_type"] === "YAPE"){
50             $items = OrderDetail::where("order_id", $order["id"])->get();
51             foreach ($items as $item){
52                 $inventory = Inventory::where("product_id", $item["product_id"])->get()->first();
53                 if($inventory){
54                     $inventory->update(["quantity" => $inventory["quantity"] - $item["quantity"]]);
55                     $inventory_variation = InventoryVariation::where("variation_id", $item["variation_id"])->where("b
56                     if($inventory_variation){
57                         $inventory_variation->update(["quantity" => $inventory_variation["quantity"] - $item["quantit
58                         InventoryMovement::create([
59                             // I: Ingreso | E: Egreso
60                             'movement_type' => "E",
61                             'quantity' => $item["quantity"],
62                             'inventory_id' => $inventory["id"],
63                             "product_name" => $item["product_name"],
```

Figura\_ 108: Modelo de Gestión de ventas

```
Order.php x
app > Models > Order.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class Order extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'order_code',
19
20         // ONLINE | OFFLINE
21         'type',
22
23         // PENDING, CANCELED, ATTENDED, REFOUNDED
24         'order_status',
25         'order_attended_date',
26         'order_closed_date',
27
28         // CREDIT-CARD | BANK-TRANSFER | CASH
29         'payment_type',
30         'bank_transfer_file',
31
32         // PENDING, CANCELED, ATTENDED, REFOUNDED
33         'payment_status',
34         'payment_code',
35         'payment_date',
36
37         // PENDING, CANCELED, ATTENDED, REFOUNDED
38         'delivery_status',
39
40         // Comentarios adicionales
41         "comments",
42
43         // Montos de pedido
44         'sub_total',
45         'extra_fee',
46         'coupon_discount',
47         'delivery_amount'
```

Figura\_ 109: Vista de Gestión de ventas

```
index.blade.php X
resources > views > back-office > admin > order > index.blade.php > ...
1 @extends('layouts.back-office.admin', ["title" => $title])
2
3 @section('content')
4 <div class="row">
5     @php
6         $builder = new \App\Utils\ReactCrudSettingsBuilder();
7
8         $order_code = new \App\Utils\ReactCrudField('order_code');
9         $order_code->title('PEDIDO #')->required(true)->width(12);
10        $builder->addField($order_code);
11
12        $client_name = new \App\Utils\ReactCrudField('client_name');
13        $client_name->title('Cliente');
14        $builder->addField($client_name);
15
16        $payment_status = new \App\Utils\ReactCrudField('payment_status');
17        $payment_status->title('Pago')->type('map_color', [
18            ['id' => 'PENDING', 'value' => 'Pendiente', 'color' => "#fd7e14"],
19            ['id' => 'ATTENDED', 'value' => 'Atendido', 'color' => "#4caf50"],
20            ['id' => 'CANCELED', 'value' => 'Cancelado', 'color' => "#ad2a20"],
21            ['id' => 'INCOMPLETE', 'value' => 'Incompleto', 'color' => "#FF5722"]
22        ]);
23        $builder->addField($payment_status);
24
25        $delivery_status = new \App\Utils\ReactCrudField('delivery_status');
26        $delivery_status->title('Envío')->type('map_color', [
27            ['id' => 'PENDING', 'value' => 'Pendiente', 'color' => "#fd7e14"],
28            ['id' => 'ATTENDED', 'value' => 'Atendido', 'color' => "#4caf50"],
29            ['id' => 'CANCELED', 'value' => 'Cancelado', 'color' => "#ad2a20"],
30        ]);
31        $builder->addField($delivery_status);
32
33        $order_status = new \App\Utils\ReactCrudField('order_status');
34        $order_status->title('Pedido')->type('map_color', [
35            ['id' => 'PENDING', 'value' => 'Pendiente', 'color' => "#fd7e14"],
36            ['id' => 'ATTENDED', 'value' => 'Atendido', 'color' => "#4caf50"],
37            ['id' => 'CANCELED', 'value' => 'Cancelado', 'color' => "#ad2a20"],
38        ]);
39        $builder->addField($order_status);
40
41        $created_at = new \App\Utils\ReactCrudField('created_at');
42        $created_at->title('Fecha')->type('date');
43        $builder->addField($created_at);
44
45        $actions = [];
46        $actions["custom"] = [];
47
Cerrar (36W)
Lín. 1, col. 1 Espacios: 4 UTF-8 LF Blade
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 110: Implementación de Gestión de ventas**

PEDIDO #	CLIENTE	PAGO	ENVÍO	PEDIDO	FECHA	ACCIONES
MKT-22021	Jhon Doe	●	●	●	08/10/2021	[+][-][✓][✗]
MKT-12021	Marzio Perez	●	●	●	03/10/2021	[+][-][✓][✗]

**RF15: El sistema permite gestionar las herramientas multimedia como imágenes para poder asignarle a los banners, productos, categorías u otros elementos de la plataforma.**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 111: Prototipo de Importar archivos multimedia



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.



Figura\_ 112: Controlador de Importar archivos multimedia

```
MultimediaResourceController.php x
app > Http > Controllers > Backoffice > Admin > MultimediaResourceController.php > ...
1 <?php
2
3 namespace App\Http\Controllers\Backoffice\Admin;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\MultimediaResource;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\DB;
9 use Illuminate\Support\Facades\Storage;
10
11 class MultimediaResourceController extends Controller {
12
13     public function index() {
14         $title = "Multimedia";
15         return view('back-office.admin.multimedia', compact('title'));
16     }
17
18     public function lists(){
19         $model = DB::select('call all_multimedia_resources');
20         return response()->json(['status'=>'ok','data' => $model]);
21     }
22
23     public function upload(Request $request){
24         $file = $request->file('file');
25         $origin_name_file = $file->getClientOriginalName();
26         $origin_name_file = str_replace(" ", "-", $origin_name_file);
27         $origin_name_file = str_replace("&quot;", "", $origin_name_file);
28         $origin_name_file = str_replace("&quot;", "", $origin_name_file);
29
30         $origin_size_file = $file->getSize();
31         $name = '';
32         $parts_name = explode("&quot;.", $origin_name_file);
33         if (count($parts_name) > 0){
34             $name = $parts_name[0];
35         }
36
37         $exists_file_name = MultimediaResource::where("file_name", $name)->get()->first();
38         if($exists_file_name) {
39             $name = $name . "-" . time();
40         }
41
42         $ext = $file->extension();
43         $origin_name_file = $name . "." . $ext;
44     }
45 }
```

**Figura\_ 113: Modelo de Importar archivos multimedia**

```
MultimediaResource.php ×
app > Models > MultimediaResource.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class MultimediaResource extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13     protected $dates = ['deleted_at'];
14
15     protected $fillable = [
16         'id',
17         'file_path',
18         'file_name',
19         'file_size',
20         'file_url',
21         'extension',
22         'store_id'
23     ];
24
25 }
26
```

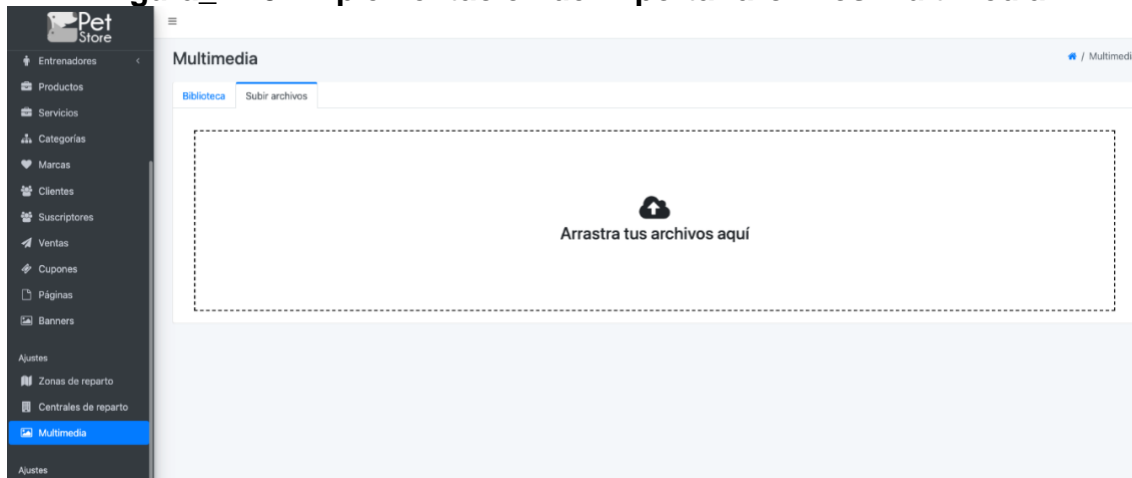
**Figura\_ 114: Vista de Importar archivos multimedia**

```
multimedia.blade.php — marketplace-pets
multimedia.blade.php ×
resources > views > back-office > admin > multimedia.blade.php > ...
1  @extends('layouts.back-office.admin', ["title" => $title])
2
3  @section('content')
4
5  <div class="multimedia"
6      data-multimedia-list-url="{{route('admin.multimedia.lists')}}"
7      data-upload-url="{{route('admin.multimedia.upload')}}"
8      data-delete-url="{{route('admin.multimedia.delete')}}"></div>
9
10 @endsection
11
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el product owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 115: Implementación de Importar archivos multimedia**



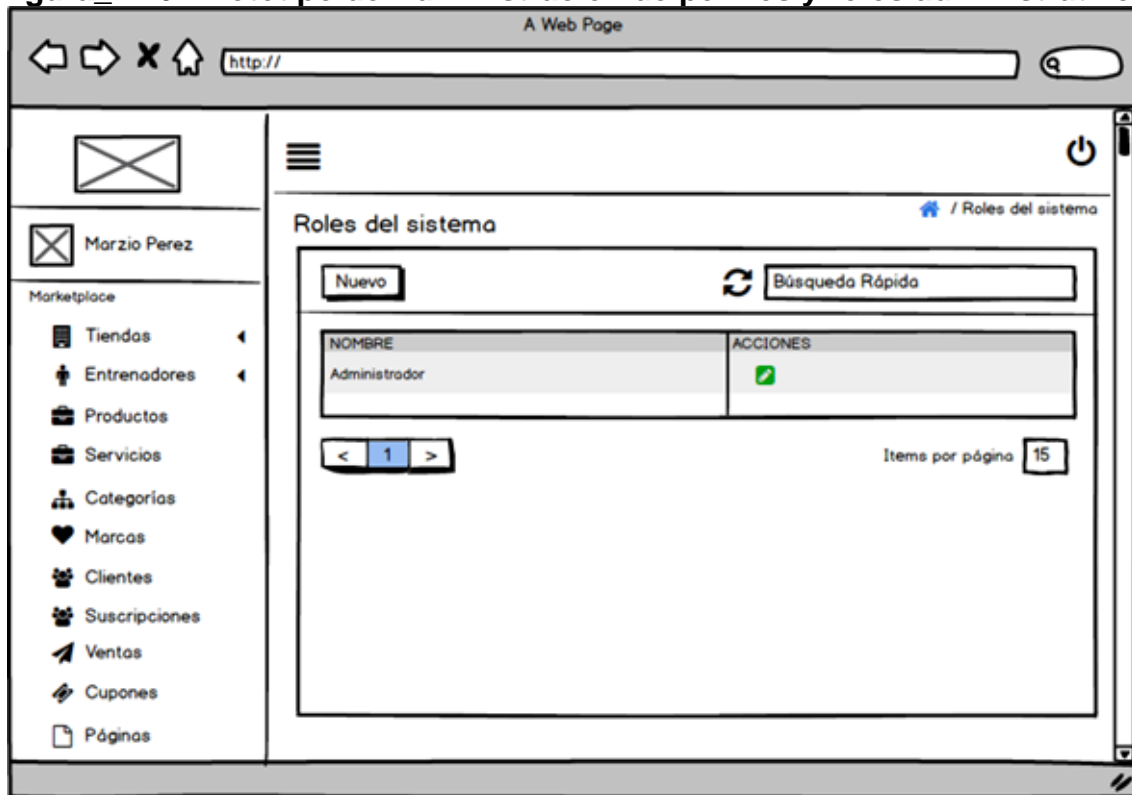
**RF16: El sistema permite gestionar los perfiles y roles de la plataforma con el fin de poder delimitar qué módulos podrán observar dichos usuarios**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 116: Prototipo de Administración de perfiles y roles administrativos



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 117: Controlador de Administración de perfiles y roles administrativos

```
RoleController.php x
app > Http > Controllers > Backoffice > Admin > Security > RoleController.php > ...
1 <?php
2
3 namespace App\Http\Controllers\Backoffice\Admin\Security;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Module;
7 use App\Models\Role;
8 use App\Models\RoleModule;
9 use App\Models\RoleModuleMenu;
10 use Illuminate\Http\Request;
11
12 class RoleController extends Controller {
13
14     public function index() {
15         $title = "Roles del sistema";
16         $modules = Module::with('menus')->where('model', 'ADMIN')->get();
17         return view('back-office.admin.security.role', compact('title', 'modules'));
18     }
19
20     public function lists() {
21         $model = Role::with('modules.module.menus')->where('model', 'ADMIN')->get();
22         return response()->json($model);
23     }
24
25     public function create(Request $request) {
26         $data = $request->all();
27         $role = Role::create([
28             "name" => $data["name"],
29             "model" => "ADMIN"
30         ]);
31         if (count($data["modules"]) > 0) {
32             foreach ($data["modules"] as $module) {
33                 if ($module["checked"]){
34                     $role_module = RoleModule::create([
35                         'role_id' => $role->id,
36                         'module_id' => $module["id"]
37                     ]);
38                     if (count($module["menus"]) > 0) {
39                         foreach ($module["menus"] as $menu) {
40                             if ($menu["checked"]) {
41                                 RoleModuleMenu::create([
42                                     'module_menu_id' => $menu["id"],
43                                     'role_module_id' => $role_module->id
44                                 ]);
45                             }
46                         }
47                     }
48                 }
49             }
50         }
51     }
52 }
```

Figura\_ 118: Modelo de Administración de perfiles y roles administrativos

```
Role.php x
app > Models > Role.php > ...
1 |<?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Role extends Model {
10     use HasFactory;
11     use SoftDeletes;
12
13     protected $dates = ["deleted_at"];
14
15     protected $fillable = [
16         'id',
17         'name',
18         'model',
19         'store_id',
20         'deleted'
21     ];
22
23     public function modules(){
24         return $this->hasMany('App\Models\RoleModule', 'role_id', 'id');
25     }
26
27 }
28
```

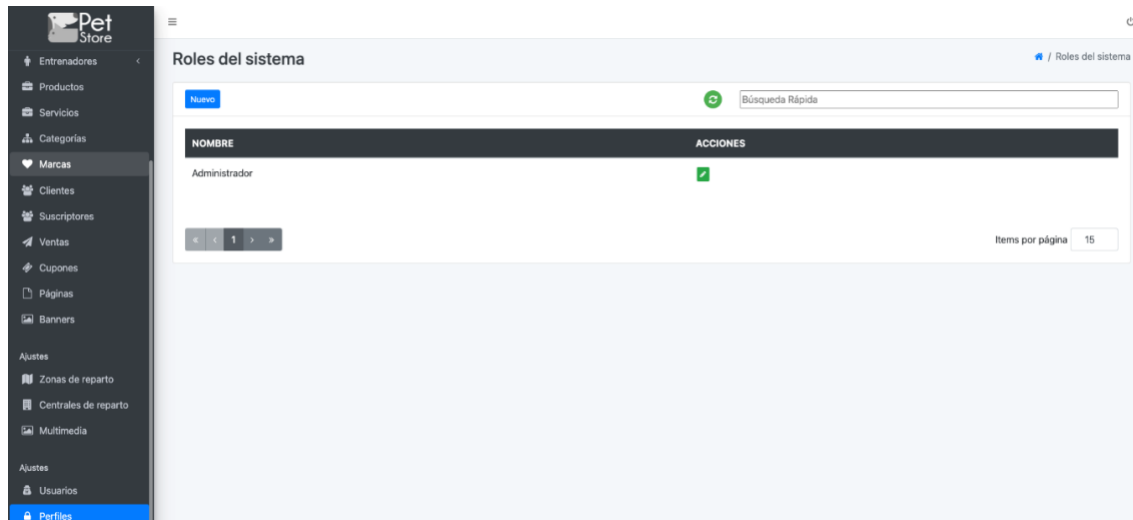
Figura\_ 119: Vista de Administración de perfiles y roles administrativos

```
Role.php role.blade.php x
resources > views > back-office > admin > security > role.blade.php > ...
1 |@extends('layouts.back-office.admin', ["title" => $title])
2
3 @section('content')
4
5     @php
6         use App\Utils\ReactCrudField;
7         use App\Utils\ReactCrudSettingsBuilder;
8
9         $builder = new ReactCrudSettingsBuilder();
10
11         $nameField = new ReactCrudField('name');
12         $nameField->title('Nombre')->required(true)->width(12);
13         $builder->addField($nameField);
14
15         $modulesField = new ReactCrudField('current_modules');
16         $modulesField->title('Módulos')->type('modules')->show(false)->fillable(false)->width(12);
17         $builder->addField($modulesField);
18
19         $deletedField = new ReactCrudField('deleted');
20         $deletedField->title('deleted')->type('deleted');
21         $builder->addField($deletedField);
22
23         $actions = [];
24         $actions["custom"] = [];
25         $actions["create-role"] = ['url' => route('admin.security.profiles.create'), 'modules' => json_encode(['role_id' => $role_id]);
26         $actions["update-role"] = ['url' => route('admin.security.profiles.update'), 'modules' => json_encode(['role_id' => $role_id]);
27         $actions["delete"] = ['url' => route('admin.security.profiles.delete')];
28         $builder->setActions($actions);
29     @endphp
30     <div class="content custom-scrollbar">
31         <div id="{{\App\Utils\ReactComponents::LARAVEL_CRUD_COMPONENT}}" data-url="{{route('admin.security.profiles.create')}}">
32     </div>
33
34 @endsection
35
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 120: Implementación de Administración de perfiles y roles administrativos**



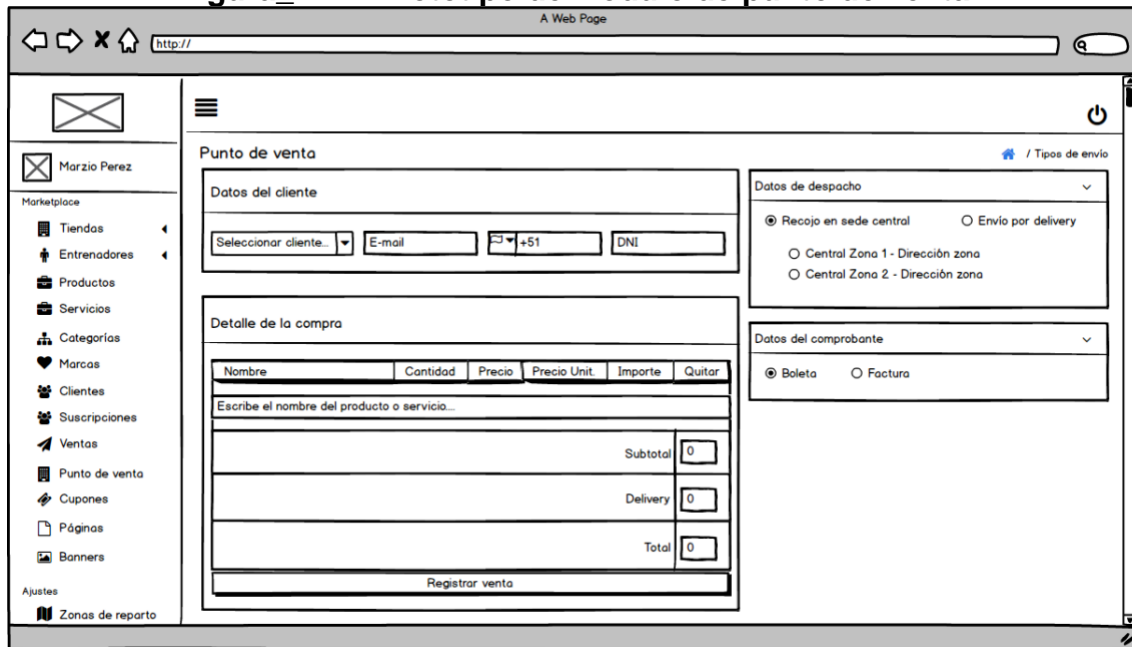
**RF17: El sistema permite registrar pedidos a través de un punto de venta con el fin de poder tener un control de los pedidos que vienen por otros canales de atención.**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 121: Prototipo de Módulo de punto de venta**



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

**Figura\_ 122: Controlador de Módulo de punto de venta**

```

PointOfSaleController.php M X
app > Http > Controllers > Backoffice > Admin > PointOfSaleController.php > PointOfSaleController > process
23
24 public function index() {
25     $title = "Punto de venta";
26     $clients = Client::where("status", "CONFIRM")->get();
27     $delivery_centrales = DeliveryCentral::all();
28     $delivery_zones = DeliveryZone::with('bounds')->get();
29     return view('back-office.admin.point-of-sale', compact('title', 'clients', 'delivery_centrales', 'delivery_zones'));
30 }
31
32 public function process(Request $request) {
33     $data = $request->all();
34     $market_util = new MarketUtil();
35
36     $order_code = Order::select(["id"])->orderBy("id", "DESC")->first();
37     if ($order_code){
38         $soc = env("ORDER_PREFIX_CODE") . "-" . ($order_code->id + 1) . date("Ym");
39     }else{
40         $soc = env("ORDER_PREFIX_CODE") . "-" . 1 . date("Ym");
41     }
42
43     $coupon_id = null;
44     if($data["coupon_code"] != "" ) {
45         $coupon = Coupon::where("code", $data["coupon_code"])->get()->first();
46         if($coupon) {
47             $coupon_id = $coupon["id"];
48         }
49     }
50
51     // Order de compra para el Marketplace
52     $order = Order::create([
53         'order_code' => $soc,
54

```



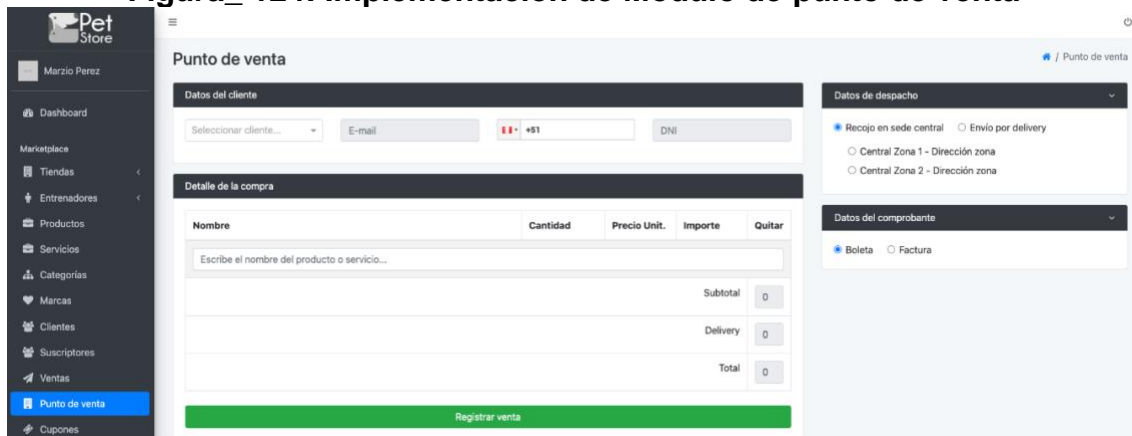
Figura\_ 123: Vista de Módulo de punto de venta

```
JS PointOfSaleComponent.js M X
resources > assets > back-office > js > components > point_of_sale > JS PointOfSaleComponent.js > PointOfSaleComponent > render
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import moment from 'moment';
4 import Select from 'react-select';
5 import 'react-select/dist/react-select.css';
6 import PhoneInput from 'react-phone-input-2';
7
8 export default class PointOfSaleComponent extends React.Component {
9
10  constructor(props) {
11    super(props);
12    this.clients = JSON.parse(this.props.clients);
13    this.delivery_centrais = JSON.parse(this.props.delivery_centrais);
14    this.delivery_zones = JSON.parse(this.props.delivery_zones);
15    this.search_product_url = this.props.search_product_url;
16
17    this.polygons = [];
18    if(this.delivery_zones.length > 0) {
19      this.delivery_zones.map((element, i) => {
20        let obj = {color: element["color"]};
21        let bounds = [];
22        if(element["bounds"].length > 0) {
23          element["bounds"].map((bound, b) => {
24            let obj_bound = {
25              lat: bound["latitude"],
26              lng: bound["longitude"]
27            };
28            bounds.push(obj_bound);
29          });
30        }
31        obj["bounds"] = bounds;
```

## IMPLEMENTACIÓN

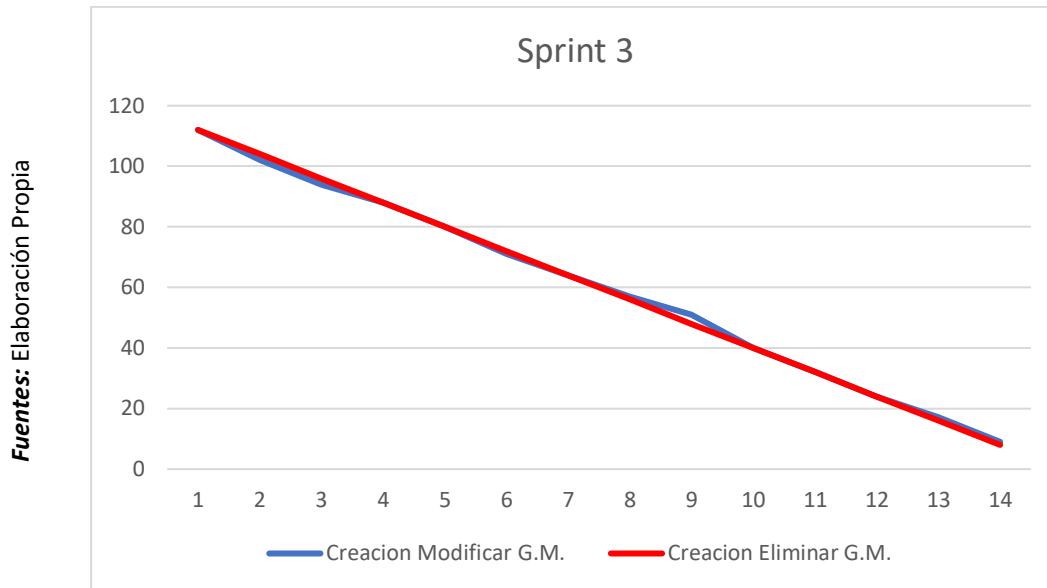
A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

Figura\_ 124: Implementación de Módulo de punto de venta



## BURNDOWN DEL SPRINT N° 3

Figura\_ 125: Burndown del Sprint 3



### Burndown Sprint 3

En la figura vista, se muestra el proceso de desarrollo de cada sprint la cual está representada a través de un gráfico, como podemos observar tenemos dos líneas la cual la línea roja representa al tiempo ideal que se define para el desarrollo del sprint, de igual manera la línea azul representa el tiempo real que tomo el desarrollo del sprint. Ya teniendo bien definido las representaciones tenemos que concluir que si la línea azul se encuentra por debajo de la roja significa que se presentó un adelanto, de lo contrario si la línea azul está por encima es decir que hubo un retraso.

### ACTA DE REUNION DEL SPRINT N° 3

Siendo las 3 pm del día 25 de junio del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.


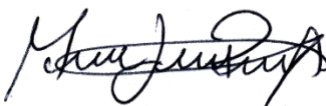


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El encargado de la Empresa AC Brands & Consulting E.I.R.L., Isaac Abensur Vargas concluye la reunión con los Sres. Marzio Perez y Nicolás Segovia aclarando los últimos puntos sobre el sprint.

Para determinar las formas de trabajo y los requerimientos respectivos para esta interacción, también la fecha de entrega y firmar el acta para dar por terminado la reunión.

Los participantes manifestaron su aprobación respecto a todos los puntos definidos en la reunión, que fueron de ayuda a la planificación del Sprint, manifestando su compromiso para la entrega de los resultados en las fechas pactadas.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

### ACTA DE ENTREGA DEL SPRINT N° 3


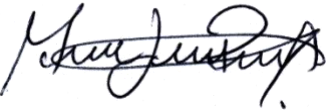


Siendo la 1 pm del día 10 de julio del 2021 se reúne en la empresa AC Brands & Consulting E.I.R.L.

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

Los Sres. Marzio Perez y Nicolás Segovia, especifican cada uno de los requerimientos desarrollados, también presentan las interfaces elaboradas y brindadas por el producto Owner.

Luego de ser verificadas las presentaciones realizadas por los Srs. Marzio Perez y Nicolás Segovia, se decidió de manera unánime, dar por aprobado el término del Sprint respectivo, del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Los presentes manifestarán su aprobación al informe presentado por los Sres. Marzio Perez y Nicolás Segovia del Sprint N° 3 terminado del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

### RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 3

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	AC Brands & Consulting E.I.R.L.
<b>Proyecto</b>	“Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Información de la reunión:

<b>Lugar</b>	AC Brands & Consulting E.I.R.L.
<b>Fecha</b>	10/07/2021
<b>Número de iteración / Sprint</b>	Sprint 3
<b>Personas Convocadas a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza
<b>Persona que asistieron a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
<p>Se desarrolló la gestión de centrales de reparto.</p> <p>Se logró desarrollar la gestión de ventas y el módulo de punto de venta.</p> <p>Se desarrolló exitosamente la importación de archivos multimedia.</p> <p>Además, se desarrolló la administración de perfiles y roles administrativos.</p>	<p>Hubo problemas al desarrollar la importación de archivos.</p>

## PLANIFICACIÓN DEL SPRINT N°4

Siendo las 4 pm del día 10 de julio del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.


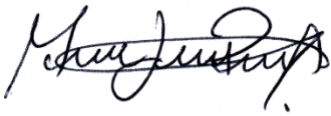


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El gerente de la Empresa AC Brands & Consulting E.I.R.L., efectúa la presentación de los requerimientos, de igual manera manifiesta los requerimientos que tienen mayor prioridad.

Realizado el análisis de los requerimientos presentados por el gerente de AC Brands & Consulting E.I.R.L., el señor Juan Galarza aclara dudas existentes y luego se compromete al cumplimiento de todos los requerimientos propuestas en el Sprint 4.

Los participantes de la reunión manifiestan su aprobación con respecto a la presentación del Sprint 4, así mismo especificando que la fecha de entrega será el día 20 de julio del 2021.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## EJECUCIÓN DEL SPRINT 4

Lo primero que se debe realizar antes del diseño, es conocer y comprender el funcionamiento del sistema, en otras palabras, analizar lo que realmente se necesita, lo cual será elaborado según las historias de usuario.

**RF18: El sistema permite ingresar con los datos que el administrador me ha proporcionado con el fin de tener acceso a los diferentes módulos que me correspondan.**

### DISEÑO

#### PROTOTIPO

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 126: Prototipo de Ingresar como comercio**



### DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 127: Controlador de Ingresar como comercio

```
AdminLoginController.php 2, M X
app > Http > Controllers > Backoffice > Auth > AdminLoginController.php > AdminLoginController
1 <?php
2
3 namespace App\Http\Controllers\Backoffice\Auth;
4
5 use App\Http\Controllers\Controller;
6 use Illuminate\Foundation\Auth\AuthenticatesUsers;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Auth;
9
10 class AdminLoginController extends Controller {
11
12     use AuthenticatesUsers;
13     protected $redirectTo = "/admin/dashboard";
14
15     public function showLoginForm() {
16         if(Auth::guard('admin')->check()){
17             return redirect()->to(route('admin.dashboard'));
18         }
19         return view('back-office.auth.admin-login');
20     }
21
22     public function login(Request $request) {
23         $this->validator($request);
24
25         if(Auth::guard('admin')->attempt($request->only('email','password'),$request->filled('remember'))){
26             return redirect()->intended(route('admin.dashboard'));
27         }
28         return $this->loginFailed();
29     }
30
31     public function logout() {
32         Auth::guard('admin')->logout();
33         return redirect()->route('admin.login');
34     }
35
36     private function validator(Request $request) {
37         $rules = [
38             'email' => 'required|email|exists:admins|min:5|max:191',
39             'password' => 'required|string|min:4|max:255',
40         ];
41
42         $messages = [
43             'email.exists' => 'These credentials do not match our records.',
44         ];
45     }
46 }
```



Figura\_ 128: Modelo de Ingresar como comercio

```
app > Models > Admin.php > Admin
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Foundation\Auth\User as Authenticatable;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8  use Illuminate\Notifications\Notifiable;
9
10 class Admin extends Authenticatable {
11
12     use HasFactory;
13     use Notifiable;
14     use SoftDeletes;
15
16     protected $dates = ['deleted_at'];
17
18     protected $guard = 'admin';
19
20     protected $fillable = [
21         'name',
22         'type',
23         'email',
24         'password',
25         'deleted'
26     ];
27
28     protected $appends = ["roles_text"];
29
30     protected $hidden = [
31         'password', 'remember_token',
32     ];
33
34     public function roles(){
35         return $this->hasMany('App\Models\RoleAdmin', 'admin_id', 'id');
36     }
37
38     public function getRolesTextAttribute(){
39         $roles = $this->roles;
40         $roles_array = [];
41         foreach ($roles as $role){
42             $roles_array[] = $role["role_id"];
43         }
44         return implode(", ", $roles_array);
45     }
46 }
47
```

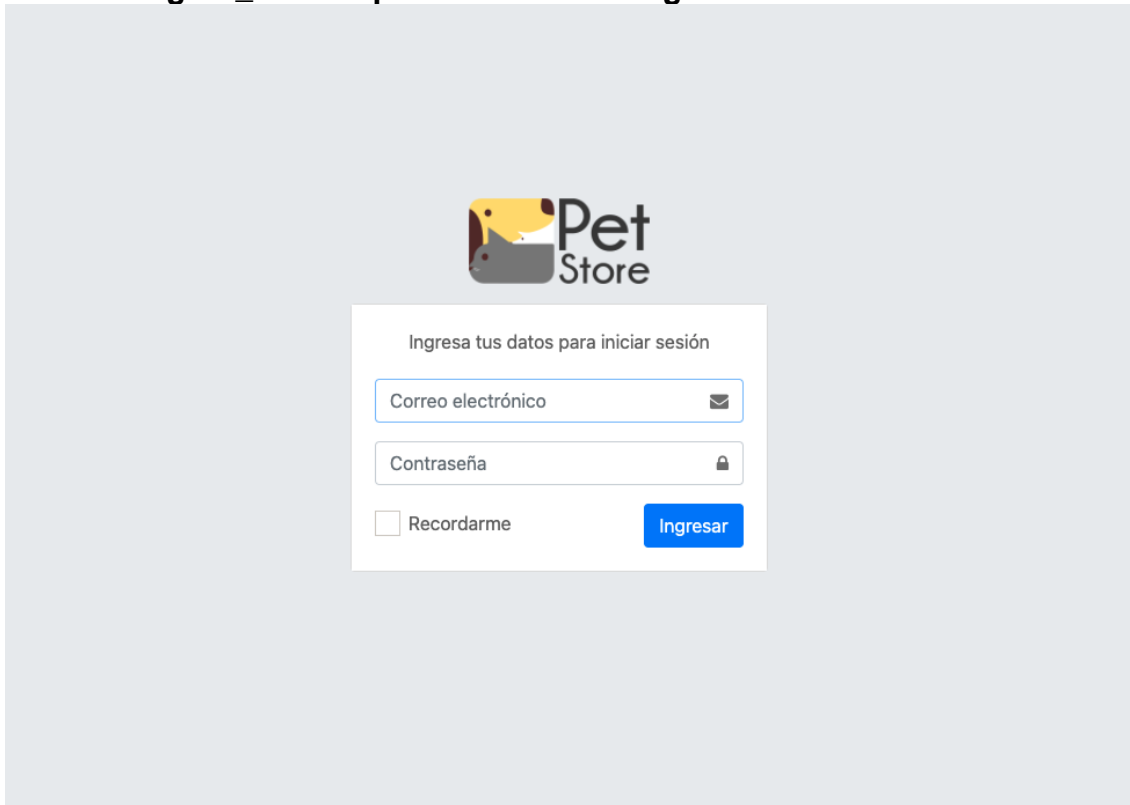
Figura\_ 129: Vista de Ingresar como comercio

```
admin-login.blade.php x
resources > views > back-office > auth > admin-login.blade.php > ...
1 @extends('layouts.back-office.login')
2
3 @section('content')
4
5 <div class="card">
6   <div class="card-body login-card-body">
7     <p class="login-box-msg">Ingresa tus datos para iniciar sesión</p>
8     <form method="POST" action="{{ route('admin.process-login') }}">
9       @csrf
10      <div class="input-group mb-3">
11        <input type="email" name="email" class="form-control @error('email') is-invalid @enderror" placeholder="Correo electr
12        <div class="input-group-append">
13          <div class="input-group-text">
14            <span class="fa fa-envelope"></span>
15          </div>
16        </div>
17      </div>
18
19      <div class="input-group mb-3">
20        <input type="password" name="password" class="form-control @error('password') is-invalid @enderror" placeholder="Cont
21        <div class="input-group-append">
22          <div class="input-group-text">
23            <span class="fa fa-lock"></span>
24          </div>
25        </div>
26      </div>
27
28      <div class="row">
29        <div class="col-8">
30          <div class="icheck-primary">
31            <input type="checkbox" id="remember" name="remember" {{ old('remember') ? 'checked' : '' }}>
32            <label for="remember">Recordarme</label>
33          </div>
34        </div>
35        <div class="col-4">
36          <button type="submit" class="btn btn-primary btn-block">Ingresar</button>
37        </div>
38      </div>
39    </form>
40  </div>
41 </div>
42 @endsection
43
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 130: Implementación de Ingresar como comercio**



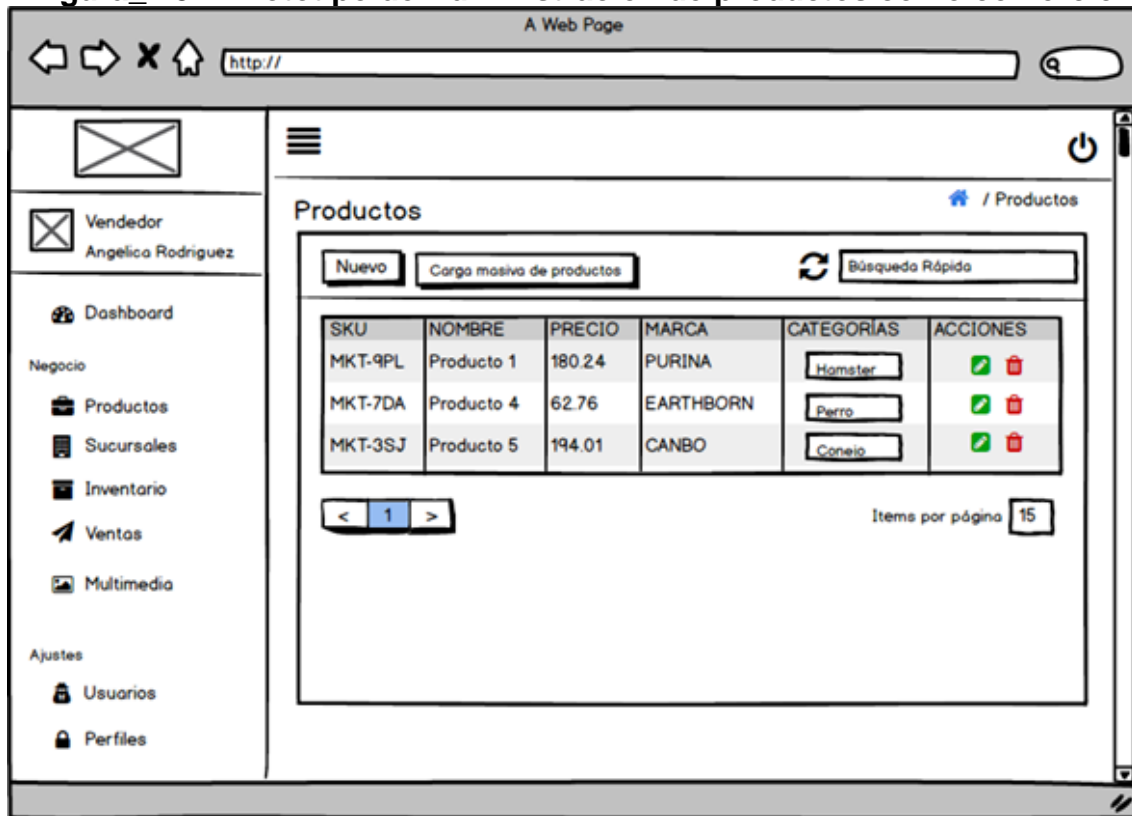
**RF19: El sistema permite gestionar los productos con el fin de que se muestran en la plataforma y los clientes los puedan adquirir, desde el usuario comercio.**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 131: Prototipo de Administración de productos como comercio



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura 132: Controlador de Administración de productos como comercio

```
ProductController.php M X
app > Http > Controllers > Backoffice > Admin > ProductController.php > ...
15 use App\Utils\SlugUtil;
16 use Exception;
17 use Illuminate\Http\Request;
18 use Illuminate\Support\Facades\DB;
19
20 class ProductController extends Controller {
21
22     public function index() {
23         $title = "Productos";
24
25         $categories = array_map(
26             function($item){
27                 return [
28                     "id" => $item['id'],
29                     "value" => $item['internal_name']
30                 ];
31             }, Category::where('type', 'products')->get()->toArray()
32         );
33
34         $brands = array_map(
35             function($item){
36                 return [
37                     "id" => $item['id'],
38                     "value" => $item['name']
39                 ];
40             }, Brand::all()->toArray()
41         );
42
43         $stores = array_map(
44             function($item){
45                 return [
46                     "id" => $item['id'],
47                     "value" => $item['name']
48                 ];
49             }, Store::where("is_trainer", false)->get()->toArray()
50         );
51         return view('back-office.admin.products.index', compact('title', 'categories', 'brands', 'stores'));
52     }
53
54     public function lists(){
55         $model = DB::select('call_all_products');
56         return response()->json($model);
57     }
58
59     public function variations(Request $request) {
60         $data = $request->all();
61         $products = Product::where('id', $data['id'])->get();
62     }
63 }
```

**Figura\_ 133: Modelo de Administración de productos como comercio**

```
Product.php X
app > Models > Product.php > Product > $fillable
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Product extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'sku_code',
20         'description',
21         'summary',
22         'featured_image_id',
23
24         'categories',
25
26         'is_featured',
27
28         'price',
29         'iva',
30         'price_with_iva',
31
32         'offer_price',
33         'offer_iva',
34         'offer_price_with_iva',
35
36         'is_service',
37         'service_attentions_per_hour',
38         'rate',
39
40         'brand_id',
41         'brand_name',
42
43         'slug_route_id',
44         'url',
45
46         'store_id',
```

**Figura\_ 134: Vista de Administración de productos como comercio**

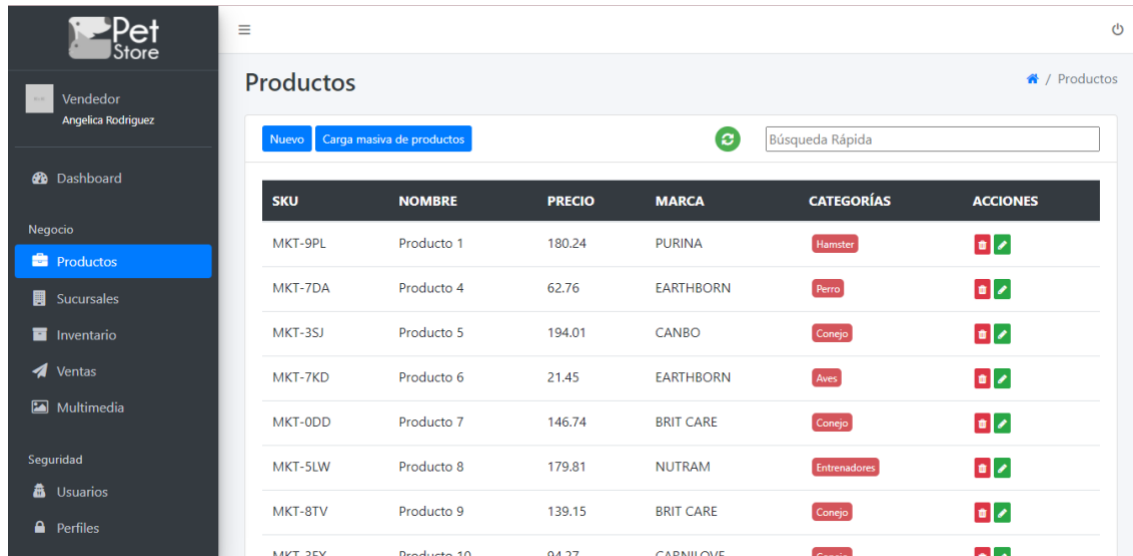
```
index.blade.php
~/Desarrollo/Proyectos/marketplace-pets/resources/Views/back-office/admin/products/index.blade.php
products > index.blade.php > ...

2
3 @section('content')
4
5 @php
6     use App\Utils\ReactCrudField;
7     use App\Utils\ReactCrudSettingsBuilder;
8
9     $builder = new ReactCrudSettingsBuilder();
10
11     $sku_code = new ReactCrudField('sku_code');
12     $sku_code->title('SKU')->required(true)->width(2);
13     $builder->addField($sku_code);
14
15     $name = new ReactCrudField('name');
16     $name->title('Nombre')->required(true)->width(10);
17     $builder->addField($name);
18
19     $price = new ReactCrudField('price');
20     $price->title('Precio')->required(true)->width(2);
21     $builder->addField($price);
22
23     $offer_price = new ReactCrudField('offer_price');
24     $offer_price->title('Precio de oferta')->show(false)->width(2);
25     $builder->addField($offer_price);
26
27     $brand_id = new ReactCrudField('brand_id');
28     $brand_id->title('Marca')->type('map', $brands)->width(3);
29     $builder->addField($brand_id);
30
31     $store_id = new ReactCrudField('store_id');
32     $store_id->title('Tienda')->type('map', $stores)->required(true)->width(3);
33     $builder->addField($store_id);
34
35     $is_featured = new ReactCrudField('is_featured');
36     $is_featured->title('Producto destacado')->type('map', [
37         ['id' => 1, 'value' => 'Si'],
38         ['id' => 0, 'value' => 'No']
39     ]->default(0)->show(false)->width(2);
40     $builder->addField($is_featured);
41
42     $categories_field = new ReactCrudField('categories');
43     $categories_field->title('Categorías')->type('json', $categories)->width(12);
44     $builder->addField($categories_field);
45
46     $description = new \App\Utils\ReactCrudField('description');
47     $description->title('Descripción')->type('editor')->show(false)->width(12);
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 135: Implementación de Administración de productos como comercio**



**RF20: El sistema permite visualizar las ventas que he generado con el fin de poder verificar el detalle de los pedidos que he recibido, por parte del usuario comercio.**

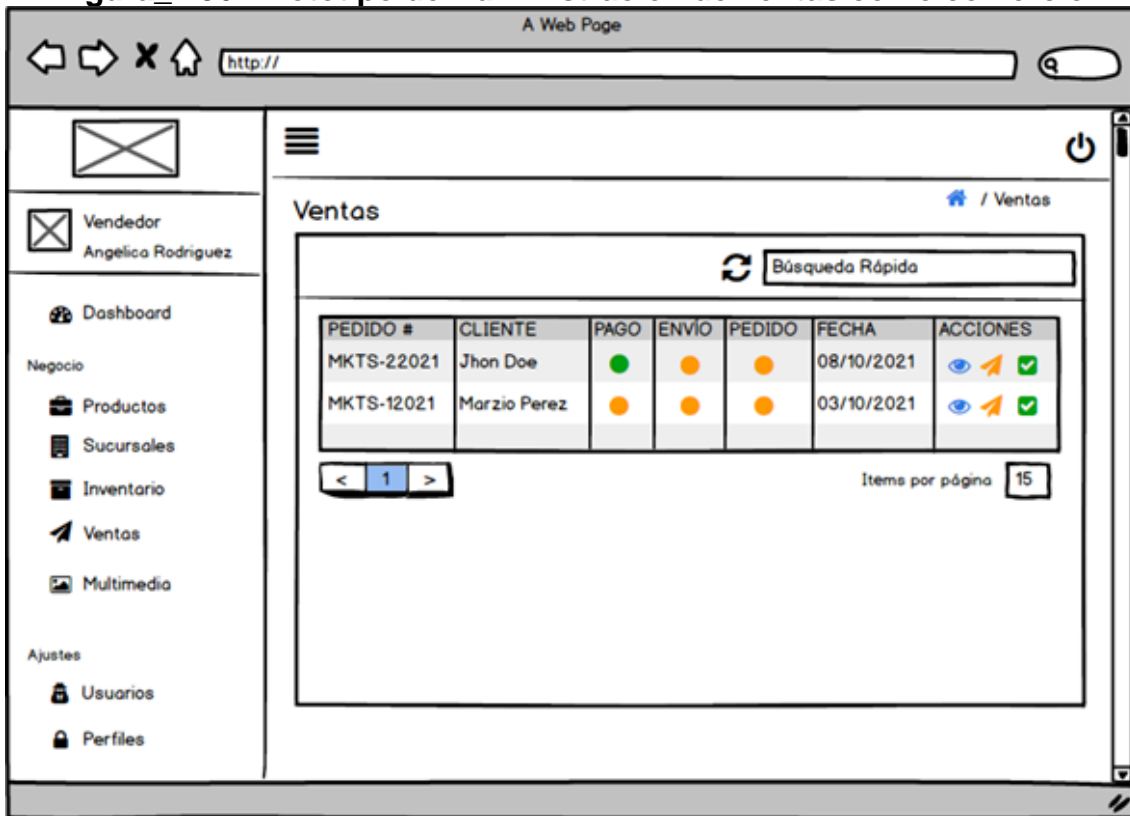
## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.



Figura\_ 136: Prototipo de Administración de ventas como comercio



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 137: Controlador de Administración de ventas como comercio

```
OrderController.php
~/Desarrollo/Proyectos/marketplace-pets/app/
Http/Controllers/Backoffice/Admin/
OrderController.php
Admin > OrderController.php > OrderController > canceled

18 use Illuminate\Http\Request;
19 use Illuminate\Support\Facades\DB;
20
21 class OrderController extends Controller {
22
23     public function index() {
24         $title = "Ventas";
25         return view('back-office.admin.order.index', compact('title'));
26     }
27
28     public function lists(){
29         $model = DB::select('call all_orders');
30         return response()->json($model);
31     }
32
33     public function detail(Request $request) {
34         $title = "Detalle de venta";
35         $data = $request->all();
36         $code = $data["order_code"];
37
38         $order = Order::with('detail')->where("order_code", $code)->get()->first();
39         return view('back-office.admin.order.detail', compact('title', 'order'));
40     }
41
42     public function attended_payment(Request $request) {
43         $data = $request->all();
44         $order = Order::find($data["id"]);
45         $order->update([
46             "payment_status" => "ATTENDED",
47             "payment_date" => date('Y-m-d H:i:s')
48         ]);
49         if($order["payment_type"] === "BANK-TRANSFER" || $order["payment_type"] === "YAPE"){
50             $items = OrderDetail::where("order_id", $order["id"])->get();
51             foreach ($items as $item){
52                 $inventory = Inventory::where("product_id", $item["product_id"])->get()->first();
53                 if($inventory){
54                     $inventory->update(["quantity" => $inventory["quantity"] - $item["quantity"]]);
55                     $inventory_variation = InventoryVariation::where("variation_id", $item["variation_id"])->where("b
56                     if($inventory_variation){
57                         $inventory_variation->update(["quantity" => $inventory_variation["quantity"] - $item["quantit
58                         InventoryMovement::create([
59                             // I: Ingreso | E: Egreso
60                             'movement_type' => "E",
61                             'quantity' => $item["quantity"],
62                             'inventory_id' => $inventory["id"],
63                             "product_name" => $item["product_name"],
```

Figura\_ 138: Modelo de Administración de ventas como comercio

```
Order.php x
app > Models > Order.php > ...
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Order extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'order_code',
19
20         // ONLINE | OFFLINE
21         'type',
22
23         // PENDING, CANCELED, ATTENDED, REFOUNDED
24         'order_status',
25         'order_attended_date',
26         'order_closed_date',
27
28         // CREDIT-CARD | BANK-TRANSFER | CASH
29         'payment_type',
30         'bank_transfer_file',
31
32         // PENDING, CANCELED, ATTENDED, REFOUNDED
33         'payment_status',
34         'payment_code',
35         'payment_date',
36
37         // PENDING, CANCELED, ATTENDED, REFOUNDED
38         'delivery_status',
39
40         // Comentarios adicionales
41         "comments",
42
43         // Montos de pedido
44         'sub_total',
45         'extra_fee',
46         'coupon_discount',
47         'delivery_amount'
```

Figura 139: Vista de Administración de ventas como comercio

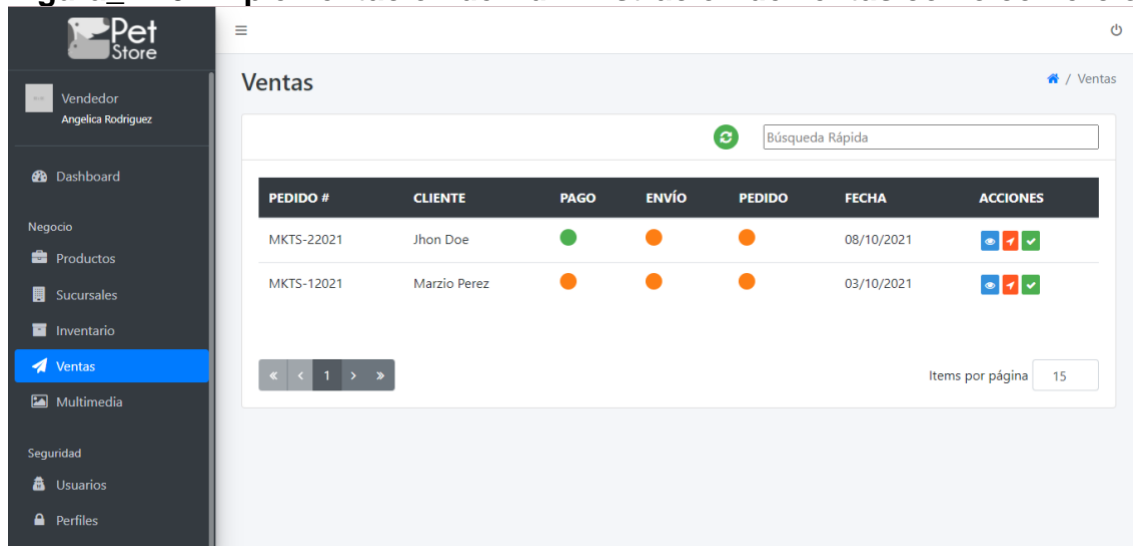
```
index.blade.php X
resources > views > back-office > admin > order > index.blade.php > ...
1 @extends('layouts.back-office.admin', ['title' => $title])
2
3 @section('content')
4 <div class="row">
5     @php
6         $builder = new \App\Utils\ReactCrudSettingsBuilder();
7
8         $order_code = new \App\Utils\ReactCrudField('order_code');
9         $order_code->title('PEDIDO #')->required(true)->width(12);
10        $builder->addField($order_code);
11
12        $client_name = new \App\Utils\ReactCrudField('client_name');
13        $client_name->title('Cliente');
14        $builder->addField($client_name);
15
16        $payment_status = new \App\Utils\ReactCrudField('payment_status');
17        $payment_status->title('Pago')->type('map_color', [
18            ['id' => 'PENDING', 'value' => 'Pendiente', 'color' => "#fd7e14"],
19            ['id' => 'ATTENDED', 'value' => 'Atendido', 'color' => "#4caf50"],
20            ['id' => 'CANCELED', 'value' => 'Cancelado', 'color' => "#ad2a20"],
21            ['id' => 'INCOMPLETE', 'value' => 'Incompleto', 'color' => "#FF5722"]
22        ]);
23        $builder->addField($payment_status);
24
25        $delivery_status = new \App\Utils\ReactCrudField('delivery_status');
26        $delivery_status->title('Envio')->type('map_color', [
27            ['id' => 'PENDING', 'value' => 'Pendiente', 'color' => "#fd7e14"],
28            ['id' => 'ATTENDED', 'value' => 'Atendido', 'color' => "#4caf50"],
29            ['id' => 'CANCELED', 'value' => 'Cancelado', 'color' => "#ad2a20"],
30        ]);
31        $builder->addField($delivery_status);
32
33        $order_status = new \App\Utils\ReactCrudField('order_status');
34        $order_status->title('Pedido')->type('map_color', [
35            ['id' => 'PENDING', 'value' => 'Pendiente', 'color' => "#fd7e14"],
36            ['id' => 'ATTENDED', 'value' => 'Atendido', 'color' => "#4caf50"],
37            ['id' => 'CANCELED', 'value' => 'Cancelado', 'color' => "#ad2a20"],
38        ]);
39        $builder->addField($order_status);
40
41        $created_at = new \App\Utils\ReactCrudField('created_at');
42        $created_at->title('Fecha')->type('date');
43        $builder->addField($created_at);
44
45        $actions = [];
46        $actions["custom"] = [];
47
Cerrar (#W)

```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 140: Implementación de Administración de ventas como comercio**



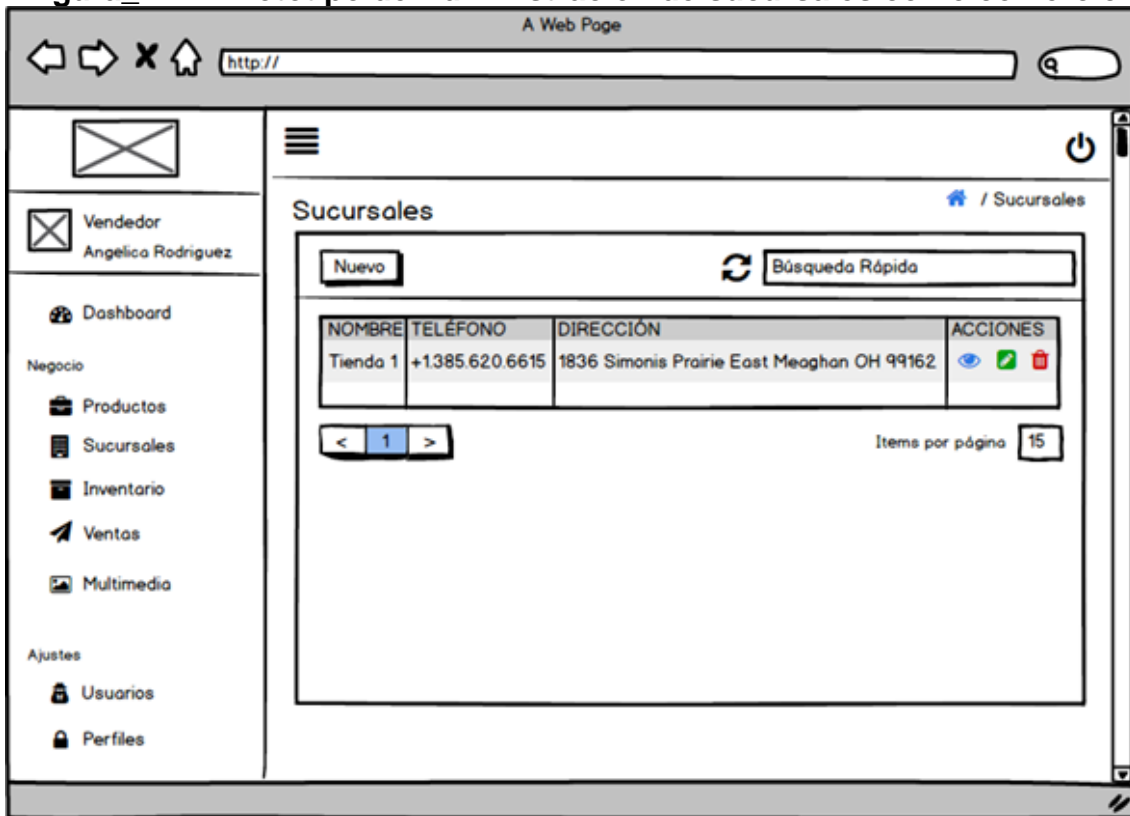
**RF21: El sistema permite gestionar las sucursales del negocio del comerciante con el fin de poder tener sus ubicaciones de entrega y almacén de productos, por parte del usuario comercio**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 141: Prototipo de Administración de sucursales como comercio



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 142: Controlador de Administración de sucursales como comercio

```
BranchController.php x
app > Http > Controllers > Backoffice > Store > BranchController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Backoffice\Store;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\StoreBranch;
7  use App\Utils\MarketUtil;
8  use Illuminate\Http\Request;
9
10 class BranchController extends Controller {
11
12     public function index() {
13         $title = "Sucursales";
14         return view('back-office.store.branch', compact('title'));
15     }
16
17     public function lists(){
18         $store_id = auth()->guard('store')->user()->store_id;
19         $model = StoreBranch::where("store_id", $store_id)->get();
20         return response()->json($model);
21     }
22
23     public function create(Request $request) {
24         $data = $request->all();
25         $place_address = json_decode($data["place_info"]);
26         $market_util = new MarketUtil();
27         $get_lat_lng = $market_util->getLatLng($place_address->value->place_id);
28         if($get_lat_lng["status"] === "ok") {
29             $store_id = auth()->guard('store')->user()->store_id;
30             $data["address"] = $place_address->label;
31             $data["latitude"] = $get_lat_lng["data"]->result->geometry->location->lat;
32             $data["longitude"] = $get_lat_lng["data"]->result->geometry->location->lng;
33             $data["store_id"] = $store_id;
34             $data["place_info"] = json_encode($place_address);
35             $model = StoreBranch::create($data);
36             return response()->json(["status" => "ok", 'data' => $model]);
37         }else {
38             return response()->json(["status" => "error", "message" => "No hemos podido ubicar su dirección"]);
39         }
40     }
41
42     public function update(Request $request){
43         $data = $request->all();
44         $place_address = json_decode($data["place_info"]);
```

**Figura\_ 143: Modelo de Administración de sucursales como comercio**

```
StoreBranch.php x
app > Models > StoreBranch.php > ...
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class StoreBranch extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'address',
20         'phone',
21         "latitude",
22         "longitude",
23         'place_info',
24         "store_id"
25     ];
26
27     public function store(){
28         return $this->hasOne('App\Models\Store', 'id', 'store_id');
29     }
30 }
31
```



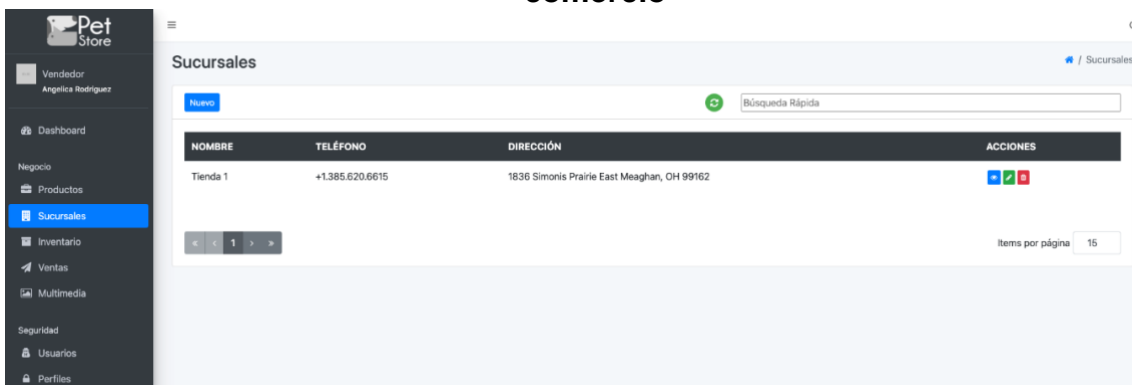
**Figura\_ 144: Vista de Administración de sucursales como comercio**

```
branch.blade.php M X
resources > views > back-office > store > branch.blade.php script
1 @extends('layouts.back-office.store', ["title" => $title])
2
3 @section('content')
4
5 @php
6     use App\Utils\ReactCrudField;
7     use App\Utils\ReactCrudSettingsBuilder;
8
9     $builder = new ReactCrudSettingsBuilder();
10
11     $name = new ReactCrudField('name');
12     $name->title('Nombre')->required(true)->width(8);
13     $builder->addField($name);
14
15     $phone = new ReactCrudField('phone');
16     $phone->title('Teléfono')->required(true)->width(4);
17     $builder->addField($phone);
18
19     $address = new ReactCrudField('address');
20     $address->title('Dirección')->fillable(false);
21     $builder->addField($address);
22
23     $place_info = new ReactCrudField('place_info');
24     $place_info->title('Dirección')->show(false)->type("google_autocomplete")->required(true)->width(12);
25     $builder->addField($place_info);
26
27     $actions = [];
28     $actions["custom"] = [];
29     $actions['view'] = [];
30     $actions['create'] = ['url' => route('store.branch.offices.create')];
31     $actions['update'] = ['url' => route('store.branch.offices.update')];
32     $actions['delete'] = ['url' => route('store.branch.offices.delete')];
33     $builder->setActions($actions);
34 @endphp
35 <div class="content custom-scrollbar">
36     <div id="{{\App\Utils\ReactComponents::LARAVEL_CRUD_COMPONENT}}" data-url="{{route('store.branch.of
37 </div>
38 @php
39     $google_maps_api_key = \App\Models\Setting::where("key", "GOOGLE_API_KEY")->get()->first();
40     $google_maps_api_key_value = $google_maps_api_key["value"];
41 @endphp
42 <script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key={{$google_maps_api_key_
43
44 @endsection
```

## IMPLEMENTACIÓN

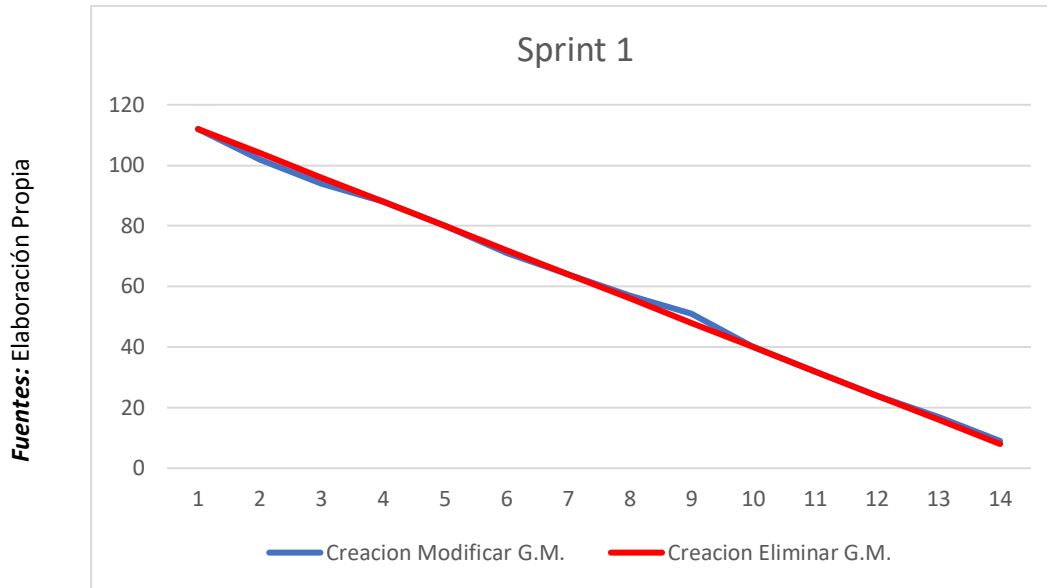
A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el product owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 145: Implementación de Administración de sucursales como comercio**



## BURNDOWN DEL SPRINT N° 4

Figura\_ 146: Burndown del Sprint 4



### Burndown Sprint 4

En la figura vista, se muestra el proceso de desarrollo de cada sprint la cual está representada a través de un gráfico, como podemos observar tenemos dos líneas la cual la línea roja representa al tiempo ideal que se define para el desarrollo del sprint, de igual manera la línea azul representa el tiempo real que tomo el desarrollo del sprint. Ya teniendo bien definido las representaciones tenemos que concluir que si la línea azul se encuentra por debajo de la roja significa que se presentó un adelanto, de lo contrario si la línea azul está por encima es decir que hubo un retraso.

## ACTA DE REUNION DEL SPRINT N° 4

Siendo las 3 pm del día 10 de julio del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.

Presentes:


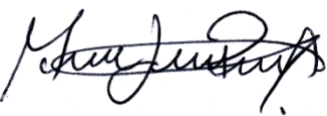


ROL	NOMBRE
-----	--------

Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El encargado de la Empresa AC Brands & Consulting E.I.R.L., Isaac Abensur Vargas concluye la reunión con los Sres. Marzio Perez y Nicolás Segovia aclarando los últimos puntos sobre el sprint.

Para determinar las formas de trabajo y los requerimientos respectivos para esta interacción, también la fecha de entrega y firmar el acta para dar por terminado la reunión.

Los participantes manifestaron su aprobación respecto a todos los puntos definidos en la reunión, que fueron de ayuda a la planificación del Sprint, manifestando su compromiso para la entrega de los resultados en las fechas pactadas.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

#### ACTA DE ENTREGA DEL SPRINT N° 4

Siendo la 1 pm del día 20 de julio del 2021 se reúne en la empresa AC Brands & Consulting E.I.R.L.


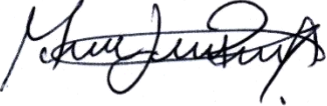


ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique

Product Owner	Juan Carlos Galagarza
---------------	-----------------------

Los Sres. Marzio Perez y Nicolás Segovia, especifican cada uno de los requerimientos desarrollados, también presentan las interfaces elaboradas y brindadas por el producto Owner.

Luego de ser verificadas las presentaciones realizadas por los Srs. Marzio Perez y Nicolás Segovia, se decidió de manera unánime, dar por aprobado el término del Sprint respectivo, del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Los presentes manifestarán su aprobación al informe presentado por los Sres. Marzio Perez y Nicolás Segovia del Sprint N° 4 terminado del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

### RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 4

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	AC Brands & Consulting E.I.R.L.
<b>Proyecto</b>	“Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Información de la reunión:

<b>Lugar</b>	AC Brands & Consulting E.I.R.L.
--------------	---------------------------------

<b>Fecha</b>	20/07/2021
<b>Número de iteración / Sprint</b>	Sprint 4
<b>Personas Convocadas a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza
<b>Persona que asistieron a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
Se desarrolló el ingreso como comercio de manera exitosa, respetando las validaciones de los privilegios. Se logró desarrollar la administración de productos, ventas y sucursales.	Se registró fallas en las validaciones de los privilegios, pero se logró solucionar.

## PLANIFICACIÓN DEL SPRINT N°5

Siendo las 4 pm del día 20 de julio del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.

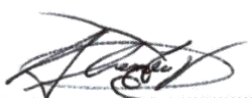
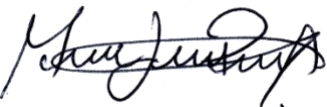


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El gerente de la Empresa AC Brands & Consulting E.I.R.L., efectúa la presentación de los requerimientos, de igual manera manifiesta los requerimientos que tienen mayor prioridad.

Realizado el análisis de los requerimientos presentados por el gerente de AC Brands & Consulting E.I.R.L., el señor Juan Galarza aclara dudas existentes y luego se compromete al cumplimiento de todos los requerimientos propuestas en el Sprint 5.

Los participantes de la reunión manifiestan su aprobación con respecto a la presentación del Sprint 5, así mismo especificando que la fecha de entrega será el día 4 de agosto del 2021.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## EJECUCIÓN DEL SPRINT 5

Lo primero que se debe realizar antes del diseño, es conocer y comprender el funcionamiento del sistema, en otras palabras, analizar lo que realmente se necesita, lo cual será elaborado según las historias de usuario.

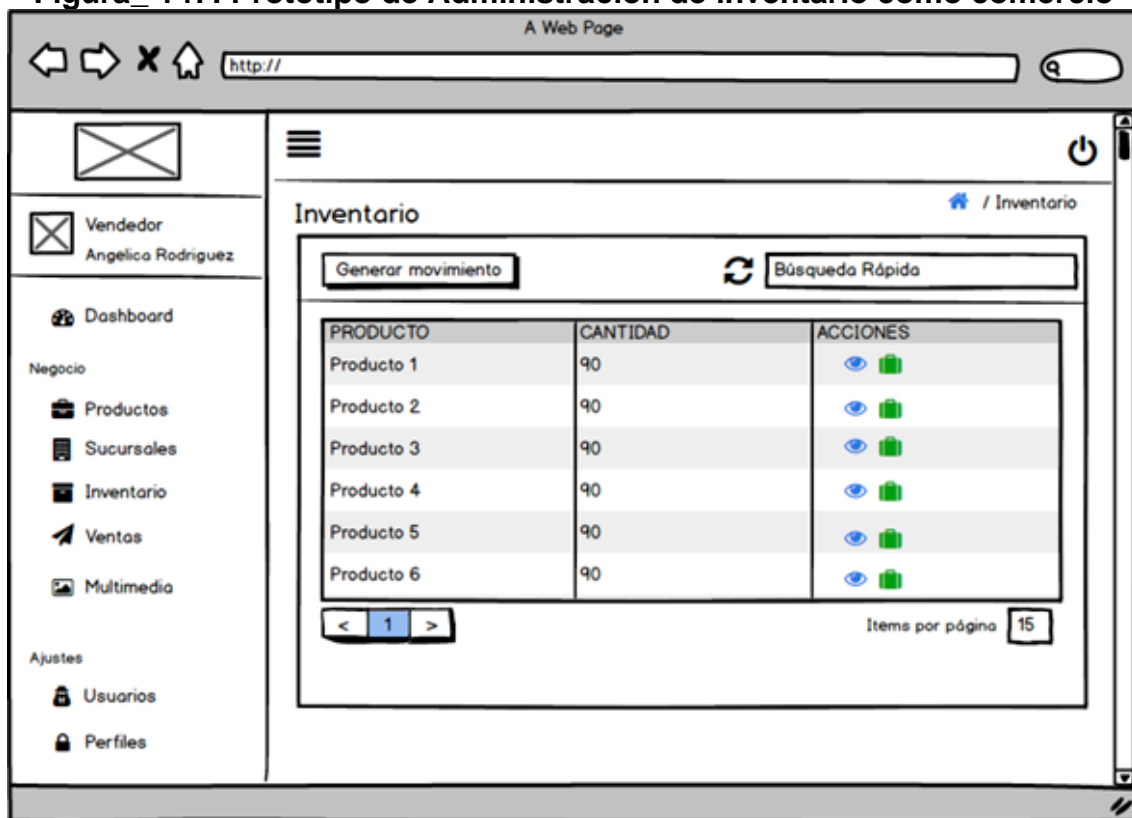
**RF22: El sistema permite gestionar el inventario de los productos con el fin de poder generar ingresos y egresos respecto a los productos que tengo en cada sucursal, por parte del usuario comercio**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 147: Prototipo de Administración de inventario como comercio**



## **DESARROLLO**

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura 148: Controlador de Administración de inventario como comercio

```
InventoryController.php x
app > Http > Controllers > Backoffice > Store > InventoryController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Backoffice\Store;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Inventory;
7  use App\Models\InventoryMovement;
8  use App\Models\InventoryVariation;
9  use App\Models\Product;
10 use App\Models\StoreBranch;
11 use Illuminate\Http\Request;
12 use Illuminate\Support\Facades\DB;
13
14 class InventoryController extends Controller {
15
16     public function index() {
17         $title = "Inventario";
18         $store_id = auth()->guard('store')->user()->store_id;
19         $branches = StoreBranch::where("store_id", $store_id)->get();
20         return view('back-office.store.inventory.index', compact('title', 'branches'));
21     }
22
23     public function lists() {
24         $store_id = auth()->guard('store')->user()->store_id;
25         $model = DB::select('call store_inventories('.$store_id.')');
26         return response()->json($model);
27     }
28
29     public function detail(Request $request){
30         $data = $request->all();
31         $title = "Detalle de inventario";
32         $inventory = Inventory::find($data["id"]);
33         $detail = InventoryVariation::with('variation', 'branch')->where("inventory_id", $data["id"])->get(
34
35         return view('back-office.store.inventory.detail', compact('title', 'detail', 'inventory'));
36     }
37
38     public function movements(Request $request){
39         $data = $request->all();
40         $title = "Movimientos de inventario";
41         $inventory = Inventory::find($data["id"]);
42         $movements = InventoryMovement::with('order', 'inventory_variation.variation')->where("inventory_id
43         return view('back-office.store.inventory.movements', compact('title', 'movements', 'inventory'));
44     }
45
46     public function search_product(Request $request) {
47         $data = $request->all();
```



**Figura\_ 149: Modelo de Administración de inventario como comercio**

```
branch.blade.php M Inventory.php x
app > Models > Inventory.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class Inventory extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'product_id',
19         'product_name',
20         'quantity'
21     ];
22 }
23
```

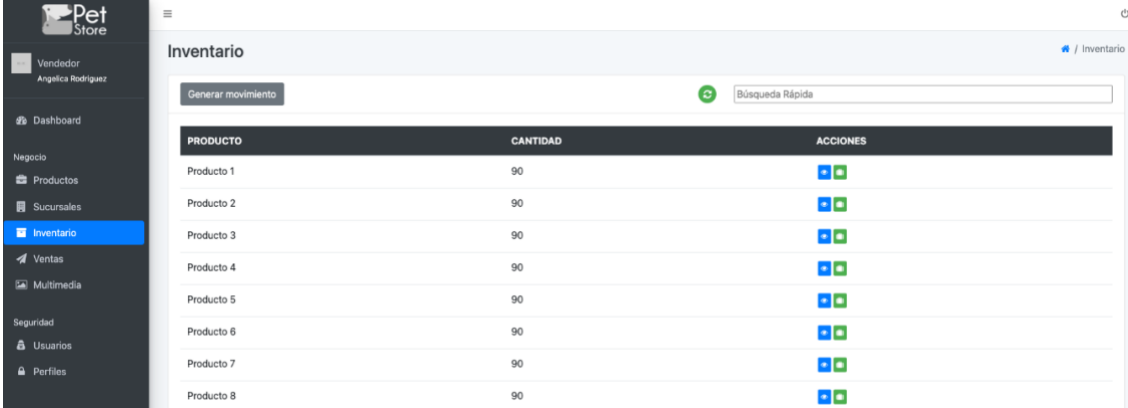
Figura\_ 150: Vista de Administración de inventario como comercio

```
index.blade.php x
resources > views > back-office > store > inventory > index.blade.php > ...
1 @extends('layouts.back-office.store', ["title" => $title])
2
3 @section('content')
4
5     @php
6         use App\Utils\ReactCrudField;
7         use App\Utils\ReactCrudSettingsBuilder;
8
9         $builder = new ReactCrudSettingsBuilder();
10
11         $product_nameField = new \App\Utils\ReactCrudField('product_name');
12         $product_nameField->title('Producto')->required(true)->width(6);
13         $builder->addField($product_nameField);
14
15         $quantityField = new \App\Utils\ReactCrudField('quantity');
16         $quantityField->title('Cantidad')->required(true)->width(6);
17         $builder->addField($quantityField);
18
19         $actions = [];
20
21         $actions["custom"] = [];
22         $actions['custom'] = array_merge(
23             $actions["custom"],
24             [
25                 "detail" => [
26                     "link" => true,
27                     'url' => route('store.inventory.detail'),
28                     'icon' => "eye",
29                     "color" => "#007bff",
30                     "params" => [ 'id' ],
31                     "title" => "Ver detalle"
32                 ],
33                 "movements" => [
34                     "link" => true,
35                     'url' => route('store.inventory.movements'),
36                     'icon' => "suitcase",
37                     "color" => "#4caf50",
38                     "params" => [ 'id' ],
39                     "title" => "Ver movimientos"
40                 ]
41             ]
42         );
43
44         $actions['inventory-movement'] = [
45             'branches' => json_encode($branches),
46             'url-process-movements' => route('store.inventory.process.movements'),
47             'url-search-product' => route('store.inventory.search-product')
48         ];
49     @endphp
50 @endsection
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 151: Implementación de Administración de inventario como comercio**



PRODUCTO	CANTIDAD	ACCIONES
Producto 1	90	[+][-]
Producto 2	90	[+][-]
Producto 3	90	[+][-]
Producto 4	90	[+][-]
Producto 5	90	[+][-]
Producto 6	90	[+][-]
Producto 7	90	[+][-]
Producto 8	90	[+][-]

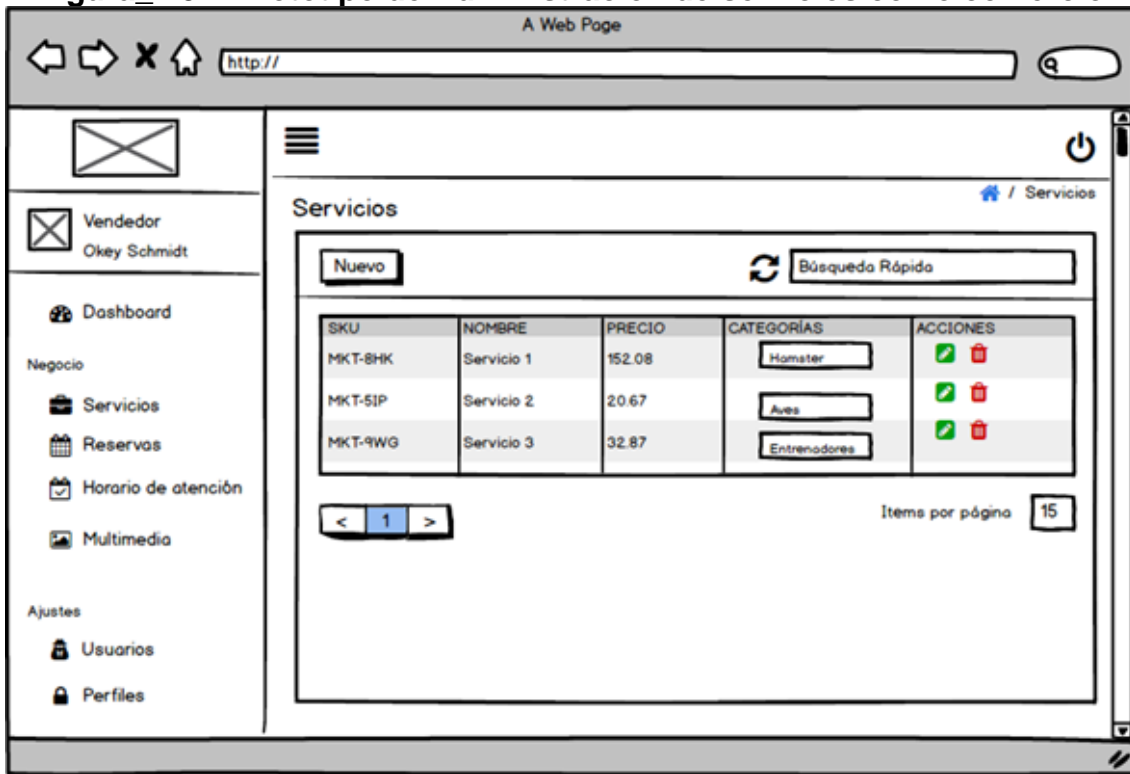
**RF23: El sistema permite registrar los servicios con el fin de que los clientes puedan realizar reservas en la plataforma, por parte del usuario entrenador**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 152: Prototipo de Administración de servicios como comercio



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 153: Controlador de Administración de servicios como comercio

```
ServiceController.php x
app > Http > Controllers > Backoffice > Admin > ServiceController.php > ServiceController > edit
15 class ServiceController extends Controller {
16
17     public function index() {
18         $title = "Servicios";
19
20         $categories = array_map(
21             function($item){
22                 return [
23                     "id" => $item['id'],
24                     "value" => $item['internal_name']
25                 ];
26             }, Category::where('type', 'products')->get()->toArray()
27         );
28
29         $stores = array_map(
30             function($item){
31                 return [
32                     "id" => $item['id'],
33                     "value" => $item['name']
34                 ];
35             }, Store::where("is_trainer", true)->get()->toArray()
36         );
37         return view('back-office.admin.services.index', compact('title', 'categories', 'stores'));
38     }
39
40     public function lists(){
41         $model = DB::select('call all_services');
42         return response()->json($model);
43     }
44
45     public function create(Request $request){
46         try{
47             $data = $request->all();
48
49             $slug_util = new SlugUtil();
50
51             $categories = explode(",", $data["categories"]);
52             $parent_slug_id = null;
53             if (count($categories) > 0){
54                 $primary_category = Category::find($categories[0]);
55                 $parent_slug_id = $primary_category["slug_route_id"];
56             }
57             $slug = $slug_util->createSlug($data["name"], $parent_slug_id);
58
59             $data["slug_route_id"] = $slug->id;
60             $data["is_service"] = true;
61         }
```

Figura\_ 154: Modelo de Administración de servicios como comercio

```
Product.php X
app > Models > Product.php > Product > $fillable
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Product extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'sku_code',
20         'description',
21         'summary',
22         'featured_image_id',
23
24         'categories',
25
26         'is_featured',
27
28         'price',
29         'iva',
30         'price_with_iva',
31
32         'offer_price',
33         'offer_iva',
34         'offer_price_with_iva',
35
36         'is_service',
37         'service_attentions_per_hour',
38         'rate',
39
40         'brand_id',
41         'brand_name',
42
43         'slug_route_id',
44         'url',
45
46         'store_id',
```

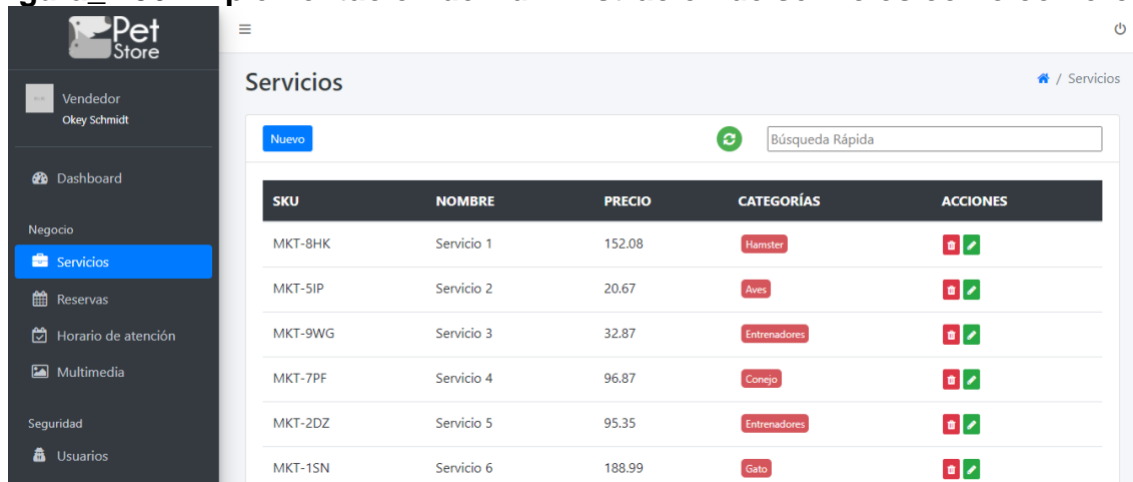
Figura\_ 155: Vista de Administración de servicios como comercio

```
index.blade.php x
resources > views > back-office > admin > services > index.blade.php > ...
1 @extends('layouts.back-office.admin', ["title" => $title])
2
3 @section('content')
4
5 @php
6     use App\Utils\ReactCrudField;
7     use App\Utils\ReactCrudSettingsBuilder;
8
9     $builder = new ReactCrudSettingsBuilder();
10
11     $sku_code = new ReactCrudField('sku_code');
12     $sku_code->title('SKU')->required(true)->width(2);
13     $builder->addField($sku_code);
14
15     $name = new ReactCrudField('name');
16     $name->title('Nombre')->required(true)->width(10);
17     $builder->addField($name);
18
19     $price = new ReactCrudField('price');
20     $price->title('Precio')->required(true)->width(2);
21     $builder->addField($price);
22
23     $store_id = new ReactCrudField('store_id');
24     $store_id->title('Entrenador')->type('map', $stores)->required(true)->width(3);
25     $builder->addField($store_id);
26
27     $categories_field = new ReactCrudField('categories');
28     $categories_field->title('Categorías')->type('json', $categories)->width(12);
29     $builder->addField($categories_field);
30
31     $actions = [];
32     $actions["custom"] = [];
33     $actions["create-custom"] = [
34         'url' => route('admin.products.create'),
35         'tabs' => [
36             [
37                 "name" => "Fotos",
38                 "type" => 'images',
39                 'multimedia-lists-url' => route('admin.multimedia.lists')
40             ],
41             [
42                 "name" => "Imagen destacada",
43                 "type" => 'featured_image',
44                 'multimedia-lists-url' => route('admin.multimedia.lists')
45             ],
46         ]
47     ]
48 }
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 156: Implementación de Administración de servicios como comercio**



**RF24: El sistema permite visualizar las reservas en un calendario con el fin de poder llevar un control de las mismas, por el usuario entrenador**

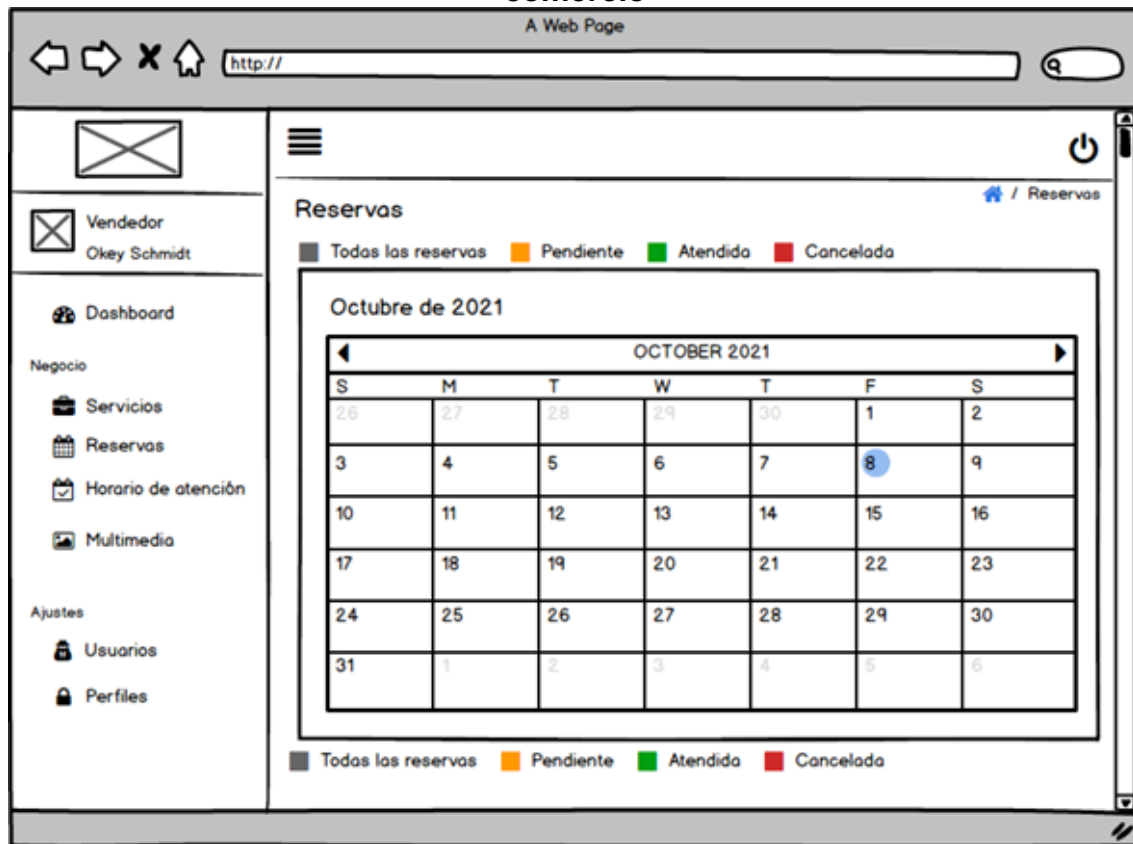
## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.



Figura\_ 157: Prototipo de Administración de reservas generadas como comercio



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 158: Controlador de Administración de reservas generadas como comercio

```
BookingController.php X
app > Http > Controllers > Backoffice > Store > BookingController.php > ...
1 <?php
2
3 namespace App\Http\Controllers\Backoffice\Store;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\StoreOrder;
7 use App\Models\StoreOrderDetail;
8 use Illuminate\Http\Request;
9 use Illuminate\Support\Facades\DB;
10
11 class BookingController extends Controller {
12
13     public function index(){
14         $title = "Reservas";
15         return view('back-office.store.booking.index', compact('title'));
16     }
17
18     public function lists(Request $request){
19         $data = $request->all();
20
21         $store_id = auth()->guard('store')->user()->store_id;
22         $orders = DB::select('call store_orders(.'.$store_id.'.)');
23         $order_ids = [];
24         $bookings = [];
25         if(count($orders) > 0) {
26             foreach($orders as $order) {
27                 $order_ids[] = $order->id;
28             }
29             $model = StoreOrderDetail::query();
30             $model->whereIn("store_order_id", $order_ids);
31             if($data["filter_by"] != "ALL"){
32                 $model->where("service_booking_status", $data["filter_by"]);
33             }
34             $bookings = $model->get();
35         }
36         return response()->json($bookings);
37     }
38
39     public function get_booking(Request $request) {
40         $data = $request->all();
41         $model = StoreOrderDetail::find($data["id"]);
42         return response()->json(["status" => "ok", "data" => $model]);
43     }
44
45     public function attended_booking(Request $request) {
46         $data = $request->all();
47         $model = StoreOrderDetail::find($data["id"]);
48     }
49 }
```

**Figura\_ 159: Modelo de Administración de reservas generadas como comercio**

```
StoreOrderDetail.php x
app > Models > StoreOrderDetail.php > ...
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class StoreOrderDetail extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'store_order_id',
19         'product_id',
20         'product_sku_code',
21         'product_name',
22
23         'variation_id',
24         'variation_sku_code',
25         'variation_name',
26
27         'store_id',
28         'store_name',
29
30         'store_branch_id',
31         'store_branch_name',
32         'store_branch_address',
33         'store_branch_phone',
34         "store_branch_latitude",
35         "store_branch_longitude",
36
37         "is_service",
38         'service_booking_date',
39         'service_booking_status',
40
41         'quantity',
42         'price',
43         'sub_total'
44     ];
45
46     protected $appends = ["title", "date", "className"];
47
```

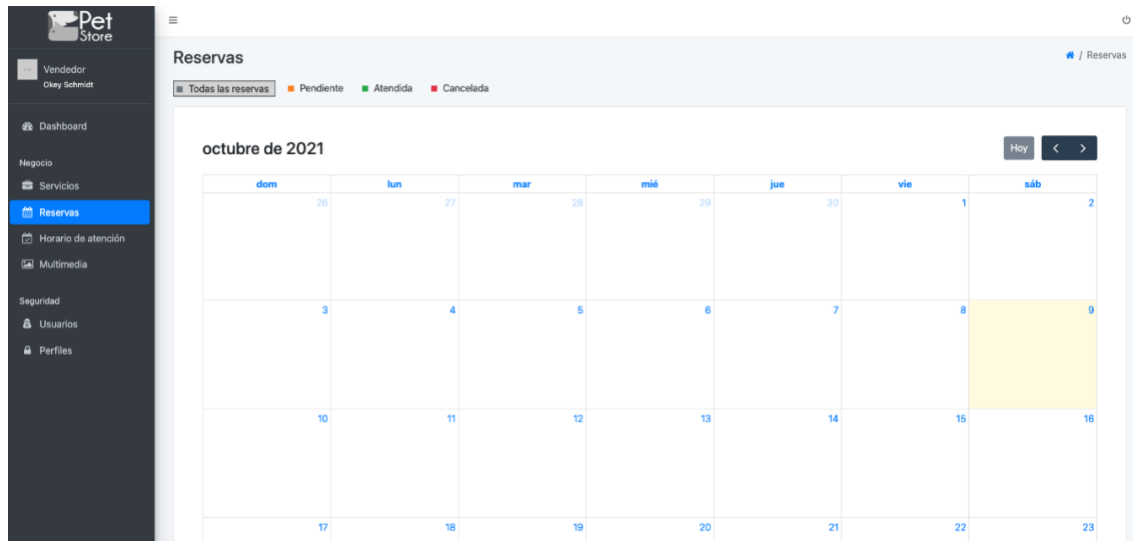
**Figura\_ 160: Vista de Administración de reservas generadas como comercio**

```
index.blade.php x
resources > views > back-office > store > booking > index.blade.php > ...
1 @extends('layouts.back-office.store', ["title" => $title])
2
3 @section('content')
4
5     <div class="booking-calendar-component"
6         data-booking-lists-url="{{route('store.bookings.lists')}}"
7         data-booking-info-url="{{route('store.bookings.info')}}"
8         data-booking-attended-url="{{route('store.bookings.attended')}}"></div>
9
10 @endsection
11
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 161: Implementación de Administración de reservas generadas como comercio**



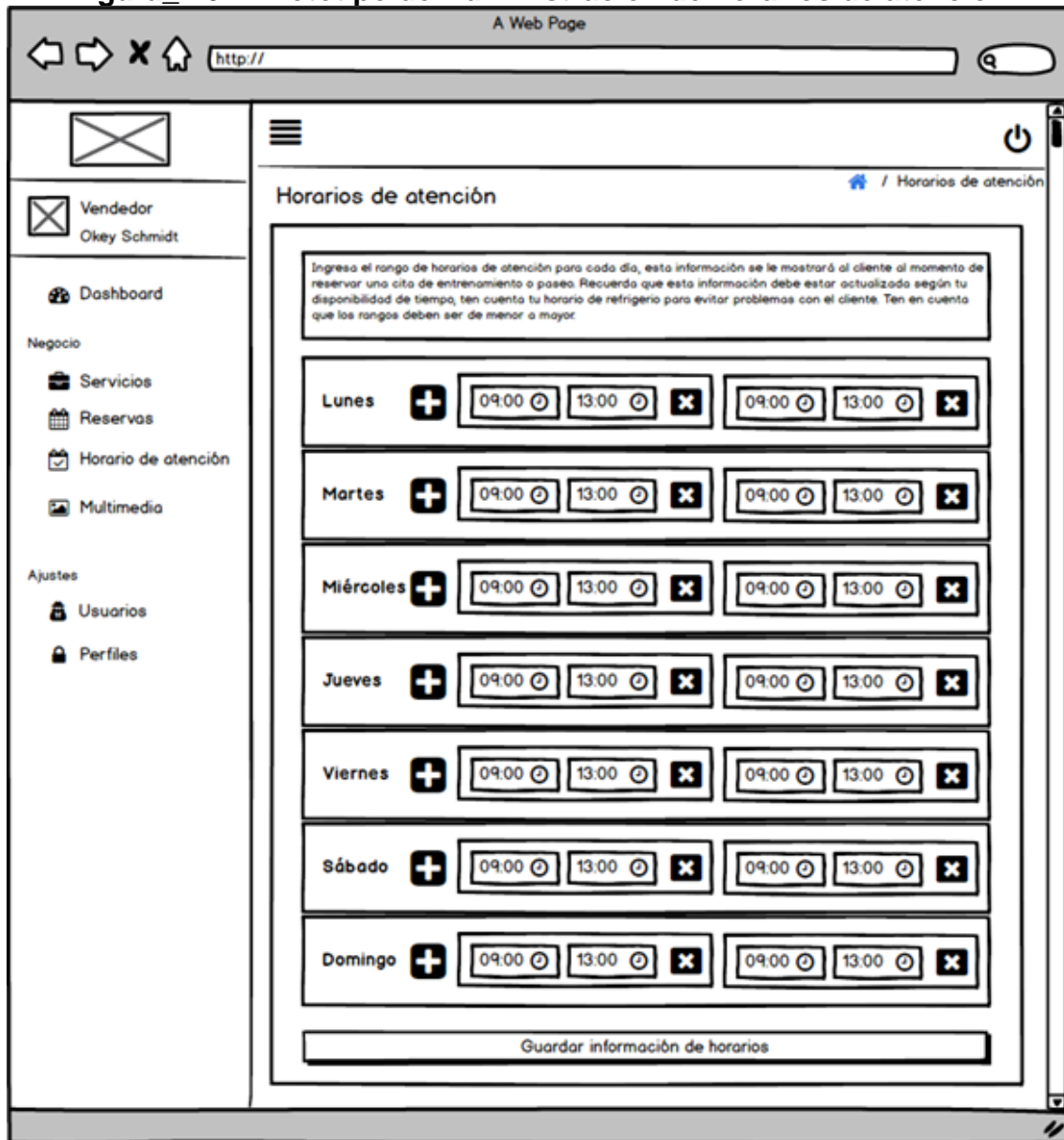
**RF25: El sistema permite gestionar los horarios de atención, con el fin de que los clientes sepan la disponibilidad de los comercios, como usuario entrenador**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 162: Prototipo de Administración de horarios de atención



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 163: Controlador de Administración de horarios de atención

```
BusinessHourController.php x
app > Http > Controllers > Backoffice > Store > BusinessHourController.php > ...
1 <?php
2
3 namespace App\Http\Controllers\Backoffice\Store;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\TrainerAttentionDate;
7 use App\Models\TrainerAttentionDateTime;
8 use Illuminate\Http\Request;
9
10 class BusinessHourController extends Controller {
11
12     public function index(){
13         $title = "Horarios de atención";
14         $store_id = auth()->guard('store')->user()->store_id;
15         $attention_dates = TrainerAttentionDate::with('times')->where("trainer_id", $store_id)->get();
16         return view('back-office.store.business-hour', compact('title', 'attention_dates'));
17     }
18
19     public function update(Request $request){
20         $data = $request->all();
21
22         foreach($data as $item) {
23             if(count($item["times"]) > 0) {
24                 foreach($item["times"] as $time){
25                     // Creación horario
26                     if (isset($time["create"])) {
27                         if ($time["create"]) {
28                             if($time["start"] && $time["end"]) {
29                                 TrainerAttentionDateTime::create([
30                                     "start" => $time["start"],
31                                     "end" => $time["end"],
32                                     "trainer_attention_date_id" => $item["id"]
33                                 ]);
34                             }
35                         }
36                     }
37
38                     if (isset($time["id"])){
39                         $trainer_attention_time = TrainerAttentionDateTime::find($time["id"]);
40                         // Actualización de horario
41                         $trainer_attention_time->update([
42                             "start" => $time["start"],
43                             "end" => $time["end"]
44                         ]);
45
46                         // Eliminación de horario
```

**Figura\_ 164: Modelo de Administración de horarios de atención**

```
TrainerAttentionDate.php x
app > Models > TrainerAttentionDate.php > ...
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class TrainerAttentionDate extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         // LUNES: 1 | MARTES: 2 | MIERCOLES | 3 JUEVES | 4 | VIERNES: 5 | SABADO: 6 | DOMINGO: 7
19         'day_of_week',
20         'trainer_id',
21         'deleted'
22     ];
23
24     public function times(){
25         return $this->hasMany('App\Models\TrainerAttentionDateTime', 'trainer_attention_date_id', 'id');
26     }
27 }
28
```

**Figura\_ 165: Vista de Administración de horarios de atención**

```
business-hour.blade.php x
resources > views > back-office > store > business-hour.blade.php > ...
1 @extends('layouts.back-office.store', ["title" => $title])
2
3 @section('content')
4
5     <div class="business-hour-component"
6         data-attention-dates="{{json_encode($attention_dates)}}"
7         data-attention-date-update-url="{{route('store.business.hours.update')}}"></div>
8
9 @endsection
10
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 166: Implementación de Administración de horarios de atención**

**Pet Store**  
Vendedor  
Oliver Schmidt

Dashboard

Negocio

- Servicios
- Reservas
- Horario de atención**
- Multimedia

Seguridad

- Usuarios
- Perfiles

### Horarios de atención

Horarios de atención

Ingresar el rango de horarios de atención para cada día, esta información se le mostrará al cliente al momento de reservar una cita de entrenamiento o paseo. Recuerda que esta información debe estar actualizada según tu disponibilidad de tiempo, ten cuenta tu horario de refrigerio para evitar problemas con el cliente. Ten en cuenta que los rangos deben ser de menor a mayor.

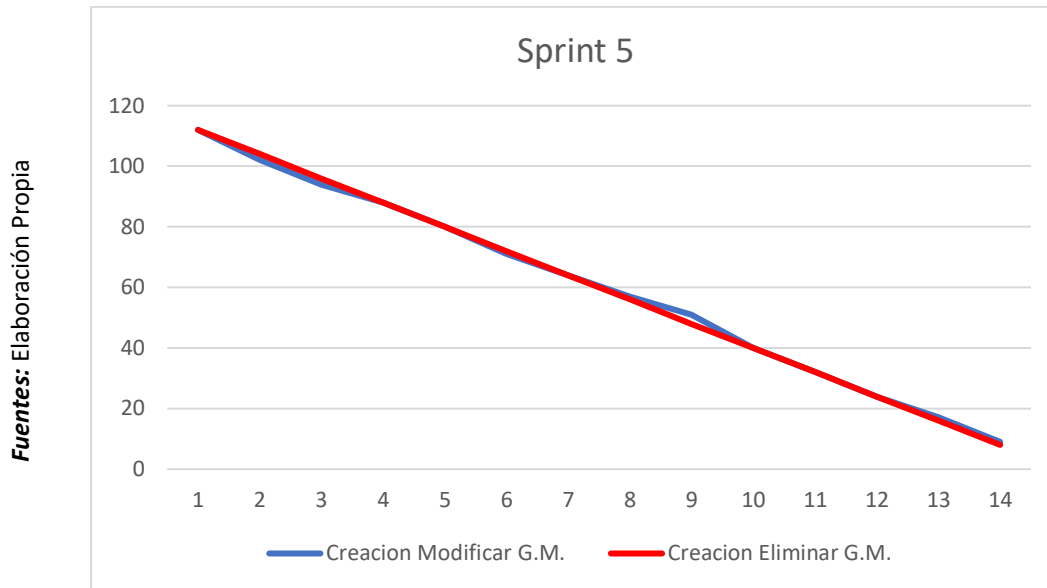
Lunes	<input type="checkbox"/>	09:00	<input type="checkbox"/>	13:00	<input type="checkbox"/>	<input type="checkbox"/>	15:00	<input type="checkbox"/>	20:00	<input type="checkbox"/>
Martes	<input type="checkbox"/>	09:00	<input type="checkbox"/>	13:00	<input type="checkbox"/>	<input type="checkbox"/>	15:00	<input type="checkbox"/>	20:00	<input type="checkbox"/>
Miércoles	<input type="checkbox"/>	09:00	<input type="checkbox"/>	13:00	<input type="checkbox"/>	<input type="checkbox"/>	15:00	<input type="checkbox"/>	20:00	<input type="checkbox"/>
Jueves	<input type="checkbox"/>	09:00	<input type="checkbox"/>	13:00	<input type="checkbox"/>	<input type="checkbox"/>	15:00	<input type="checkbox"/>	20:00	<input type="checkbox"/>
Viernes	<input type="checkbox"/>	09:00	<input type="checkbox"/>	13:00	<input type="checkbox"/>	<input type="checkbox"/>	15:00	<input type="checkbox"/>	20:00	<input type="checkbox"/>
Sábado	<input type="checkbox"/>	09:00	<input type="checkbox"/>	13:00	<input type="checkbox"/>	<input type="checkbox"/>	15:00	<input type="checkbox"/>	20:00	<input type="checkbox"/>
Domingo	<input type="checkbox"/>	09:00	<input type="checkbox"/>	13:00	<input type="checkbox"/>	<input type="checkbox"/>	15:00	<input type="checkbox"/>	20:00	<input type="checkbox"/>

Guardar información de horarios



## BURNDOWN DEL SPRINT N° 5

Figura\_ 167: Burndown del Sprint 5



### Burndown Sprint 5

En la figura vista, se muestra el proceso de desarrollo de cada sprint la cual está representada a través de un gráfico, como podemos observar tenemos dos líneas la cual la línea roja representa al tiempo ideal que se define para el desarrollo del sprint, de igual manera la línea azul representa el tiempo real que tomo el desarrollo del sprint. Ya teniendo bien definido las representaciones tenemos que concluir que si la línea azul se encuentra por debajo de la roja significa que se presentó un adelanto, de lo contrario si la línea azul está por encima es decir que hubo un retraso.

## ACTA DE REUNION DEL SPRINT N° 5

Siendo las 3 pm del día 20 de julio del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.


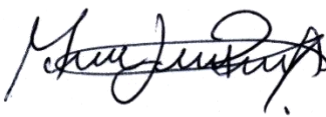


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El encargado de la Empresa AC Brands & Consulting E.I.R.L., Isaac Abensur Vargas concluye la reunión con los Sres. Marzio Perez y Nicolás Segovia aclarando los últimos puntos sobre el sprint.

Para determinar las formas de trabajo y los requerimientos respectivos para esta interacción, también la fecha de entrega y firmar el acta para dar por terminado la reunión.

Los participantes manifestaron su aprobación respecto a todos los puntos definidos en la reunión, que fueron de ayuda a la planificación del Sprint, manifestando su compromiso para la entrega de los resultados en las fechas pactadas.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## ACTA DE ENTREGA DEL SPRINT N° 5


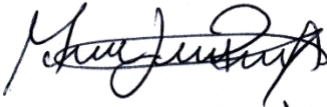


Siendo la 1 pm del día 4 de agosto del 2021 se reúne en la empresa AC Brands & Consulting E.I.R.L.

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

Los Sres. Marzio Perez y Nicolás Segovia, especifican cada uno de los requerimientos desarrollados, también presentan las interfaces elaboradas y brindadas por el producto Owner.

Luego de ser verificadas las presentaciones realizadas por los Srs. Marzio Perez y Nicolás Segovia, se decidió de manera unánime, dar por aprobado el término del Sprint respectivo, del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Los presentes manifestarán su aprobación al informe presentado por los Sres. Marzio Perez y Nicolás Segovia del Sprint N° 5 terminado del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 5

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	AC Brands & Consulting E.I.R.L.
<b>Proyecto</b>	“Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Información de la reunión:

<b>Lugar</b>	AC Brands & Consulting E.I.R.L.
<b>Fecha</b>	04/08/2021
<b>Número de iteración / Sprint</b>	Sprint 5
<b>Personas Convocadas a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza
<b>Persona que asistieron a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
Se desarrolló la administración de inventario, servicios, de reservas y los horarios de atención.	Se presentaron fallas al momento de seleccionar los horarios de atención, se llegó a solucionar.

## PLANIFICACIÓN DEL SPRINT N°6

Siendo las 4 pm del día 4 de agosto del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.

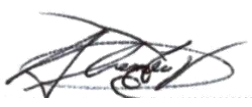
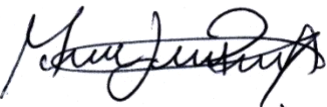


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El gerente de la Empresa AC Brands & Consulting E.I.R.L., efectúa la presentación de los requerimientos, de igual manera manifiesta los requerimientos que tienen mayor prioridad.

Realizado el análisis de los requerimientos presentados por el gerente de AC Brands & Consulting E.I.R.L., el señor Juan Galarza aclara dudas existentes y luego se compromete al cumplimiento de todos los requerimientos propuestas en el Sprint 6.

Los participantes de la reunión manifiestan su aprobación con respecto a la presentación del Sprint 6, así mismo especificando que la fecha de entrega será el día 20 de agosto del 2021.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## EJECUCIÓN DEL SPRINT 6

Lo primero que se debe realizar antes del diseño, es conocer y comprender el funcionamiento del sistema, en otras palabras, analizar lo que realmente se necesita, lo cual será elaborado según las historias de usuario.

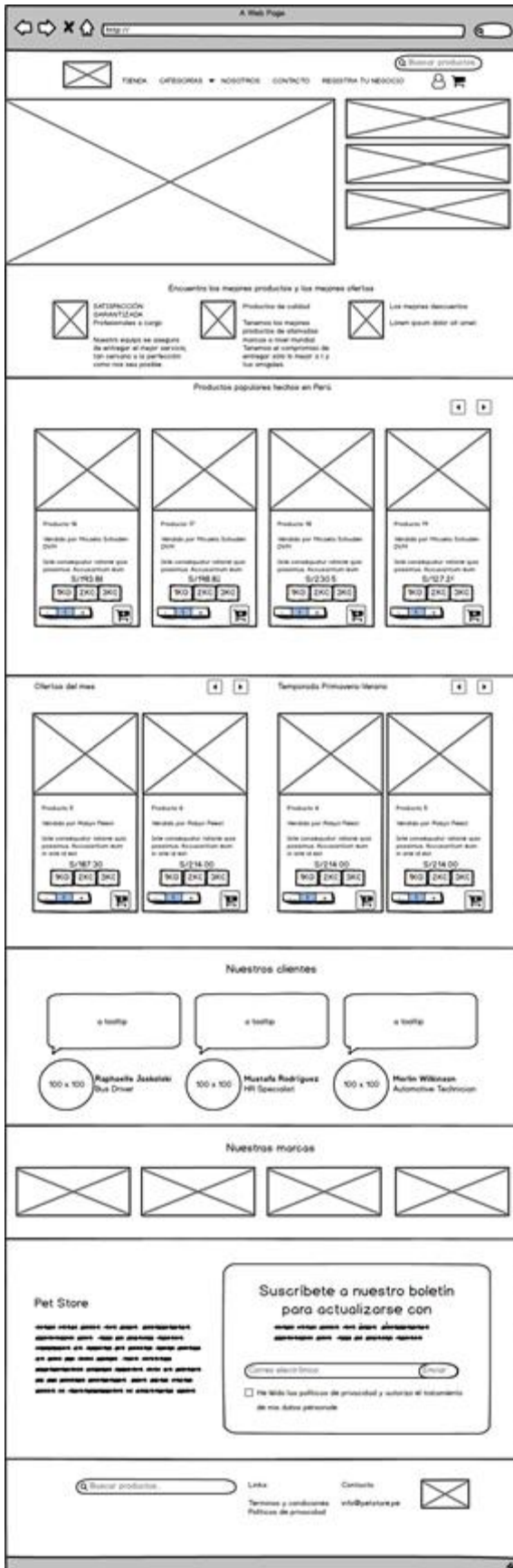
**RF26: El sistema permite que el cliente pueda visualizar la página de inicio con la finalidad de interactuar con todos los elementos**

### DISEÑO

#### PROTOTIPO

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

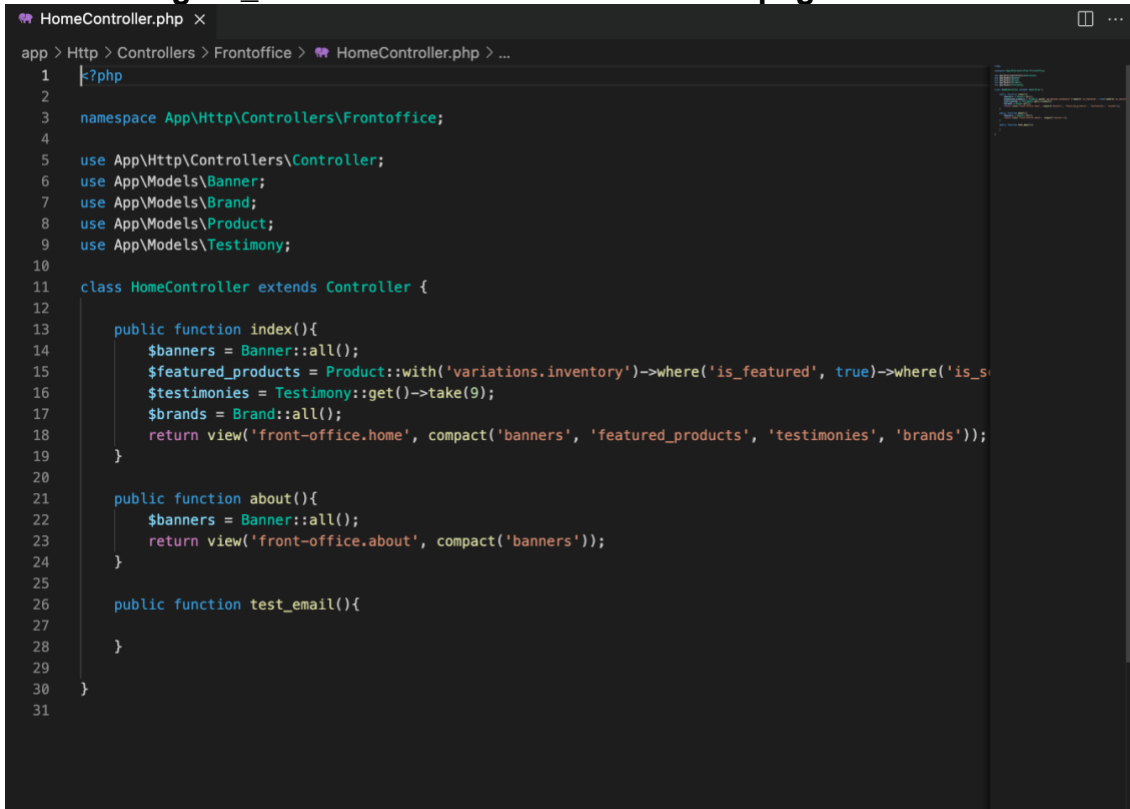
**Figura\_ 168: Prototipo de Interacción página de inicio**



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

**Figura\_ 169: Controlador de Interacción página de inicio**



```
HomeController.php x
app > Http > Controllers > Frontoffice > HomeController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Frontoffice;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Banner;
7  use App\Models\Brand;
8  use App\Models\Product;
9  use App\Models\Testimony;
10
11 class HomeController extends Controller {
12
13     public function index(){
14         $banners = Banner::all();
15         $featured_products = Product::with('variations.inventory')->where('is_featured', true)->where('is_s
16         $testimonies = Testimony::get()->take(9);
17         $brands = Brand::all();
18         return view('front-office.home', compact('banners', 'featured_products', 'testimonies', 'brands'));
19     }
20
21     public function about(){
22         $banners = Banner::all();
23         return view('front-office.about', compact('banners'));
24     }
25
26     public function test_email(){
27
28     }
29
30 }
31
```

**Figura\_ 170: Modelo de Interacción página de inicio**



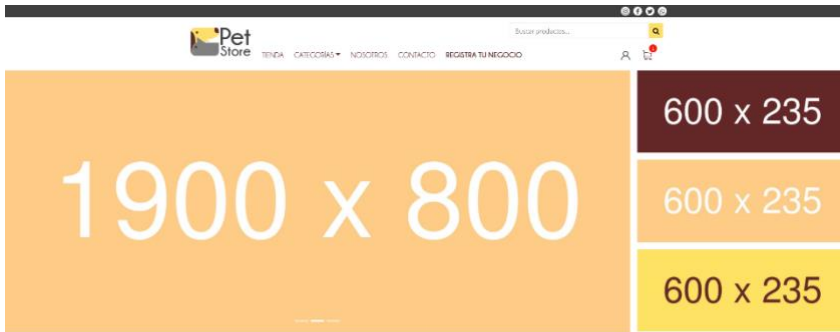
Figura\_ 171: Vista de Interacción página de inicio

```
home.blade.php x
resources > views > front-office > home.blade.php > ...
1 @extends('layouts.front-office.app')
2
3 @section('content')
4
5 @if (count($banners) > 0)
6 <div class="container-fluid pr-0 pl-0 mt-2 mb-3">
7   <div class="row no-gutters w-100">
8     <div class="col-md-9 pr-lg-3 pr-0">
9       <div id="carouselExampleIndicators" class="carousel slide carousel-fade" data-interval="400">
10        <ol class="carousel-indicators">
11          @foreach ($banners as $b => $banner)
12            <li data-target="#carouselExampleIndicators" data-slide-to="{{ $b }}" class="{{ ($b == 0 ? 'active' : '') }}"></li>
13          @endforeach
14        </ol>
15        <div class="carousel-inner">
16          @foreach ($banners as $b => $banner)
17            <div class="carousel-item{{ ($b == 0 ? ' active' : '') }}">
18              @if ($banner["url"])
19                <a href="{{ $banner["url"] }}">
20                  @endif
21                  <picture>
22                    <source srcset="{{ $banner["image_1900x600_url"] }}" media="(min-width: 1900px)">
23                    <source srcset="{{ $banner["image_1400x600_url"] }}" media="(min-width: 1400px)">
24                    <source srcset="{{ $banner["image_800x700_url"] }}" media="(min-width: 800px)">
25                    <img srcset="{{ $banner["image_600x700_url"] }}" alt="responsive image" />
26                  </picture>
27                  @if ($banner["url"])
28                    </a>
29                  @endif
30                </div>
31              @endforeach
32            </div>
33          </div>
34        </div>
35      <div class="col-md-3">
36        <div class="sub-banner w-100 mb-3 wow fadeInRight">
37          
38        </div>
39
40        <div class="sub-banner w-100 mb-3 wow fadeInRight">
41          
42        </div>
43
44        <div class="sub-banner w-100 wow fadeInRight">
45          
46        </div>
47      </div>
48    </div>
49  </div>
50 </div>
```

## IMPLEMENTACIÓN

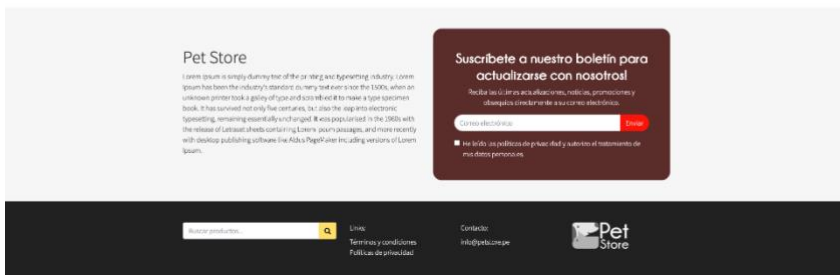
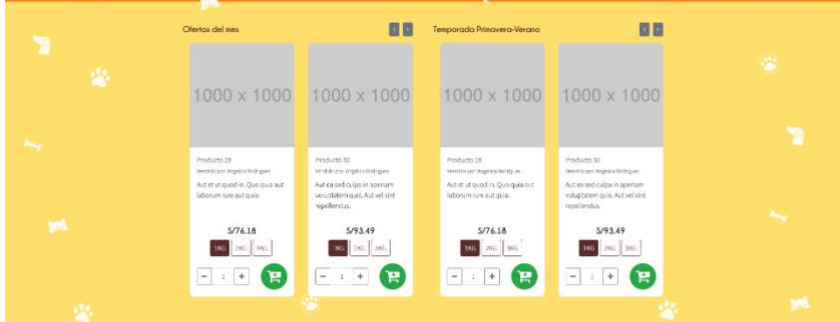
A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

Figura\_ 172: Implementación de Interacción página de inicio



Encuentra los mejores productos y las mejores ofertas

- DISTRIBUCION GARANTIDA**  
**Profesionales a cargo**  
 Nuestra red de expertos es el mejor al mejor servicio, con certificaciones de perfeccionamiento en sus procesos.
- Productos de calidad**  
 Tenemos los mejores productos de alta calidad en materia animal. Tenemos el compromiso de entregar solo lo mejor a ti y tus amigos.
- Los mejores descuentos**  
 ¡Serán tu mejor aliado!



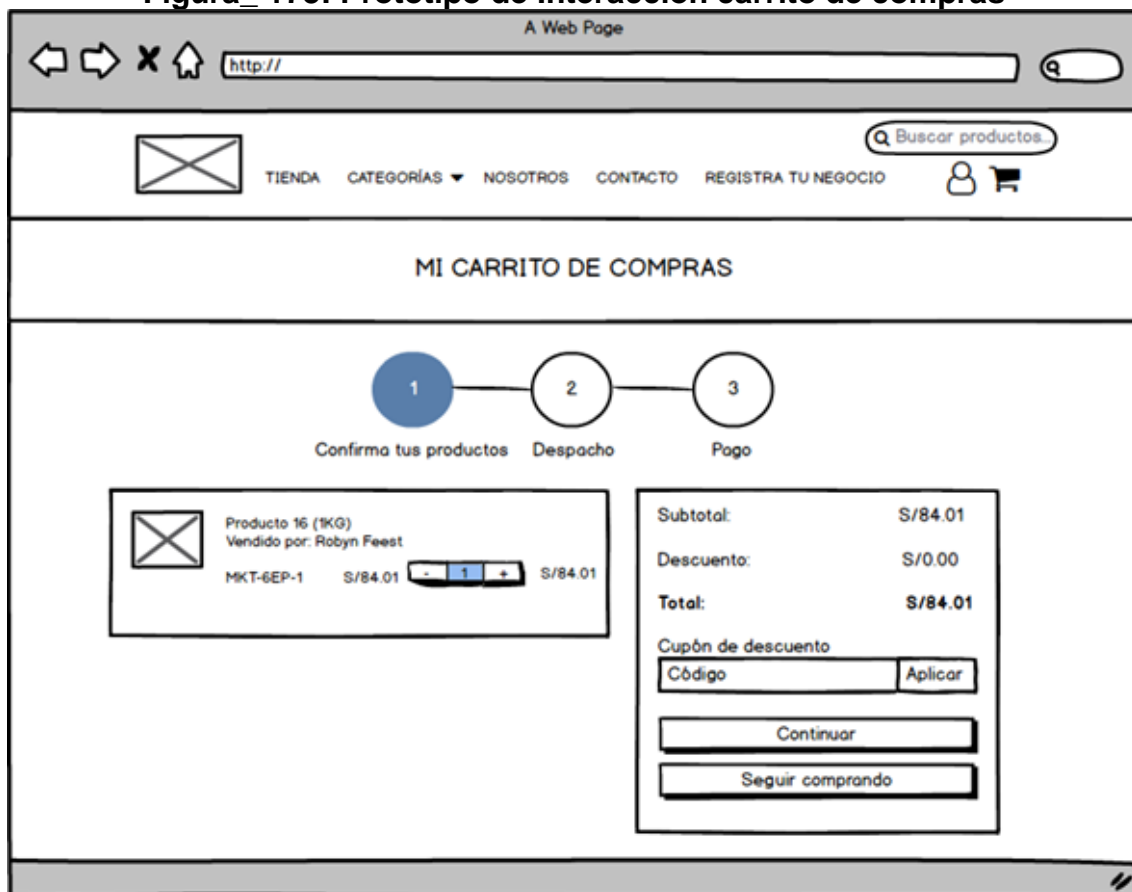
**RF27: El sistema permite poder agregar productos al carrito de compras con el fin de realizar el proceso de compra, con el usuario cliente**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 173: Prototipo de Interacción carrito de compras**



## **DESARROLLO**

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 174: Controlador de Interacción carrito de compras

```
CartController.php x
app > Http > Controllers > Frontoffice > CartController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Frontoffice;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Coupon;
7  use App\Models\CouponUsage;
8  use App\Models\DeliveryCentral;
9  use App\Models\DeliveryZone;
10 use App\Models\Setting;
11 use App\Models\StoreBranch;
12 use App\Utils\MarketUtil;
13 use App\Utils\ShoppingCart;
14 use GuzzleHttpClient;
15 use Illuminate\Http\Request;
16 use Symfony\Component\HttpFoundation\Exception\BadRequestException;
17
18 class CartController extends Controller {
19
20     public function index(){
21         $cart_util = new ShoppingCart();
22         $items = $cart_util->get_items();
23         $total = $cart_util->get_total();
24         $sub_total = $cart_util->get_sub_total();
25         $logged_in = auth()->check();
26         $client_data = (auth()->check() ? auth()->user()->client : null);
27         $delivery_zones = DeliveryZone::with('bounds')->get();
28         $delivery_central = DeliveryCentral::all();
29         return view('front-office.cart.index', compact('items', 'sub_total', 'total', 'logged_in', 'client_data', 'delivery_zones', 'delivery_central'));
30     }
31
32     public function add_item(Request $request){
33         $cart_util = new ShoppingCart();
34         $data = $request->all();
35
36         $validStock = true;
37         if(!$data["is_service"]){
38             $validStock = $cart_util->validateStock($data["id"], $data["quantity"], true);
39             $data["branch_id"] = ($validStock ? $validStock["branch_id"] : null);
40         }
41         if($validStock){
42             $cart_util->add_item($data);
43             $items = $cart_util->get_items();
44             $total = $cart_util->get_total();
45             $sub_total = $cart_util->get_sub_total();
46             $count = $cart_util->count();
47         }
48     }
49 }
```

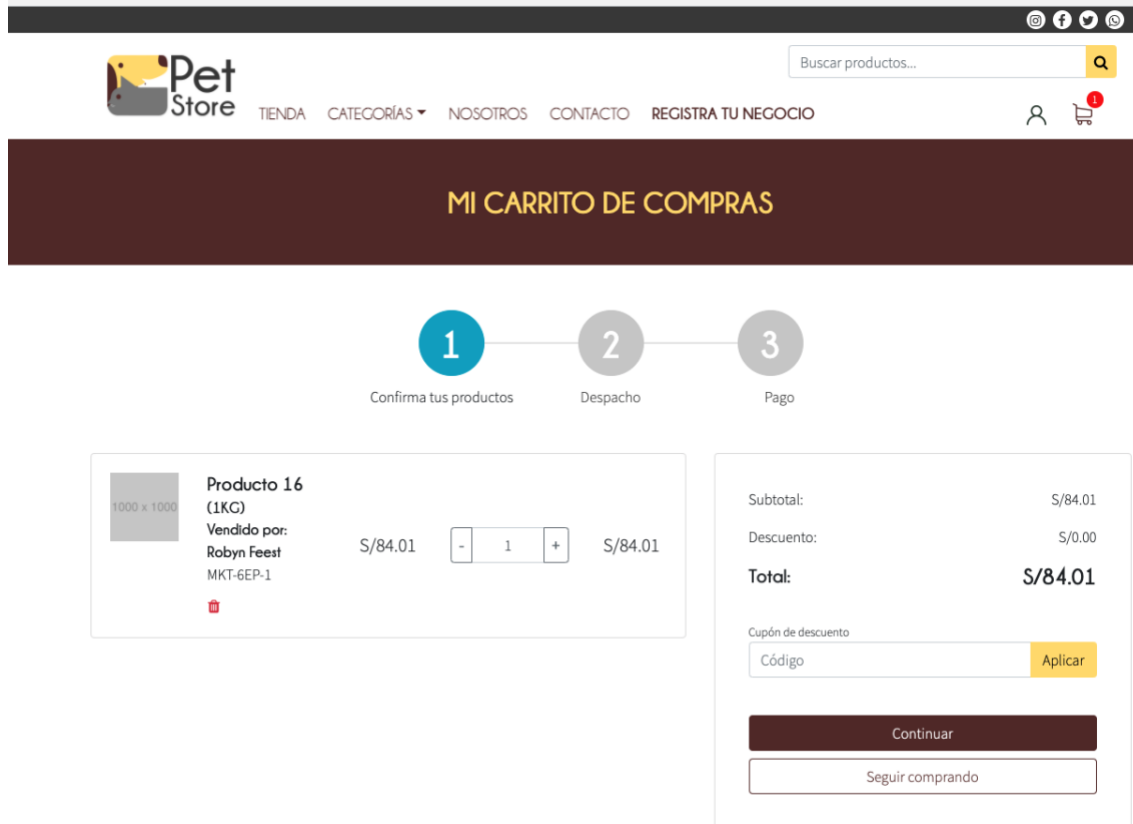
Figura\_ 175: Vista de Interacción carrito de compras

```
index.blade.php x
resources > views > front-office > cart > index.blade.php > ...
1 @extends('layouts.front-office.app')
2
3 @section('content')
4
5 <div class="banners-internos bg-brown d-flex justify-content-center align-items-center pt-lg-5 pt-3 pl-3 pr-3">
6   <div class="w-100 text-center">
7     <h2 class="champagne-limousines-bold text-beige text-uppercase fs-40 fs-xs-30 mb-0 wow fadeIn">
8   </div>
9 </div>
10 @php
11   $culqi_public_key = \App\Models\Setting::where("key", "CULQI_PUBLIC_KEY")->get()->first();
12   $culqi_public_key_value = $culqi_public_key["value"];
13
14   $google_maps_api_key = \App\Models\Setting::where("key", "GOOGLE_API_KEY")->get()->first();
15   $google_maps_api_key_value = $google_maps_api_key["value"];
16 @endphp
17 <div class="container">
18   <div class="row">
19     <div class="col-md-12 pt-lg-5 pt-3 pb-lg-5 pb-3">
20       <div class="cart-component">
21         data-currency-symbol="{{env('CURRENCY_SYMBOL')}}"
22         data-remove-item-url="{{route('cart.remove_item')}}"
23         data-update-item-url="{{route('cart.update_item')}}"
24         data-process-order-url="{{route('checkout.process')}}"
25         data-process-coupon-url="{{route('cart.validate.coupon')}}"
26         data-continue-shopping-url="{{route('store.all')}}"
27         data-get-address-by-placeid-url="{{route('cart.get.address.by.place.id')}}"
28         data-calculate-delivery-url="{{route('cart.calculate.delivery')}}"
29         data-get-service-hours-url="{{route('cart.service.hours')}}"
30         data-update-service-hour-url="{{route('cart.update.service.hour')}}"
31         data-validate-cart-url="{{route('cart.validate')}}"
32         data-items="{{json_encode($items)}}"
33         data-total="{{total}}"
34         data-sub-total="{{sub_total}}"
35         data-symbol="{{env('CURRENCY_SYMBOL')}}"
36         data-loggedin="{{logged_in}}"
37         data-client="{{client_data}}"
38         data-google-api-key="{{google_maps_api_key_value}}"
39         data-delivery-zones="{{json_encode($delivery_zones)}}"
40         data-delivery-centrals="{{json_encode($delivery_central)}}"
41         data-culqi-key="{{culqi_public_key_value}}"</div>
42       </div>
43     </div>
44   </div>
45 @endsection
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 176: Implementación de Interacción carrito de compras**



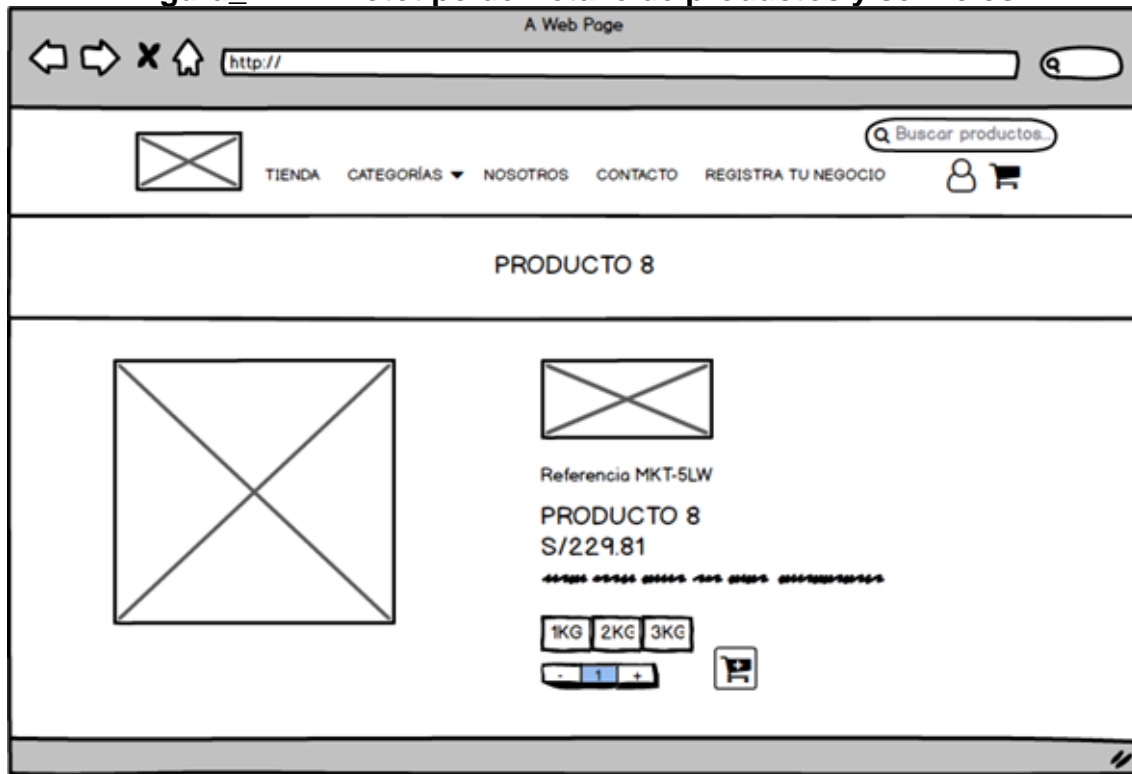
**RF28: El sistema permite al cliente poder visualizar el detalle de los productos o servicios con la finalidad de agregarlos a mi carrito de compras**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 177: Prototipo de Detalle de productos y servicios



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 178: Controlador de Detalle de productos y servicios

```
PageController.php X
app > Http > Controllers > Frontoffice > PageController.php > ...
76     if ($route_url["model"] == "product"){
77         $product = Product::with('images', 'features.values', 'variations.inventory', 'brand')->f
78         $category_ids = $product["categories"];
79         $category_ids = explode(",", $category_ids);
80         $product_util = new ProductUtil();
81         $relateds = $product_util->getRelateds($category_ids, $product["id"], $product["store_id"];
82         $title = $product["name"];
83         return view('front-office.product', compact('banner_url', 'title', 'product', 'relateds'));
84     }
85 }
```

Figura\_ 179: Modelo de Detalle de productos y servicios

```
Product.php x
app > Models > Product.php > Product > $fillable
use Illuminate\Database\Eloquent\{Product,
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Product extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'sku_code',
20         'description',
21         'summary',
22         'featured_image_id',
23
24         'categories',
25
26         'is_featured',
27
28         'price',
29         'iva',
30         'price_with_iva',
31
32         'offer_price',
33         'offer_iva',
34         'offer_price_with_iva',
35
36         'is_service',
37         'service attentions_per_hour',
38         'rate',
39
40         'brand_id',|
41         'brand_name',
42
43         'slug_route_id',
44         'url',
45
46         'store_id',
47         'store_name'
48     ];
```



Figura\_ 180: Vista de Detalle de productos y servicios

```
product.blade.php x
resources > views > front-office > product.blade.php > ...
1 @extends('layouts.front-office.app')
2
3 @section('content')
4
5 @if ($banner_url)
6 <div class="w-100">
7 
8 </div>
9
10 @else
11 <div class="banners-internos bg-brown d-flex justify-content-center align-items-center pt-lg-5 pt-3 pb-lg-5 pb-3" style-
12 <div class="w-100 text-center">
13 <h2 class="champagne-limousines-bold text-beige text-uppercase fs-40 fs-xs-30 mb-0 wow fadeInDown" data-wow-dela
14 </div>
15 @endif
16
17 <div class="container mt-lg-5 mt-3 mb-lg-5 mb-3 product-detail">
18 <div class="row">
19 <div class="col-md-12 bg-white p-5">
20 <div class="{{ $product["is_service"] ? "single-service-component" : "single-product-component" }}"
21 data-default-qty="{{ $product["default_qty"] }}"
22 data-product="{{ $product }}"
23 data-symbol="{{ env('CURRENCY_SYMBOL') }}"
24 data-add-cart-url="{{ route('cart.add_item') }}"></div>
25 </div>
26 </div>
27
28 @if ($product["description"])
29 <div class="row mt-3">
30 <div class="col-md-12 bg-white pt-3 pb-3">
31 <h4 class="champagne-limousines-bold text-brown mb-3">Descripción</h4>
32 <div class="w-100">
33 {!! $product["description"] !!}
34 </div>
35 </div>
36 </div>
37 @endif
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

Figura\_ 181: Implementación de Detalle de servicios

Pet Store TIENDA CATEGORÍAS ▾ NOSOTROS CONTACTO REGISTRA TU NEGOCIO

Buscar productos...

## SERVICIO 3

Referencia: MKT-9WG

### Servicio 3

S/32.87

Suscipit architecto molestias voluptatem adipisci commodi quo. Nisi maiores et asperiores fugit.

Agregar servicio

#### Descripción

Aliquam ex et magnam accusantium facere necessitatibus id. Ea sunt id quis ex sed, Temporibus omnis est deserunt dolor deserunt.

Figura\_ 182: Implementación de Detalle de Productos

Pet Store TIENDA CATEGORÍAS ▾ NOSOTROS CONTACTO REGISTRA TU NEGOCIO

Buscar productos...

## PRODUCTO 1

Referencia: MKT-9PL

### Producto 1

S/230.24

Ipsum impedit perferendis qui earum inventore atque. Est ipsa illo reprehenderit suscipit.

1KG 2KG 3KG

- 1 +

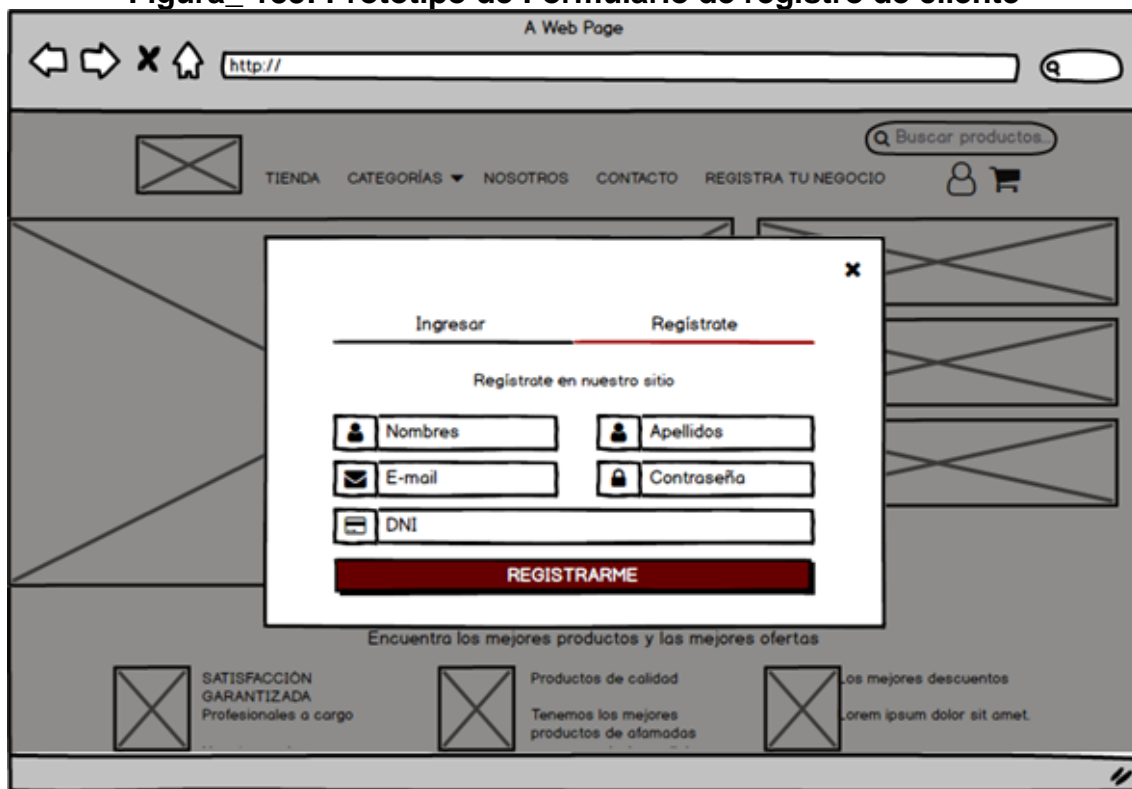
**RF29: El sistema permite que los clientes puedan registrarse para poder realizar las compras**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 183: Prototipo de Formulario de registro de cliente**



## **DESARROLLO**

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura 184: Controlador de Formulario de registro de cliente

```
RegisterController.php x
app > Http > Controllers > Frontoffice > RegisterController.php > ...
1 |?php
2
3 namespace App\Http\Controllers\Frontoffice;
4
5 use App\Http\Controllers\Controller;
6 use App\Mail\ClientRegister;
7 use App\Models\Client;
8 use App\Models\User;
9 use Illuminate\Http\Request;
10 use Illuminate\Support\Facades\Mail;
11 use Illuminate\Support\Str;
12
13 class RegisterController extends Controller {
14
15     public function register_process(Request $request) {
16         $data = $request->all();
17
18         $exist_user = User::where("email", $data["email"])->get()->first();
19         if($exist_user) {
20             return response()->json([
21                 'status'=>'error',
22                 'message' => "El Correo electrónico ingresado ya existe, por favor intente con uno diferente."
23             ]);
24         }else {
25             $user = User::create([
26                 'first_name' => $data["first_name"],
27                 'last_name' => $data["last_name"],
28                 'email' => $data["email"],
29                 'password' => $data["password"]
30             ]);
31
32             $token = (string) Str::uuid();
33
34             $client = Client::create([
35                 'document_type' => "DNI",
36                 'document_number' => $data["document_number"],
37                 'status' => "PENDING",
38                 'user_id' => $user->id,
39                 'verify_token' => $token,
40                 'verified' => false
41             ]);
42
43             $model = Client::find($client->id);
44             Mail::to($user["email"])->send(new ClientRegister($model));
45
46             $redirect_url = request()->headers->get('referer');
```

**Figura\_ 185: Modelo de Formulario de registro de cliente**

```
app > Models > Client.php > Client
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\SoftDeletes;
8
9  class Client extends Model {
10
11     use HasFactory;
12
13     use SoftDeletes;
14
15     protected $dates = ["deleted_at"];
16
17     protected $fillable = [
18         'id',
19         'cellphone',
20         // DNI | CEDULA | PASAPORTE
21         'document_type',
22         'document_number',
23         'address',
24         "user_id",
25         // N: NATURAL - J: JURIDICO
26         'type',
27         'ruc',
28         'business_name',
29         // PENDING | BLOCKED | CONFIRM
30         'status',
31         'verified',
32         'verify_token'
33     ];
```

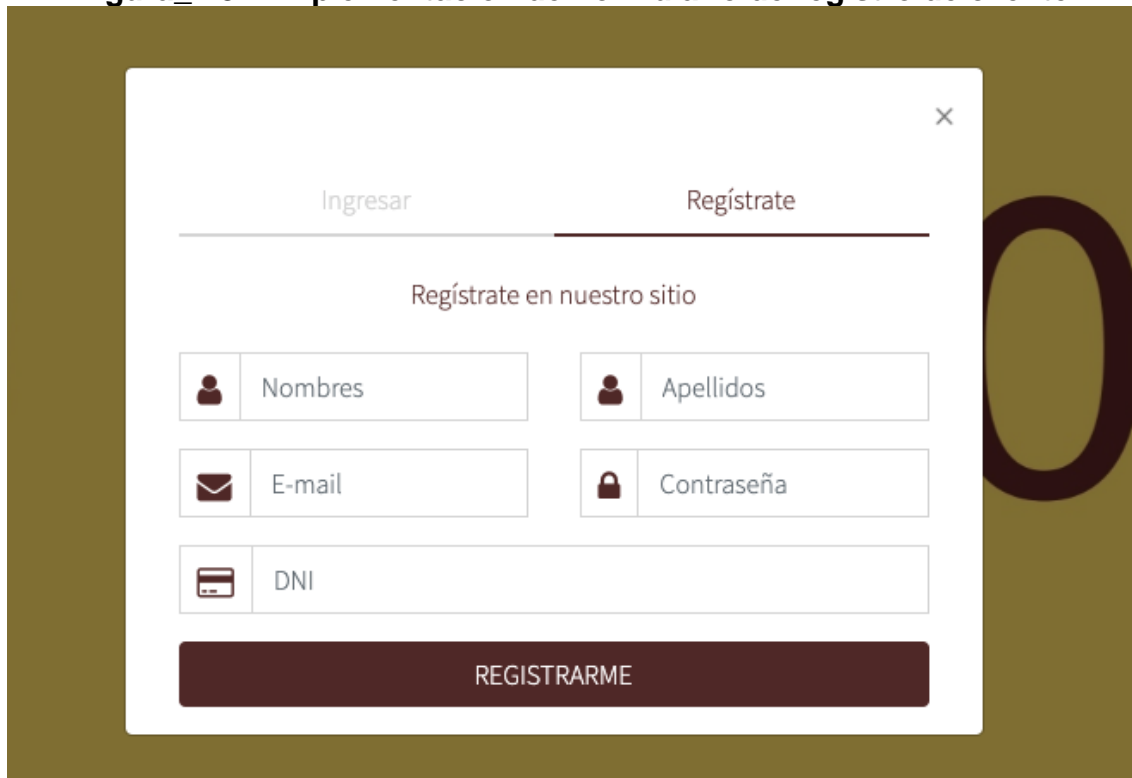
**Figura\_ 186: Vista de Formulario de registro de cliente**

```
82     data-login-url="{{route('store.client.login.process')}}"></div>
83 </div>
84 <div class="tab-pane fade" id="register" role="tabpanel" aria-labelledby="reg">
85     <h6 class="text-brown text-center">Regístrate en nuestro sitio</h6>
86     <div class="client-register-component"
87         data-register-url="{{route('store.client.register.process')}}"></div>
88 </div>
89 </div>
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 187: Implementación de Formulario de registro de cliente**



El formulario de registro de cliente se muestra en un modal con un fondo oscuro. En la parte superior, hay dos pestañas: "Ingresar" y "Regístrate", con "Regístrate" seleccionada. Debajo de las pestañas, se encuentra el texto "Regístrate en nuestro sitio". El formulario contiene los siguientes campos de entrada:

- Un campo para "Nombres" con un ícono de persona.
- Un campo para "Apellidos" con un ícono de persona.
- Un campo para "E-mail" con un ícono de correo electrónico.
- Un campo para "Contraseña" con un ícono de candado.
- Un campo para "DNI" con un ícono de tarjeta de identificación.

Debajo de los campos, hay un botón rojo con el texto "REGISTRARME".

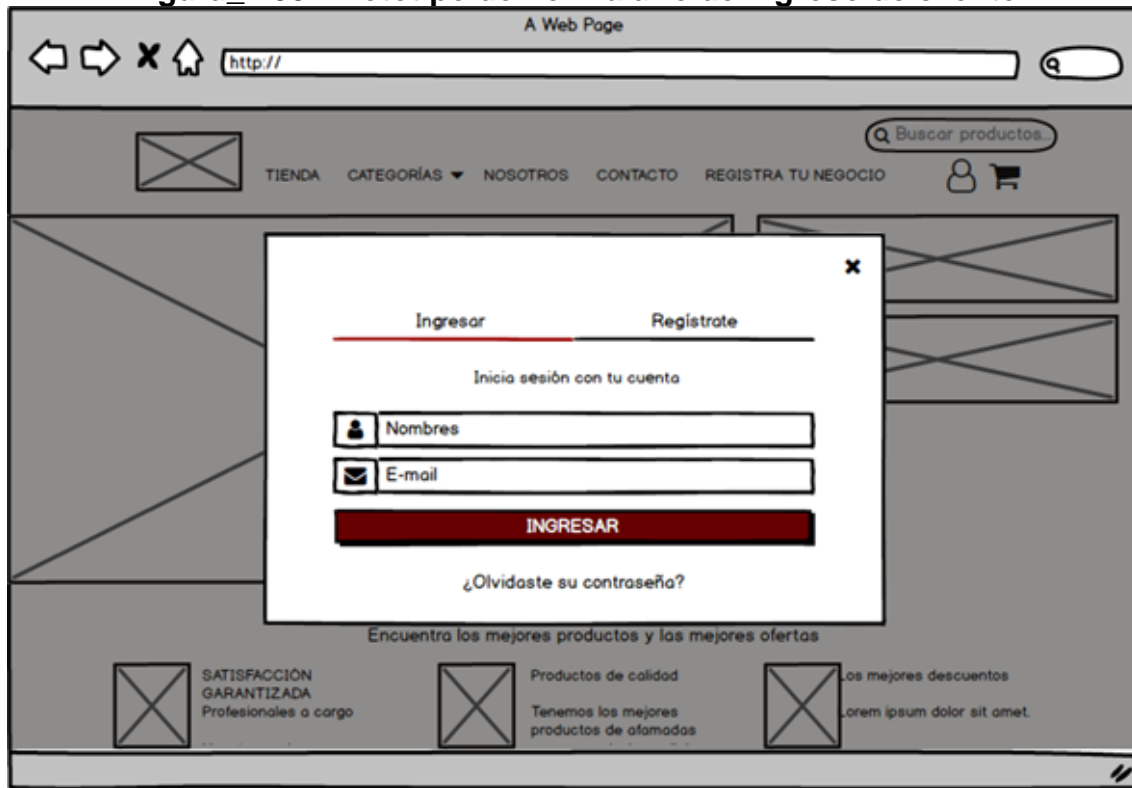
**RF30: El sistema permite que los usuarios registrados puedan iniciar sesión.**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 188: Prototipo de Formulario de ingreso de cliente**



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 189: Controlador de Formulario de ingreso de cliente

```
AuthController.php M X
app > Http > Controllers > Frontoffice > AuthController.php > ...
1 <?php
2
3 namespace App\Http\Controllers\Frontoffice;
4
5 use App\Http\Controllers\Controller;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Auth;
8
9 class AuthController extends Controller {
10
11     public function login_process(Request $request) {
12         $validator = $request->validate([
13             'email' => "required",
14             'password' => "required"
15         ]);
16
17         if (Auth::attempt($validator)){
18             $redirect_url = route('client.profile');
19             if(request()->headers->get('referer') != env('APP_URL')){
20                 $redirect_url = request()->headers->get('referer');
21             }
22
23             return response()->json([
24                 'status'=>'ok',
25                 'redirect_url' => $redirect_url
26             ]);
27         }else {
28             return response()->json([
29                 'status'=>'error',
30                 'message' => "Los datos ingresados son incorrectos."
31             ]);
32         }
33     }
34 }
35 }
36
```

Figura\_ 190: Modelo de Formulario de ingreso de cliente

```
Client.php X
app > Models > Client.php > Client
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Client extends Model {
10
11     use HasFactory;
12
13     use SoftDeletes;
14
15     protected $dates = ["deleted_at"];
16
17     protected $fillable = [
18         'id',
19         'cellphone',
20         // DNI | CEDULA | PASAPORTE
21         'document_type',
22         'document_number',
23         'address',
24         "user_id",
25         // N: NATURAL - J: JURIDICO
26         'type',
27         'ruc',
28         'business_name',
29         // PENDING | BLOCKED | CONFIRM
30         'status',
31         'verified',
32         'verify_token'
33     ];
34 }
```



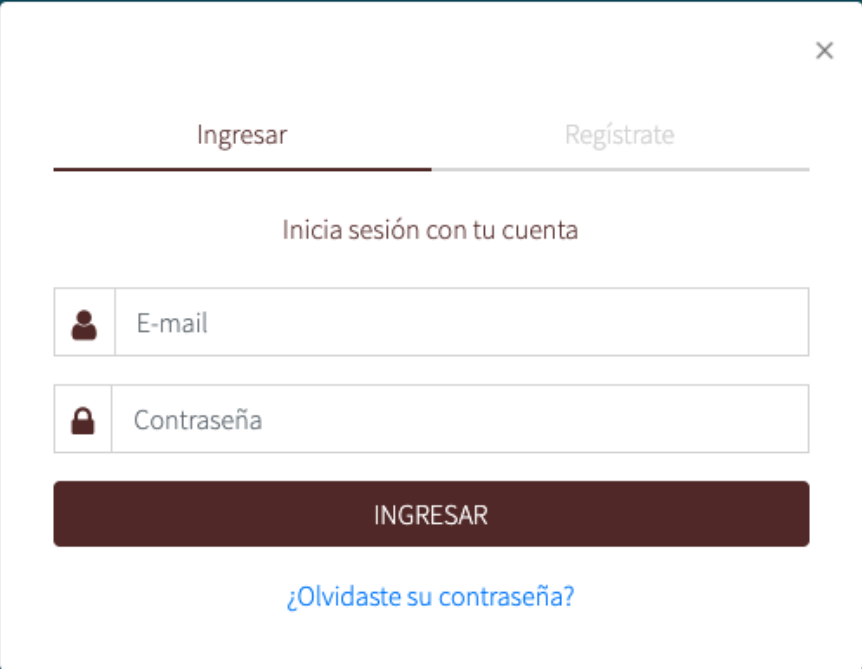
**Figura 191: Vista de Formulario de ingreso de cliente**

```
resources > views > layouts > front-office > footer.blade.php > ...
59 <div class="modal-content dr-z0">
60 <div class="modal-header border-bottom-0">
61 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
62 <span aria-hidden="true">&times;</span>
63 </button>
64 </div>
65 <div class="modal-body pt-0">
66 <div class="container">
67 <div class="row mb-lg-4 mb-2">
68 <div class="col-md-12">
69 <nav class="nav nav-pills nav-justified" role="tablist">
70 <a class="nav-item nav-link active" id="login-tab" data-toggle="tab" href="#login">
71 <a class="nav-item nav-link" id="register-tab" data-toggle="tab" href="#register">
72 </nav>
73 </div>
74 </div>
75 <div class="row">
76 <div class="col-md-12">
77 <div class="tab-content" id="myTabContent">
78 <div class="tab-pane fade show active" id="login" role="tabpanel" aria-labelledby="login-tab">
79 <h6 class="text-brown text-center">Inicia sesión con tu cuenta</h6>
80 <div class="client-login-component"
81 data-reset-password-url="{{route('password.request')}}"
82 data-login-url="{{route('store.client.login.process')}}"></div>
83 </div>
84 <div class="tab-pane fade" id="register" role="tabpanel" aria-labelledby="register-tab">
85 <h6 class="text-brown text-center">Regístrate en nuestro sitio</h6>
86 <div class="client-register-component"
87 data-register-url="{{route('store.client.register.process')}}"></div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 192: Implementación de Formulario de ingreso de cliente**



The image shows a login form mockup. At the top, there are two tabs: 'Ingresar' (selected) and 'Regístrate'. Below the tabs is the text 'Inicia sesión con tu cuenta'. There are two input fields: 'E-mail' with a person icon and 'Contraseña' with a lock icon. Below the fields is a dark blue button labeled 'INGRESAR'. At the bottom, there is a blue link that says '¿Olvidaste su contraseña?'. The form is set against a dark teal background with a large white 'O' on the right side.

**RF31: El sistema permite que el cliente pueda visualizar y actualizar sus datos en la sección de “mi perfil”**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 193: Prototipo de Perfil de cliente

El prototipo muestra una interfaz de usuario para un perfil de cliente. En la parte superior, hay una barra de navegación con un ícono de correo electrónico, los enlaces 'TIENDA', 'CATEGORÍAS', 'NOSOTROS', 'CONTACTO' y 'REGISTRA TU NEGOCIO', un campo de búsqueda con el texto 'Buscar productos...', un ícono de usuario y un ícono de carrito. El título principal de la página es 'MI PERFIL'. A la izquierda, hay dos botones: 'Perfil' y 'Historial de pedidos'. El formulario principal, titulado 'Actualiza tus datos', contiene los siguientes campos:

Nombres	Apellidos
Jhon	Doe
DNI	Celular
47804233	95037541
E-mail	Contraseña
jhondoe@gmail.com	Contraseña

Debajo de los campos de contraseña hay un botón 'GUARDAR'.

## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 194: Controlador de Perfil de cliente

```
ClientController.php M X
app > Http > Controllers > Frontoffice > ClientController.php > ...
1 <?php
2
3 namespace App\Http\Controllers\Frontoffice;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Client;
7 use App\Models\Order;
8 use App\Models\User;
9 use Illuminate\Http\Request;
10
11 class ClientController extends Controller {
12
13     public function profile(){
14         $client = auth()->user()->client;
15         $client["first_name"] = $client["user"]["first_name"];
16         $client["last_name"] = $client["user"]["last_name"];
17         $client["email"] = $client["user"]["email"];
18
19         $client_fields = [
20             "first_name" => [
21                 "title" => "Nombres",
22                 "sub_text" => null,
23                 "required" => true,
24                 "readonly" => false,
25                 "show" => true,
26                 "type" => "text",
27                 "options" => [],
28                 "error" => null,
29                 "width" => 6
30             ],
31             "last_name" => [
32                 "title" => "Apellidos",
33                 "sub_text" => null,
34                 "required" => true,
35                 "readonly" => false,
36                 "show" => true,
37                 "type" => "text",
38                 "options" => [],
39                 "error" => null,
40                 "width" => 6
41             ],
42             'document_number' => [
43                 "title" => "Número de documento",
44                 "sub_text" => null,
45                 "required" => true,
46                 "readonly" => false,
47                 "show" => true,
48                 "type" => "text",
49                 "options" => [],
50                 "error" => null,
51                 "width" => 6
52             ]
53         ];
54     }
55 }
```

Figura\_ 195: Modelo de Perfil de cliente

```
Client.php x
app > Models > Client.php > Client
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Client extends Model {
10
11     use HasFactory;
12
13     use SoftDeletes;
14
15     protected $dates = ["deleted_at"];
16
17     protected $fillable = [
18         'id',
19         'cellphone',
20         // DNI | CEDULA | PASAPORTE
21         'document_type',
22         'document_number',
23         'address',
24         'user_id',
25         // N: NATURAL - J: JURIDICO
26         'type',
27         'ruc',
28         'business_name',
29         // PENDING | BLOCKED | CONFIRM
30         'status',
31         'verified',
32         'verify_token'
33     ];
34
35     protected $appends = ["first_name", "last_name", "email"];
36
```

Figura\_ 196: Vista de Perfil de cliente

```
profile.blade.php x
resources > views > front-office > client > profile.blade.php > ...
1 @extends('layouts.front-office.app')
2
3 @section('content')
4
5 <div class="banners-internos bg-brown d-flex justify-content-center align-items-center pt-lg-5 pt-3 pb-lg-5 pt-3">
6     <div class="w-100 text-center">
7         <h2 class="champagne-limousines-bold text-beige text-uppercase fs-40 fs-xs-30 mb-0 wow fadeInDown" data-wow-duration="1s">
8             Perfil de cliente
9         </h2>
10    </div>
11 </div>
12
13 <div class="container">
14     <div class="row mt-lg-5 mt-3 mb-lg-5 mb-3">
15         <div class="col-md-3">
16             <p><a href="#" class="btn btn-brown btn-block">Perfil</a></p>
17             <p><a href="{{route('client.orders')}}" class="btn btn-brown-outline btn-block">Historial de pedidos</a></p>
18         </div>
19         <div class="col-md-9">
20             <div class="card">
21                 <div class="card-header bg-beige">
22                     <div class="col-md-12 d-flex align-items-center">
23                         <h5 class="champagne-limousines-bold mb-0 text-brown">Actualiza tus datos</h5>
24                     </div>
25                 </div>
26                 <div class="card-body">
27                     <div class="client-profile-component"
28                         data-fields="{{json_encode($client_fields)}}"
29                         data-client="{{json_encode($client)}}"
30                         data-update-profile-url="{{route('client.profile.update')}}">
31                     </div>
32                 </div>
33             </div>
34         </div>
35     </div>
36 </div>
37 @endsection
```

## IMPLEMENTACIÓN

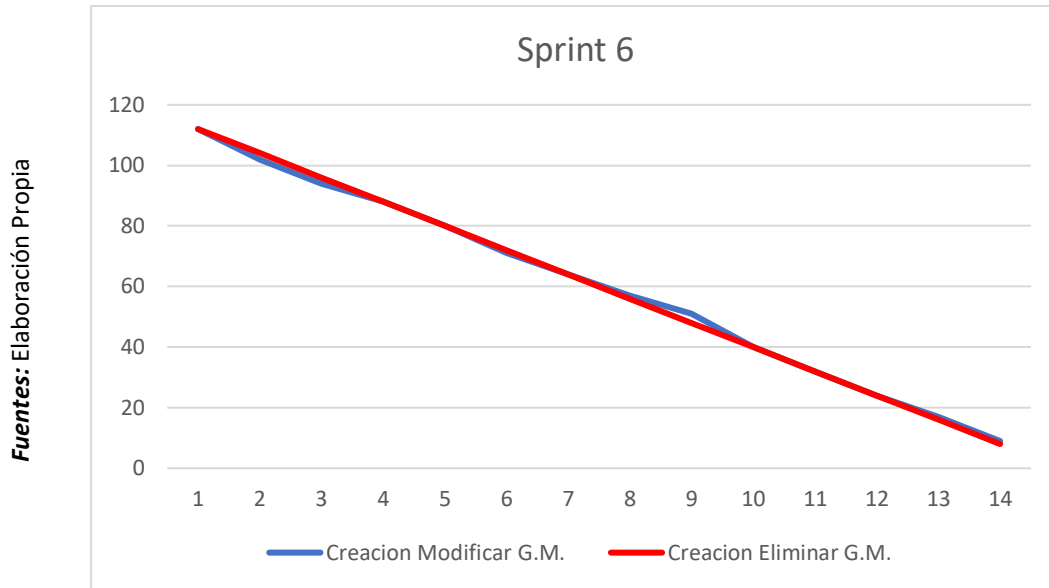
A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 197: Implementación de Perfil de cliente**

The screenshot displays the 'MI PERFIL' (My Profile) section of the Pet Store website. The page features a dark blue header with the Pet Store logo and navigation links: TIENDA, CATEGORÍAS, NOSOTROS, CONTACTO, and REGISTRA TU NEGOCIO. A search bar is located in the top right corner. Below the header, the 'MI PERFIL' title is centered in a dark blue banner. The main content area is divided into two columns. The left column contains two buttons: 'Perfil' (highlighted in dark blue) and 'Historial de pedidos'. The right column is titled 'Actualiza tus datos' and contains a form with the following fields: Nombres (Jhon), Apellidos (Doe), DNI (47804233), Celular (950375741), E-mail (jhondoe@gmail.com), and Contraseña (Contraseña). A yellow 'GUARDAR' button is positioned at the bottom of the form. The footer of the page includes a search bar, links to 'Términos y condiciones' and 'Políticas de privacidad', contact information (info@petstore.pe), and the Pet Store logo.

## BURNDOWN DEL SPRINT N° 6

**Figura\_ 198: Burndown del Sprint 6**



### Burndown Sprint 6

En la figura vista, se muestra el proceso de desarrollo de cada sprint la cual está representada a través de un gráfico, como podemos observar tenemos dos líneas la cual la línea roja representa al tiempo ideal que se define para el desarrollo del sprint, de igual manera la línea azul representa el tiempo real que tomo el desarrollo del sprint. Ya teniendo bien definido las representaciones tenemos que concluir que si la línea azul se encuentra por debajo de la roja significa que se presentó un adelanto, de lo contrario si la línea azul está por encima es decir que hubo un retraso.

## ACTA DE REUNION DEL SPRINT N° 6

Siendo las 3 pm del día 4 de agosto del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.


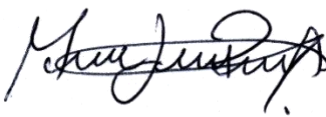


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El encargado de la Empresa AC Brands & Consulting E.I.R.L., Isaac Abensur Vargas concluye la reunión con los Sres. Marzio Perez y Nicolás Segovia aclarando los últimos puntos sobre el sprint.

Para determinar las formas de trabajo y los requerimientos respectivos para esta interacción, también la fecha de entrega y firmar el acta para dar por terminado la reunión.

Los participantes manifestaron su aprobación respecto a todos los puntos definidos en la reunión, que fueron de ayuda a la planificación del Sprint, manifestando su compromiso para la entrega de los resultados en las fechas pactadas.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza



## ACTA DE ENTREGA DEL SPRINT N° 6


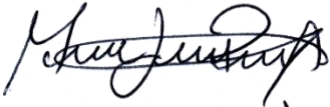


Siendo la 1 pm del día 20 de agosto del 2021 se reúne en la empresa AC Brands & Consulting E.I.R.L.

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

Los Sres. Marzio Perez y Nicolás Segovia, especifican cada uno de los requerimientos desarrollados, también presentan las interfaces elaboradas y brindadas por el producto Owner.

Luego de ser verificadas las presentaciones realizadas por los Srs. Marzio Perez y Nicolás Segovia, se decidió de manera unánime, dar por aprobado el término del Sprint respectivo, del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Los presentes manifestarán su aprobación al informe presentado por los Sres. Marzio Perez y Nicolás Segovia del Sprint N° 6 terminado del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 6

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	AC Brands & Consulting E.I.R.L.
<b>Proyecto</b>	“Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Información de la reunión:

<b>Lugar</b>	AC Brands & Consulting E.I.R.L.
<b>Fecha</b>	20/08/2021
<b>Número de iteración / Sprint</b>	Sprint 6
<b>Personas Convocadas a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza
<b>Persona que asistieron a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
Se desarrolló la página de inicio Se logró desarrollar el carrito de compras. Se desarrolló el detalle de productos y servicios. Se desarrolló en formulario de registro e ingreso de cliente. Se desarrolló el perfil de cliente.	Se presentaron dificultades para mostrar el detalle de los productos y servicios por categorías

## PLANIFICACIÓN DEL SPRINT N°7

Siendo las 4 pm del día 20 de agosto del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.

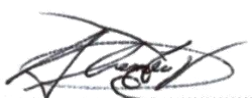
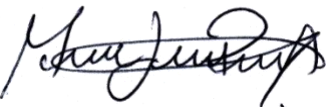


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El gerente de la Empresa AC Brands & Consulting E.I.R.L., efectúa la presentación de los requerimientos, de igual manera manifiesta los requerimientos que tienen mayor prioridad.

Realizado el análisis de los requerimientos presentados por el gerente de AC Brands & Consulting E.I.R.L., el señor Juan Galarza aclara dudas existentes y luego se compromete al cumplimiento de todos los requerimientos propuestas en el Sprint 7.

Los participantes de la reunión manifiestan su aprobación con respecto a la presentación del Sprint 7, así mismo especificando que la fecha de entrega será el día 31 de agosto del 2021.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## EJECUCIÓN DEL SPRINT 7

Lo primero que se debe realizar antes del diseño, es conocer y comprender el funcionamiento del sistema, en otras palabras, analizar lo que realmente se necesita, lo cual será elaborado según las historias de usuario.

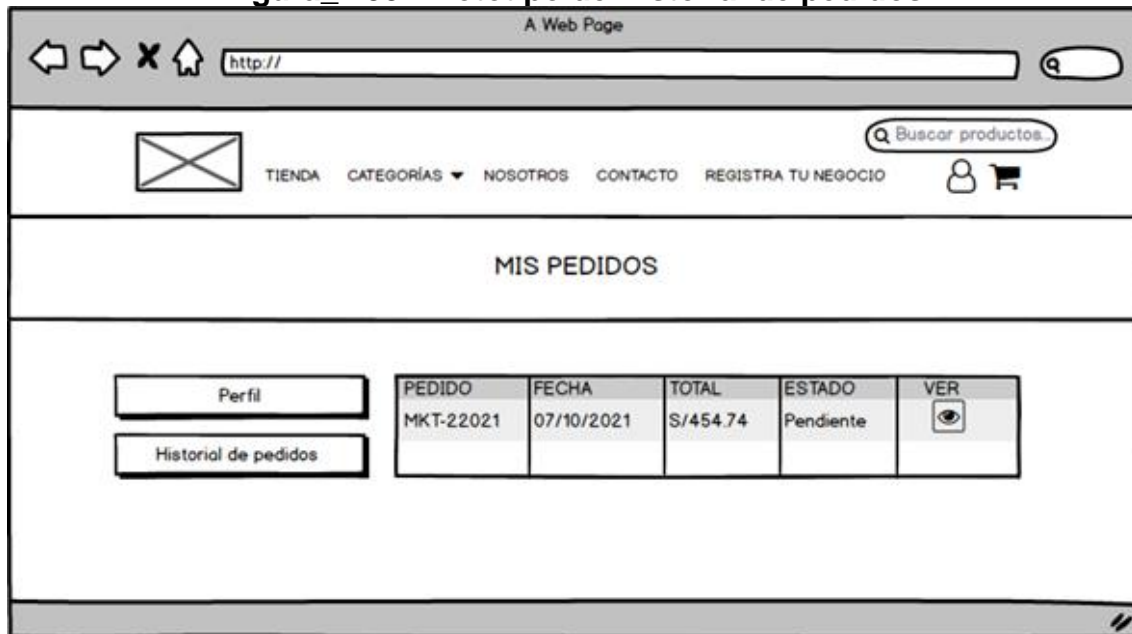
**RF32: El sistema permite que el cliente pueda visualizar sus pedidos, con el fin de visualizar el detalle de cada uno**

### DISEÑO

#### PROTOTIPO

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 199: Prototipo de Historial de pedidos**



### DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.



Figura\_ 202: Vista de Historial de pedidos

```
orders.blade.php x
resources > views > front-office > client > orders.blade.php > ...
1 @extends('layouts.front-office.app')
2
3 @section('content')
4
5 <div class="banners-internos bg-brown d-flex justify-content-center align-items-center pt-lg-5 pt-3 pb-lg-5 pt
6 <div class="w-100 text-center">
7 <h2 class="champagne-limousines-bold text-beige text-uppercase fs-40 fs-xs-30 mb-0 wow fadeInDown" dat
8 </div>
9 </div>
10
11 <div class="container">
12 <div class="row mt-lg-5 mt-3 mb-lg-5 mb-3">
13 <div class="col-md-3">
14 <p><a href="{{route('client.profile')}}" class="btn btn-brown-outline btn-block">Perfil</a></p>
15 <p><a href="#" class="btn btn-brown btn-block">Historial de pedidos</a></p>
16 </div>
17 <div class="col-md-9">
18 <div class="table-responsive">
19 <table class="table table-bordered bg-white">
20 <thead>
21 <tr>
22 <th class="champagne-limousines-bold text-uppercase text-center">Pedido</th>
23 <th class="champagne-limousines-bold text-uppercase text-center">Fecha</th>
24 <th class="champagne-limousines-bold text-uppercase text-center">Total</th>
25 <th class="champagne-limousines-bold text-uppercase text-center">Estado</th>
26 <th class="champagne-limousines-bold text-uppercase text-center">Ver</th>
27 </tr>
28 </thead>
29 <tbody>
30 @foreach ($orders as $order)
31 <tr>
32 <td class="text-center">
33 <a href="{{route('client.order.detail', ['code' => $order['order_code']])}}
34 </td>
35 <td class="text-center">{{date("d/m/Y", strtotime($order["created_at"])}</td>
36 <td class="text-center">{{env('CURRENCY_SYMBOL') . number_format($order["total
37 <td class="text-center">
38 @if($order["order_status"] === "PENDING")
39 <span class="text-warning champagne-limousines-bold">Pendiente</span>
40 @endif
41 @if($order["order_status"] === "ATTENDED")
42 <span class="text-success champagne-limousines-bold">Atendida</span>
43 @endif
44 @if($order["order_status"] === "CANCELED")
45 <span class="text-danger champagne-limousines-bold">Cancelada</span>
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

Figura\_ 203: Implementación de Historial de pedidos



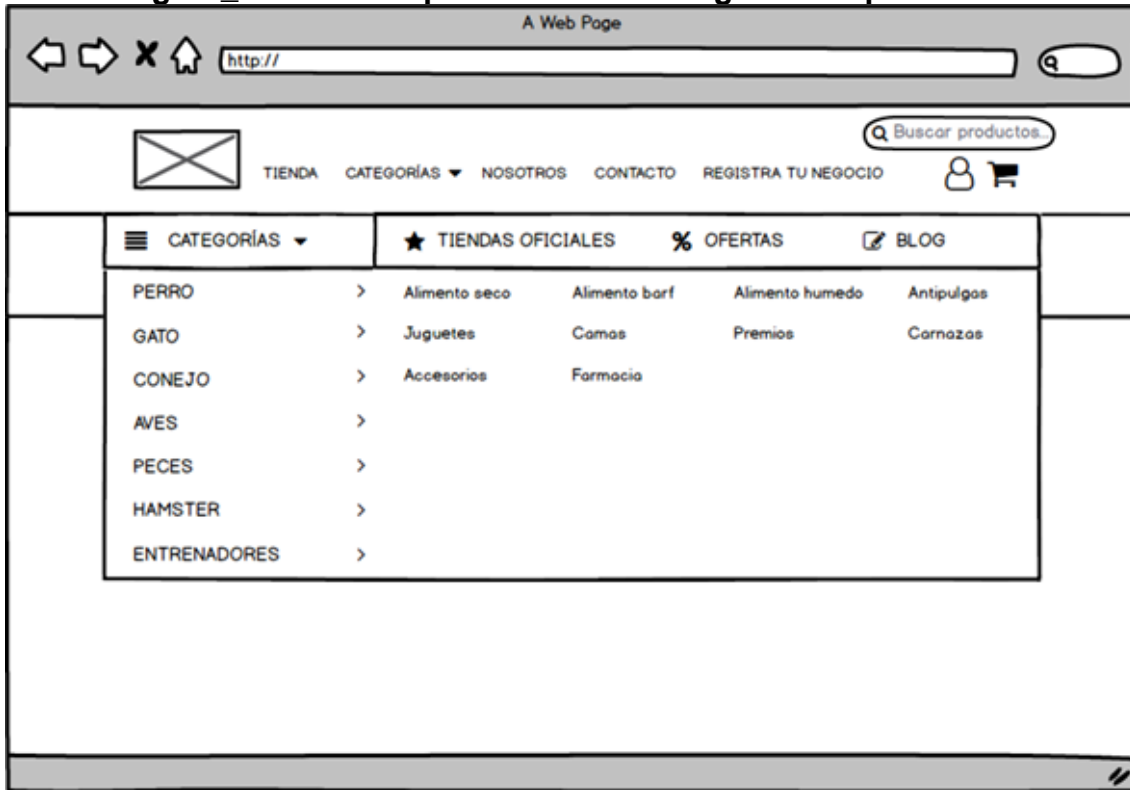
**RF33: El sistema permite que el cliente pueda visualizar las categorías de los productos con el fin de poder encontrar el/los productos o servicios que desea adquirir.**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 204: Prototipo de Vista de categorías de productos



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.



Figura\_ 205: Controlador de Vista de categoría de productos

```
12 use App\Utils\ProductUtil;
13 use Illuminate\Http\Request;
14
15 class PageController extends Controller {
16
17     public function index(Request $request) {
18         $parts = $request->segments();
19         if (count($parts) > 0){
20             $slug = implode("/", $parts);
21             $route_url = SlugRoute::where("slug", $slug)->get()->first();
22
23             $banner_url = null;
24             $parent_category_id = null;
25             $brand_id = null;
26             $category_id = null;
27             $brand_ids = [];
28
29             if ($route_url["model"] === "category"){
30                 $category = Category::find($route_url["model_id"]);
31
32                 if($category["is_child"]){
33                     $parent_category = Category::find($category["category_id"]);
34                     $parent_category_id = $parent_category["id"];
35                 }
36
37                 $brands = Brand::orderBy("name", "ASC")->get();
38
39                 $model = Product::query();
40                 $model->with('variations.inventory');
41                 $filter = null;
42
43                 if($request->has("sort_by")){
44                     $filter = $request->get("sort_by");
45
46                     if ($filter === "min-price") {
47                         $model->orderBy('price', 'ASC');
48                     }
49                 }
50             }
51         }
52     }
53 }
```

Figura\_ 206: Modelo de Vista de categoría de productos

```
Category.php x
app > Models > Category.php > Category
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class Category extends Model {
10
11     use HasFactory;
12     use SoftDeletes;
13
14     protected $dates = ["deleted_at"];
15
16     protected $fillable = [
17         'id',
18         'name',
19         'internal_name',
20         'is_child',
21         // articles - products
22         "type",
23         'category_id',
24         'featured_image_id',
25         "slug_route_id"
26     ];
27 }
```

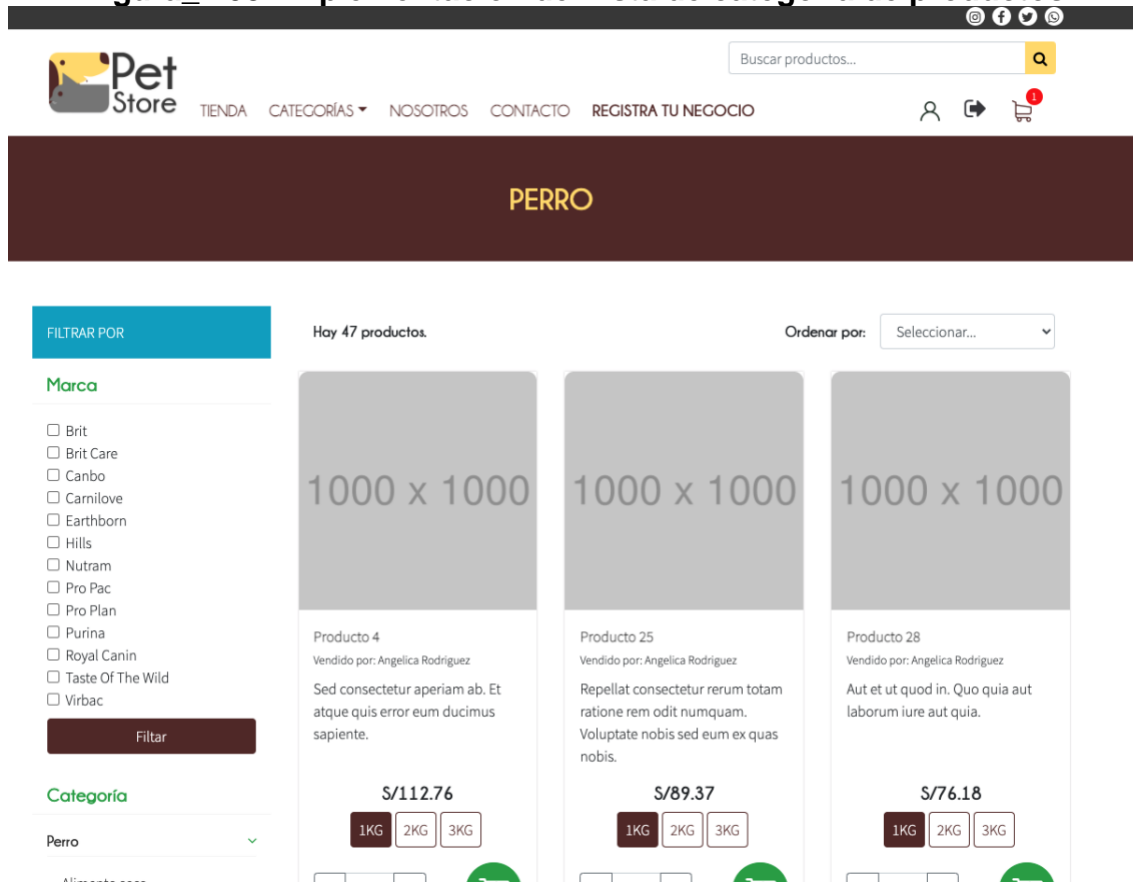
Figura\_ 207: Vista de Vista de categoría de productos

```
category.blade.php x
resources > views > front-office > category.blade.php > ...
1 @extends('layouts.front-office.app')
2
3 @section('content')
4
5 @if ($banner_url)
6 <div class="w-100">
7 
8 </div>
9
10 @else
11 <div class="banners-internos bg-brown d-flex justify-content-center align-items-center pt-lg-5 pt-3 pb-lg-5 pb-3">
12 <div class="w-100 text-center">
13 <h2 class="champagne-limousines-bold text-beige text-uppercase fs-40 fs-xs-30 mb-0 wow fadeInDown">
14 </div>
15 </div>
16 @endif
17
18 <div class="container mt-lg-5 mt-3">
19 <div class="row">
20 <div class="col-md-3 d-lg-block d-none">
21 @component('layouts.front-office.sidebar', ["brand_ids" => $brand_ids, "category_id" => $category_id])
22 </div>
23 <div class="col-md-9">
24 @if(count($products) > 0)
25 <div class="w-100 bg-white pt-2 pb-2 pl-3 pr-3 mb-3">
26 <div class="row">
27 <div class="col-md-4 d-flex align-items-center">
28 <p class="text-black champagne-limousines-bold mb-0">Hay {{ $products->total() }} productos
29 </div>
30 <div class="col-md-8 d-flex align-items-center justify-content-md-end">
31 <div class="form-inline w-xs-100">
32 <div class="form-group w-xs-100">
33 <label class="text-black champagne-limousines-bold mb-0 mr-3" for="filter">Filtrar
34 <select name="page_filter" id="filter_page" class="form-control">
35 <option value="">Seleccionar...</option>
36 <option value="min-price" @if($filter === "min-price") selected @endif>Minimo
37 <option value="max-price" @if($filter === "max-price") selected @endif>Maximo
38 <option value="alphabetical" @if($filter === "alphabetical") selected @endif>Alfabetico
39 <option value="alphabetical_za" @if($filter === "alphabetical_za") selected @endif>Alfabetico (Z-A)
40 <option value="recommend" @if($filter === "recommend") selected @endif>Recomendado
41 </select>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 208: Implementación de Vista de categoría de productos**



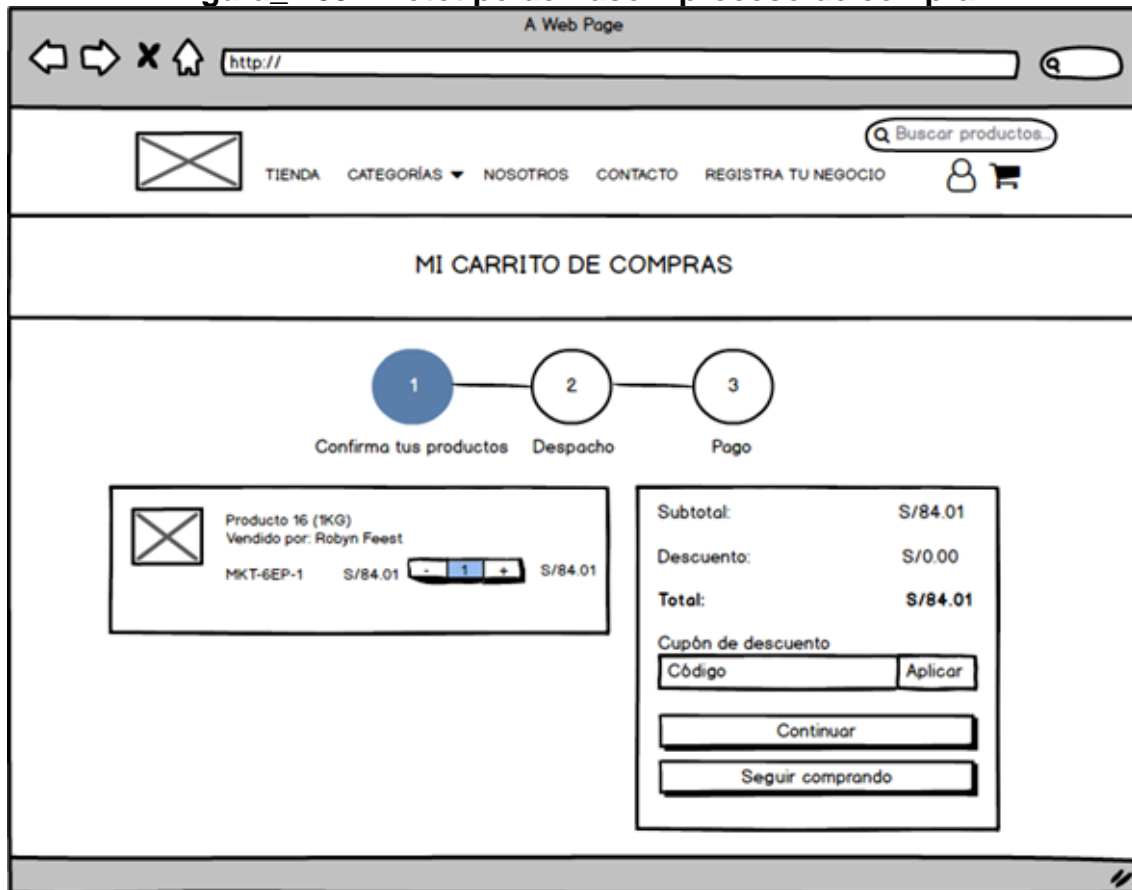
**RF34: El sistema permite realizar el primer paso del proceso de compras con el fin de poder validar los productos o servicios que va adquirir**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 209: Prototipo de Paso 1 proceso de compra



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 210: Controlador de Paso 1 proceso de compra

```
CartController.php x
app > Http > Controllers > Frontoffice > CartController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Frontoffice;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Coupon;
7  use App\Models\CouponUsage;
8  use App\Models\DeliveryCentral;
9  use App\Models\DeliveryZone;
10 use App\Models\Setting;
11 use App\Models\StoreBranch;
12 use App\Utils\MarketUtil;
13 use App\Utils\ShoppingCart;
14 use GuzzleHttpClient;
15 use Illuminate\Http\Request;
16 use Symfony\Component\HttpFoundation\Exception\BadRequestException;
17
18 class CartController extends Controller {
19
20     public function index(){
21         $cart_util = new ShoppingCart();
22         $items = $cart_util->get_items();
23         $total = $cart_util->get_total();
24         $sub_total = $cart_util->get_sub_total();
25         $logged_in = auth()->check();
26         $client_data = (auth()->check() ? auth()->user()->client : null);
27         $delivery_zones = DeliveryZone::with('bounds')->get();
28         $delivery_central = DeliveryCentral::all();
29         return view('front-office.cart.index', compact('items', 'sub_total', 'total', 'logged_in', 'client_data', 'delivery_zones', 'delivery_central'));
30     }
31
32     public function add_item(Request $request){
33         $cart_util = new ShoppingCart();
34         $data = $request->all();
35
36         $validStock = true;
37         if(!$data["is_service"]){
38             $validStock = $cart_util->validateStock($data["id"], $data["quantity"], true);
39             $data["branch_id"] = ($validStock ? $validStock["branch_id"] : null);
40         }
41         if($validStock){
42             $cart_util->add_item($data);
43             $items = $cart_util->get_items();
44             $total = $cart_util->get_total();
45             $sub_total = $cart_util->get_sub_total();
46             $count = $cart_util->count();
47         }
48     }
49 }
```

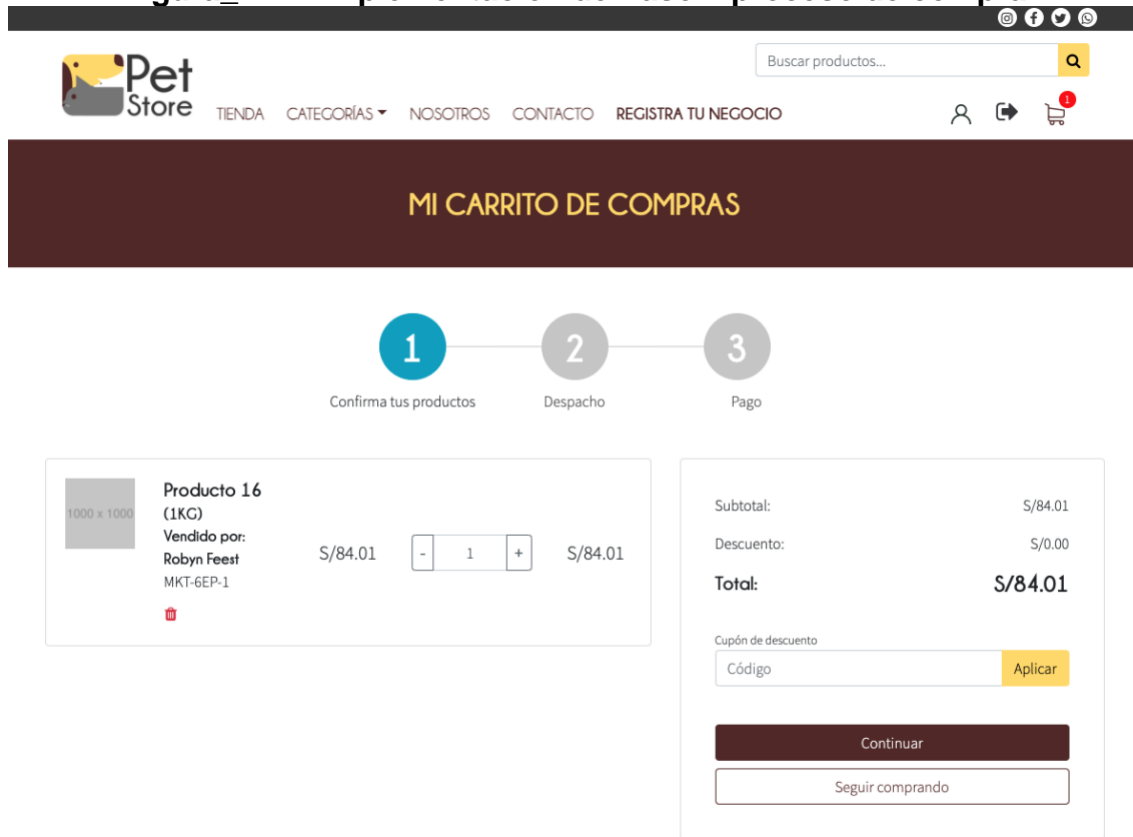
Figura\_ 211: Vista de Paso 1 proceso de compra

```
JS CartComponent.js x
resources > assets > front-office > js > components > cart > JS CartComponent.js > CartComponent > showDeliveryMap
962 ((this.state.items.length > 0) ?
963 <div className="col-md-12">
964 <div className="row">
965   /* Paso 1 */
966   <div className={"col-md-7" + (this.state.current_step === 1 ? " d-block" : " d-none")}>
967     {this.state.items.map((row, ri) => {
968       return <div key={ri} className={"card mb-2" + (!row.attributes.available ? " card-danger" : "")}>
969         <div className="card-body">
970           <div className="row">
971             <div className="col-md-2 col-6">
972               <img src={row.attributes.image_url} alt={row.name} className="w-100" />
973             </div>
974             <div className="col-md-3 col-6 d-flex align-items-center justify-content-center">
975               <div className="w-100">
976                 <h5 className="mb-0 champagne-limousines-bold">{row.name} {!!row.attributes.is_se
977                 <p className="mb-0"><b className="champagne-limousines-bold">{!!row.attributes.is
978                 <p className="mb-2">{row.attributes.sku_code}</p>
979                 <button className="btn p-0" type="button" onClick={(e)=> {this.deleteItem(row)}}>
980                   {(this.state.loading_remove && this.state.loading_item_id === row.id) ? <em c
981                 </button>
982               </div>
983             </div>
984             <div className="col-md-2 d-lg-flex d-none align-items-center justify-content-center">
985               <h5 className="mb-0">{this.symbol}{Number(row.price).toLocaleString('en', amount_opti
986             </div>
987             <div className="col-md-3 col-12 d-flex align-items-center justify-content-center">
988               {(row.attributes.is_service) ?
989                 <div className="w-100">
990                   <div className="mb-1">
991                     <label htmlFor={"date-" + row.id} className="mb-0 champagne-limousines-bo
992                     <DatePicker selected={(row.attributes.service_date ? new Date(row.attribu
993                       className="form-control input-date-booking"
994                       locale="es"
995                       onChange={(v) => this.handleDateChange(row, v)}
996                       excludeDates={[new Date("2021-01-01"), new Date("2021-12-25"), new Da
997                       id={"date-" + row.id}
998                       dateFormat="dd/MM/Y"
999                       autoComplete="new-date"
1000                       disabled={this.state.loading}
1001                       minDate={moment(new Date()).add(1, 'days').toDate()} />
1002                   </div>
1003                 </div>
1004                 <div>
1005                   <label htmlFor={"hour-" + row.id} className="mb-0 champagne-limousines-bo
1006                   <Select value={row.attributes.service_time}
1007                     className="form-control"
1008                     placeholder="Seleccionar..."
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 212: Implementación de Paso 1 proceso de compra**



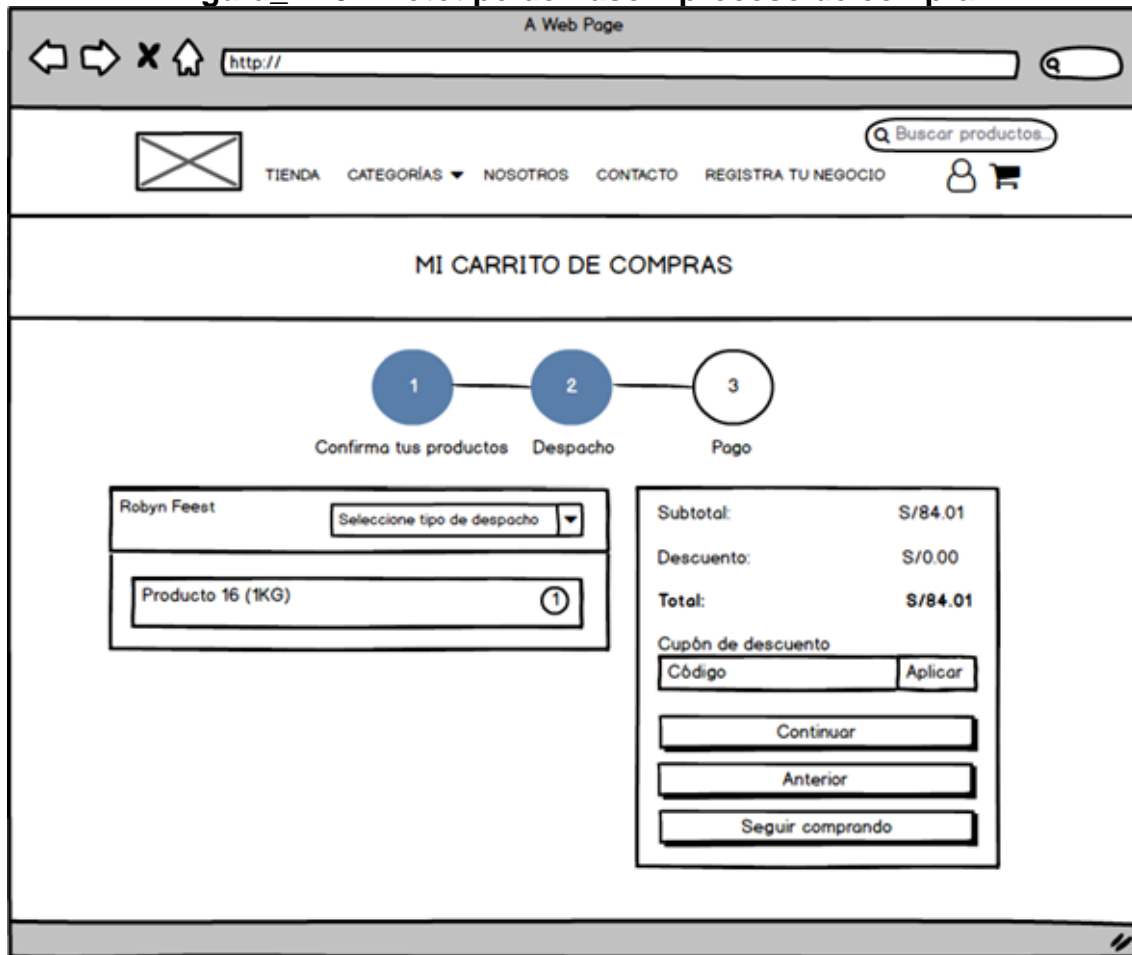
**RF35: El sistema permite realizar el segundo paso del proceso de compras con el fin de poder seleccionar la forma de envío de los productos de cada comercio que va adquiriendo**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 213: Prototipo de Paso 2 proceso de compra



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.



Figura\_ 214: Controlador de Paso 2 proceso de compra

```
JS CartComponent.js x
resources > assets > front-office > js > components > cart > JS CartComponent.js > CartComponent > showDeliveryMap
962 ((this.state.items.length > 0) ?
963 <div className="col-md-12">
964 <div className="row">
965   /* Paso 1 */
966   <div className={"col-md-7" + (this.state.current_step === 1 ? " d-block" : " d-none")}>
967     {this.state.items.map((row, ri) => {
968       return <div key={ri} className={"card mb-2" + (!row.attributes.available ? " card-danger" : "")}>
969         <div className="card-body">
970           <div className="row">
971             <div className="col-md-2 col-6">
972               <img src={row.attributes.image_url} alt={row.name} className="w-100" />
973             </div>
974             <div className="col-md-3 col-6 d-flex align-items-center justify-content-center">
975               <div className="w-100">
976                 <h5 className="mb-0 champagne-limousines-bold">{row.name} {(!row.attributes.is_se
977                 <p className="mb-0"><b className="champagne-limousines-bold">{(!row.attributes.is
978                 <p className="mb-2">{row.attributes.sku_code}</p>
979                 <button className="btn p-0" type="button" onClick={(e)=> {this.deleteItem(row)}}>
980                   {(this.state.loading_remove && this.state.loading_item_id === row.id) ? <em c
981                 </button>
982               </div>
983             </div>
984             <div className="col-md-2 d-lg-flex d-none align-items-center justify-content-center">
985               <h5 className="mb-0">{this.symbol}{Number(row.price).toLocaleString('en', amount_opti
986             </div>
987             <div className="col-md-3 col-12 d-flex align-items-center justify-content-center">
988               {(row.attributes.is_service) ?
989                 <div className="w-100">
990                   <div className="mb-1">
991                     <label htmlFor={"date-" + row.id} className="mb-0 champagne-limousines-bo
992                     <DatePicker selected={(row.attributes.service_date ? new Date(row.attribu
993                       className="form-control input-date-booking"
994                       locale="es"
995                       onChange={(v) => this.handleDateChange(row, v)}
996                       excludeDates={[new Date("2021-01-01"), new Date("2021-12-25"), new Da
997                       id={"date-" + row.id}
998                       dateFormat="dd/MM/Y"
999                       autoComplete="new-date"
1000                       disabled={this.state.loading}
1001                       minDate={moment(new Date()).add(1, 'days').toDate()} />
1002                   </div>
1003                 </div>
1004                 <div>
1005                   <label htmlFor={"hour-" + row.id} className="mb-0 champagne-limousines-bo
1006                   <Select value={row.attributes.service_time}
1007                     className="form-control"
1008                     placeholder="Seleccionar..."
```

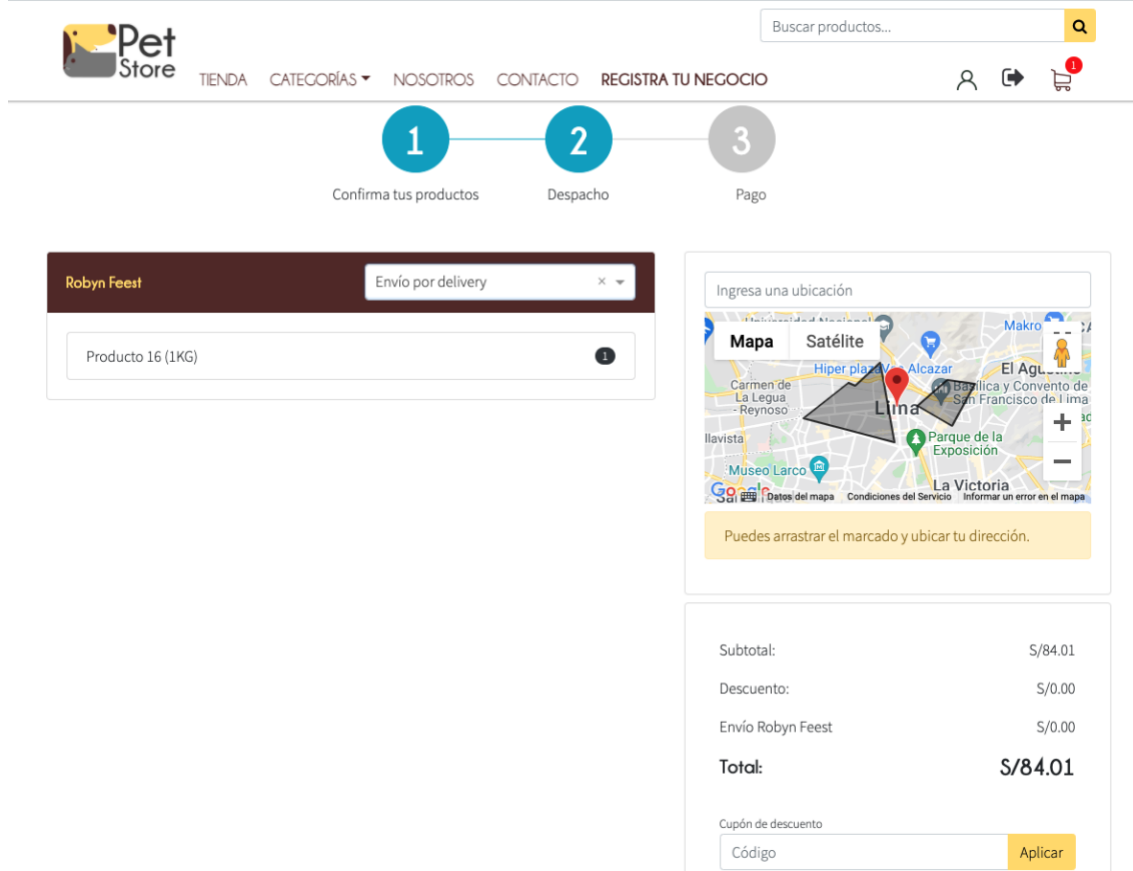
Figura\_ 215: Vista de Paso 2 proceso de compra

```
JS CartComponent.js x
resources > assets > front-office > js > components > cart > JS CartComponent.js > CartComponent > showDeliveryMap
1041
1042 /* Paso 2 */
1043 <div className={"col-md-7" + (this.state.current_step === 2 ? " d-block" : " d-none")}>
1044   {(this.logged_in) ?
1045     <div className="row">
1046       <div className="col-md-12">
1047         {(this.state.delivery_store_items.length > 0) ?
1048           this.state.delivery_store_items.map((store, b) => {
1049             return <div className="w-100 card mb-2" key={b}>
1050               <div className="card-header bg-brown">
1051                 <div className="row">
1052                   <div className="col-md-6 d-flex align-items-center">
1053                     <p className="mb-0 champagne-limousines-bold text-beige">{store["name"]}</p>
1054                   </div>
1055                   {(store["is_trainer"] === 0) ?
1056                     <div className="col-md-6">
1057                       <Select value={store["delivery_type"]}
1058                         placeholder="Seleccione un tipo de despacho"
1059                         className="form-control"
1060                         onChange={(opt) => this.handleStoreFieldChange(b, "delivery_type", opt)}
1061                         disabled={this.state.loading}
1062                         options={store["delivery_client_types"].filter((sdt) => sdt["delivery_type"]
1063                           let obj = {};
1064                           obj["label"] = dt["delivery_type"]["name"];
1065                           obj["value"] = dt["delivery_type"]["code"];
1066                           return obj;
1067                         )} />
1068                     </div>
1069                   : ""}
1070                 </div>
1071               </div>
1072             <div className="card-body">
1073               {(store["delivery_type"] === "PICK-IN-CENTRAL") ?
1074                 <div className="mb-3">
1075                   <p>Selecciona una sede central para el recojo de tus productos:</p>
1076                   {this.delivery_centrales.map((delivery_central, dc) => {
1077                     return <div className="form-check" key={dc}>
1078                       <input className="form-check-input" onChange={(e) => this.handleStoreField
1079                         <label className="form-check-label" htmlFor={"dc-" + b + "-" + dc}>{deliv
1080                     </div>
1081                   })}
1082                 </div>
1083               : ""}
1084             <ul className="list-group">
1085               {store["items"].map((item, i) => {
1086                 return <li key={i} className="list-group-item d-flex justify-content-between align
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

**Figura\_ 216: Implementación de Paso 2 proceso de compra**



**RF36: El sistema permite realizar el tercer paso del proceso de compras con el fin de poder seleccionar el método de pago de interés**

## **DISEÑO**

### **PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

Figura\_ 217: Prototipo de Paso 3 proceso de compra

A Web Page

http://

TIENDA CATEGORÍAS ▼ NOSOTROS CONTACTO REGISTRA TU NEGOCIO

Buscar productos

MI CARRITO DE COMPRAS

1 Confirma tus productos 2 Despacho 3 Pago

DATOS DE PARA EL COMPROBANTE

Boleta  Factura

Jhon Jhon Jhon

SELECCIONA TU FORMA DE PAGO

Transferencia  Yape  Tarjeta

Siga los siguientes pasos:

1. Puede realizar el pago por medio de transferencia bancaria o en ventanilla a los siguientes números de cuenta o nombre de PET STORE SAC:

Continental:  
Soles: 0011-0191-0100042320

BCP:  
Soles: 193-2558169-0-89

2. Enviar la constancia de pago o transferencia al mail: [ventas@petstore.pe](mailto:ventas@petstore.pe) indicando:

- Nombres y Apellidos.
- RUC o DNI.
- Adjuntar imagen del voucher o constancia de transferencia.

Pet Store responderá dentro de las próximas 24 horas confirmando su pedido.

Acepto los [términos y condiciones](#)

Subtotal: S/84.01  
Descuento: S/0.00  
Total: S/84.01

Cupón de descuento

Código Aplicar

Continuar

Anterior

Seguir comprando

## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

Figura\_ 218: Controlador de Paso 3 proceso de compra

```
JS CartComponent.js x
resources > assets > front-office > js > components > cart > JS CartComponent.js > CartComponent > showDeliveryMap
962 ((this.state.items.length > 0) ?
963 <div className="col-md-12">
964 <div className="row">
965   /* Paso 1 */
966   <div className={"col-md-7" + (this.state.current_step === 1 ? " d-block" : " d-none")}>
967     {this.state.items.map((row, ri) => {
968       return <div key={ri} className={"card mb-2" + (!row.attributes.available ? " card-danger" : "")}>
969         <div className="card-body">
970           <div className="row">
971             <div className="col-md-2 col-6">
972               <img src={row.attributes.image_url} alt={row.name} className="w-100" />
973             </div>
974             <div className="col-md-3 col-6 d-flex align-items-center justify-content-center">
975               <div className="w-100">
976                 <h5 className="mb-0 champagne-limousines-bold">{row.name} {(!row.attributes.is_se
977                   <p className="mb-0"><b className="champagne-limousines-bold">{(!row.attributes.is
978                     <p className="mb-2">{row.attributes.sku_code}</p>
979                     <button className="btn p-0" type="button" onClick={(e)=> {this.deleteItem(row)}}>
980                       {(this.state.loading_remove && this.state.loading_item_id === row.id) ? <em c
981                     </button>
982                   </div>
983                 </div>
984               <div className="col-md-2 d-lg-flex d-none align-items-center justify-content-center">
985                 <h5 className="mb-0">{this.symbol}{Number(row.price).toLocaleString('en', amount_opti
986               </div>
987             <div className="col-md-3 col-12 d-flex align-items-center justify-content-center">
988               {(row.attributes.is_service) ?
989                 <div className="w-100">
990                   <div className="mb-1">
991                     <label htmlFor={"date-" + row.id} className="mb-0 champagne-limousines-bo
992                     <DatePicker selected={(row.attributes.service_date ? new Date(row.attribu
993                       className="form-control input-date-booking"
994                       locale="es"
995                       onChange={(v) => this.handleDateChange(row, v)}
996                       excludeDates={[new Date("2021-01-01"), new Date("2021-12-25"), new Da
997                       id={"date-" + row.id}
998                       dateFormat="dd/MM/Y"
999                       autoComplete="new-date"
1000                       disabled={this.state.loading}
1001                       minDate={moment(new Date()).add(1, 'days').toDate()} />
1002                   </div>
1003                 <div>
1004                   <label htmlFor={"hour-" + row.id} className="mb-0 champagne-limousines-bo
1005                   <Select value={row.attributes.service_time}
1006                     className="form-control"
1007                     placeholder="Seleccionar..."
```



**Figura\_ 220: Implementación de Paso 3 proceso de compra**

**DATOS DE PARA EL COMPROBANTE**

Boleta  Factura

**SELECCIONA TU FORMA DE PAGO**

Transferencia  Yape  Tarjeta

**Siga los siguientes pasos:**

1. Puede realizar el pago por medio de transferencia bancaria o en ventanilla a los siguientes números de cuenta a nombre de PET STORE SAC:

**Continental:**  
Soles: 0011-0191-0100042320

**BCP:**  
Soles: 193-2558169-0-89

2. Enviar la constancia de pago o transferencia al mail: [ventas@petstore.pe](mailto:ventas@petstore.pe) indicando:

- Nombres y Apellidos.
- RUC o DNI.
- Adjuntar imagen del voucher o constancia de transferencia.

Subtotal: S/84.01  
Descuento: S/0.00  
**Total: S/84.01**

Cupón de descuento

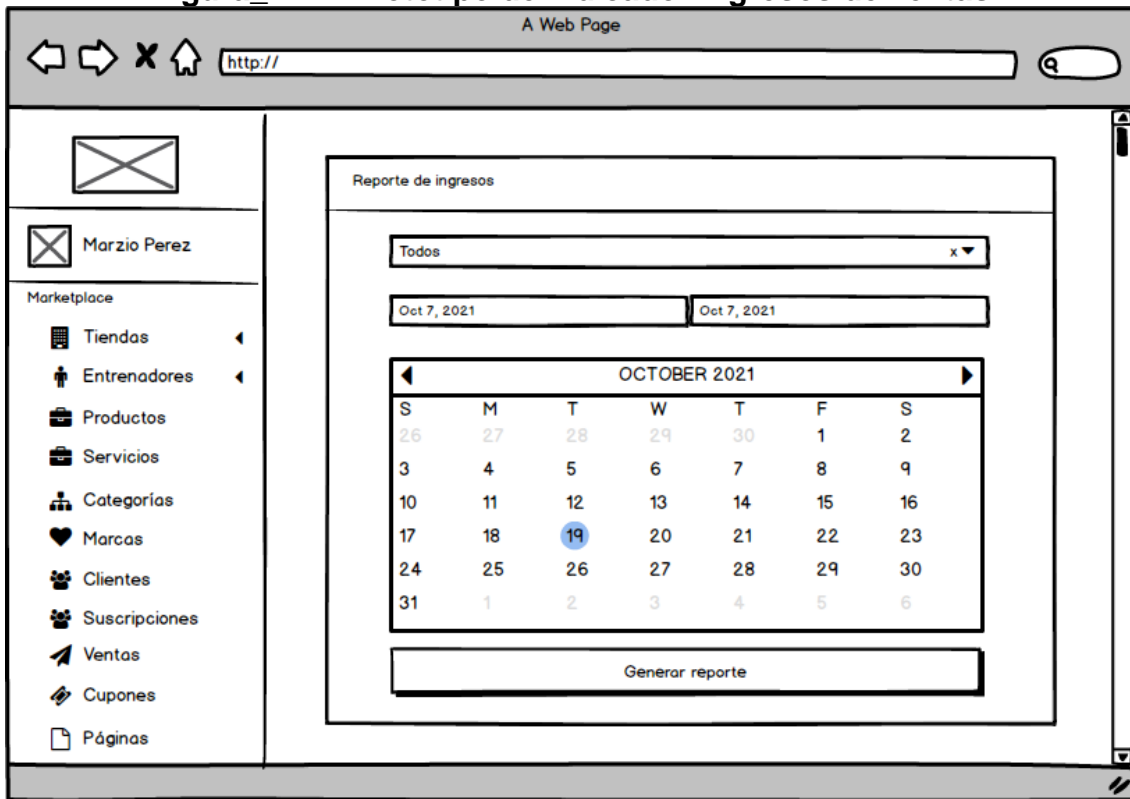
**RF37: El sistema permite generar el reporte de calidad de pedidos generados en un determinado periodo de tiempo con el fin de poder saber las comisiones que va a recibir.**

## DISEÑO

### PROTOTIPO

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura 221: Prototipo de Indicador Ingresos de ventas**



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

**Figura 222: Controlador de Indicador Ingresos de ventas**

```

15     protected $start_date;
16     protected $end_date;
17     protected $param;
18     protected $store_id;
19
20     public function __construct($start_date, $end_date, $param, $store_id = null) {
21         $this->start_date = $start_date;
22         $this->end_date = $end_date;
23         $this->param = $param;
24         $this->store_id = $store_id;
25     }
26
27     public function view(): View {
28         if($this->store_id) {
29             $model = OrderDetail::query();
30             $model->with('order');
31             $model->where('store_id', $this->store_id);
32             $model->whereBetween('created_at', [$this->start_date . " 00:00:00", $this->end_date . " 23:59:59"]);
33             $orders = $model->get();
34             return view('back-office.export.orders-store', ["orders" => $orders]);
35         } else {
36             $model = Order::query();
37             if($this->param != "ALL") {
38                 $model->where("order_status", $this->param);
39             }
40             $model->whereBetween('created_at', [$this->start_date . " 00:00:00", $this->end_date . " 23:59:59"]);
41             $orders = $model->get();
42             return view('back-office.export.orders', ["orders" => $orders]);
43         }
44     }
45 }

```



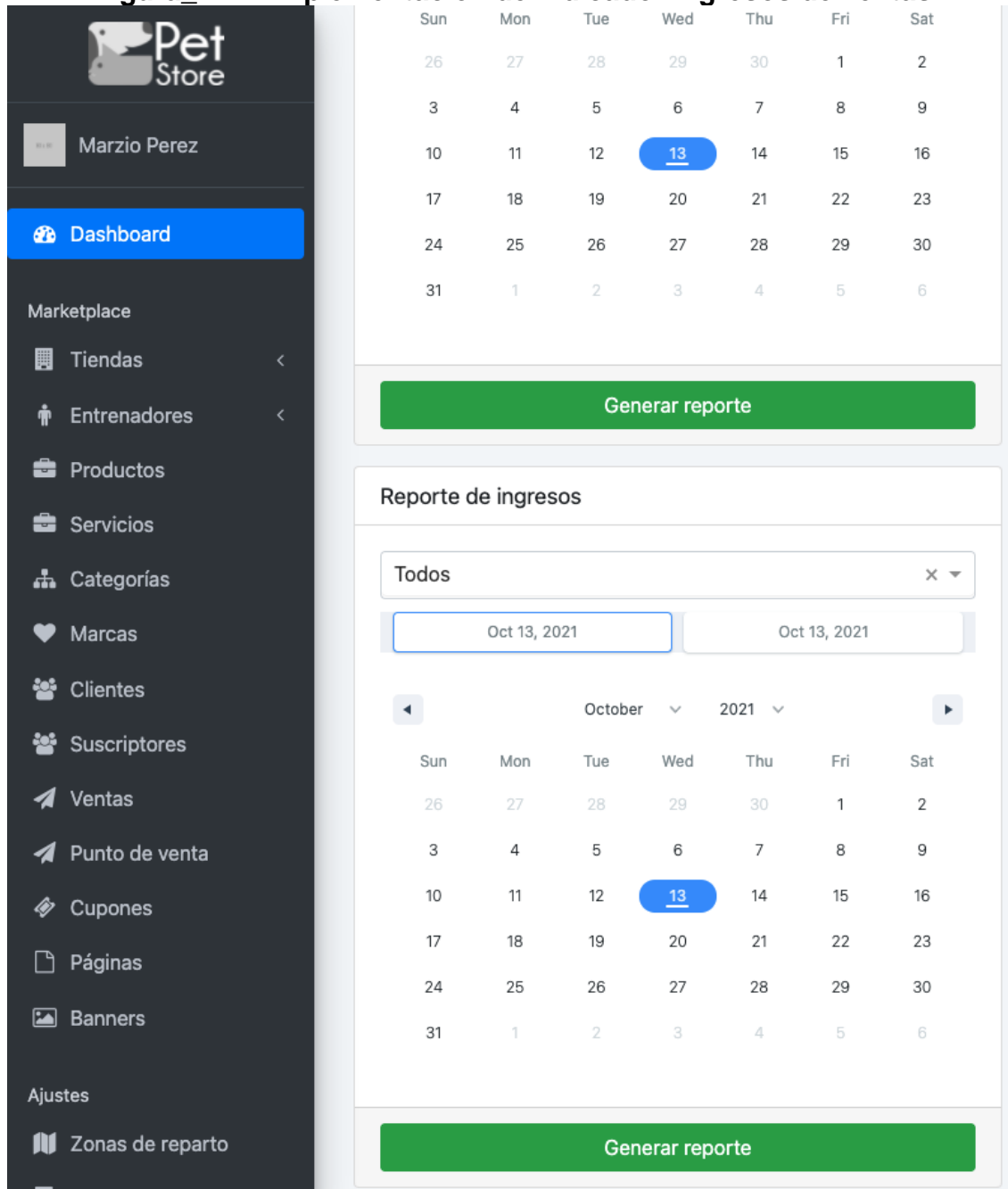
**Figura\_ 223: Vista de Indicador Ingresos de ventas**

```
resources > assets > back-office > js > components > report > JS ReportComponent.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import { DateRange } from 'react-date-range';
4  import moment from 'moment';
5  import Select from 'react-select';
6  import 'react-select/dist/react-select.css';
7
8  export default class ReportComponent extends React.Component {
9
10     constructor(props) {
11         super(props);
12         this.url = this.props.url;
13         this.params = JSON.parse(this.props.params);
14
15         this.state = {
16             date: [{startDate: new Date(), endDate: new Date(), key: 'selection'}],
17             param_id: "ALL"
18         };
19
20         this.handleChangeDate = this.handleChangeDate.bind(this);
21         this.handleSubmit = this.handleSubmit.bind(this);
22         this.handleFieldChange = this.handleFieldChange.bind(this);
23     }
24
25     handleChangeDate(range){
26         this.setState({date: [range.selection]});
27     }
28
29     handleFieldChange(field, value) {
30         let fieldUpdate = {};
31         fieldUpdate[field] = value;
```

## IMPLEMENTACIÓN

A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

Figura\_ 224: Implementación de Indicador Ingresos de ventas



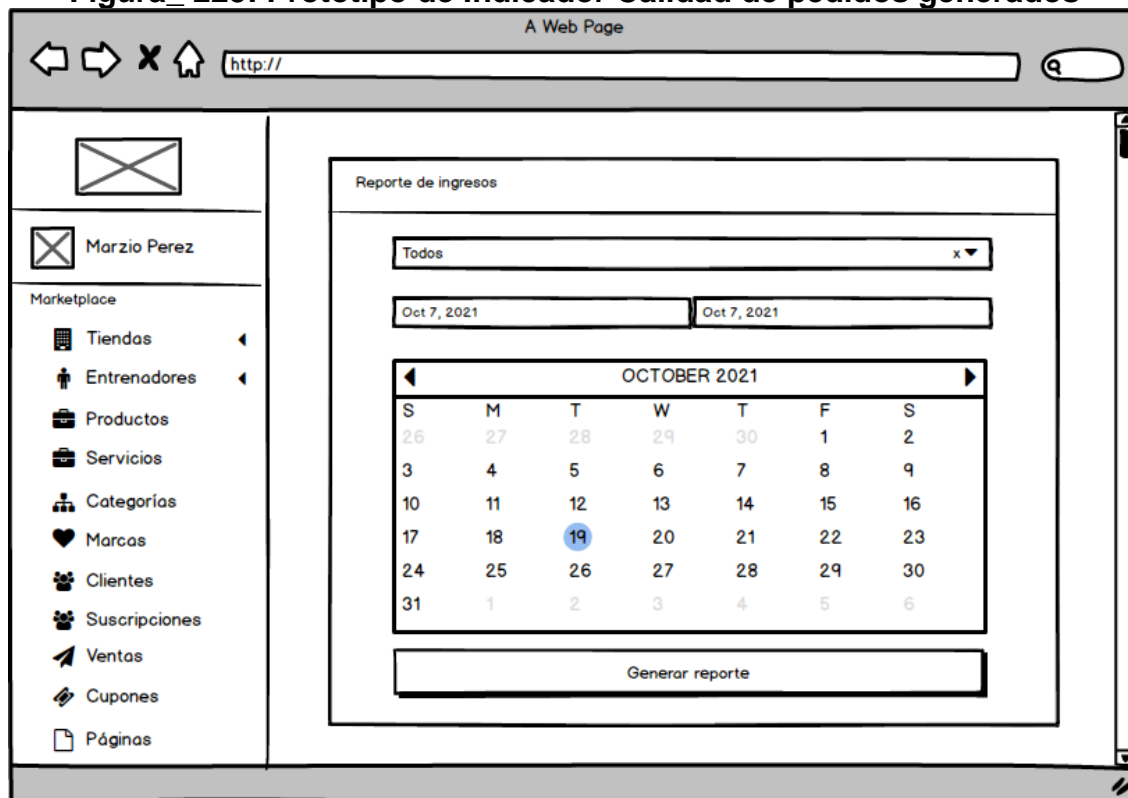
**RF38: El sistema permite generar el reporte de calidad de pedidos generados en un determinado periodo de tiempo con el fin de poder determinar si los pedidos se están entregando en las fechas pactadas.**

**DISEÑO**

**PROTOTIPO**

A continuación, se puede observar el prototipo hecho para el requerimiento señalado la cual fue previamente mostrado al product owner para ser aprobado. Además, el prototipo fue elaborado por el equipo de trabajo en el programa Balsamiq mockups.

**Figura\_ 225: Prototipo de Indicador Calidad de pedidos generados**



## DESARROLLO

A continuación, se presenta el desarrollo del sistema web, este empieza por el controlador, continuando por el modelo y para finalizar la vista.

**Figura\_ 226: Controlador de Indicador Calidad de pedidos generados**

```
app > Exports > OrdersExport.php > ...
1  <?php
2
3  namespace App\Exports;
4
5  use App\Models\Order;
6  use App\Models\OrderDetail;
7  use Illuminate\Contracts\View\View;
8  use Maatwebsite\Excel\Concerns\FromView;
9  use Maatwebsite\Excel\Concerns\Exportable;
10
11 class OrdersExport implements FromView {
12
13     use Exportable;
14
15     protected $start_date;
16     protected $end_date;
17     protected $param;
18     protected $store_id;
19
20     public function __construct($start_date, $end_date, $param, $store_id = null) {
21         $this->start_date = $start_date;
22         $this->end_date = $end_date;
23         $this->param = $param;
24         $this->store_id = $store_id;
25     }
26
27     public function view(): View {
28         if($this->store_id) {
29             $model = OrderDetail::query();
30             $model->with('order');
31             $model->where('store_id', $this->store_id);
```

**Figura\_ 227: Vista de Indicador Calidad de pedidos generados**

```
resources > assets > back-office > js > components > report > JS ReportComponent.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import { DateRange } from 'react-date-range';
4  import moment from 'moment';
5  import Select from 'react-select';
6  import 'react-select/dist/react-select.css';
7
8  export default class ReportComponent extends React.Component {
9
10     constructor(props) {
11         super(props);
12         this.url = this.props.url;
13         this.params = JSON.parse(this.props.params);
14
15         this.state = {
16             date: [{startDate: new Date(), endDate: new Date(), key: 'selection'}],
17             param_id: "ALL"
18         };
19
20         this.handleChangeDate = this.handleChangeDate.bind(this);
21         this.handleSubmit = this.handleSubmit.bind(this);
22         this.handleFieldChange = this.handleFieldChange.bind(this);
23     }
24
25     handleChangeDate(range){
26         this.setState({date: [range.selection]});
27     }
28
29     handleFieldChange(field, value) {
30         let fieldUpdate = {};
31         fieldUpdate[field] = value;
```

## IMPLEMENTACIÓN

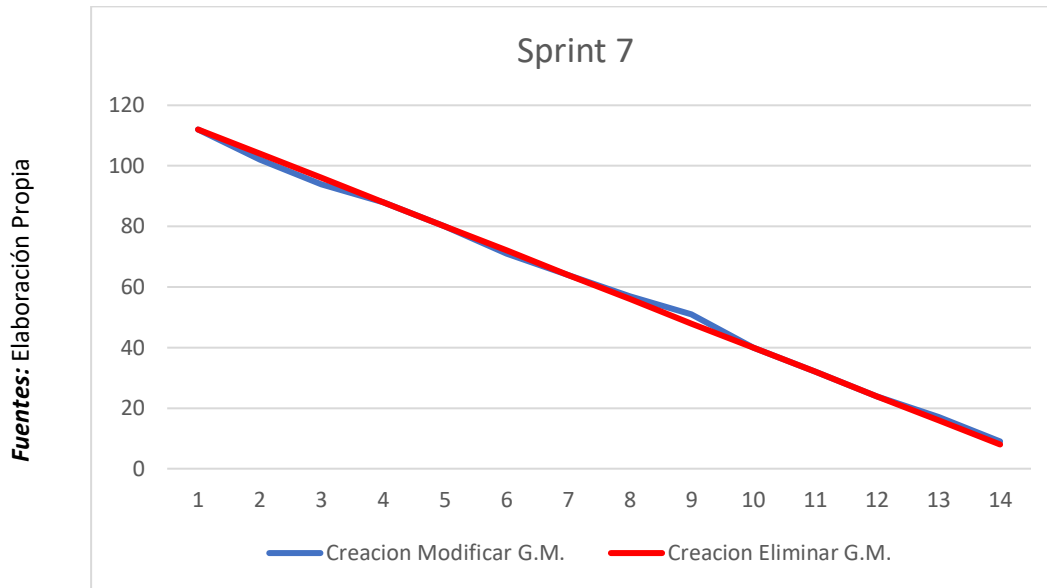
A continuación, se presenta la interfaz gráfica del requerimiento señalado, la cual fue previamente definida por el producto owner para que el equipo de trabajo pueda desarrollarlo.

Figura\_ 228: Implementación de Indicador Calidad de pedidos generados

The image shows a dashboard for 'Pet Store' with a user profile for 'Marzio Perez'. The left sidebar contains navigation options: Dashboard, Marketplace (Tiendas, Entrenadores, Productos, Servicios, Categorías, Marcas, Clientes, Suscriptores), Ventas (Punto de venta, Cupones, Páginas, Banners), and Ajustes (Zonas de reparto). The main content area features two calendar widgets. The top calendar is a standard monthly view for October 2021, with the 13th highlighted. Below it is a green button labeled 'Generar reporte'. The second widget is titled 'Reporte de ingresos' and includes a dropdown menu set to 'Todos', a date selector for 'Oct 13, 2021', and another calendar view for October 2021 with the 13th highlighted. A second green button labeled 'Generar reporte' is located below this calendar.

## BURNDOWN DEL SPRINT N° 7

Figura\_ 229: Burndown del Sprint 7



### Burndown Sprint 7

En la figura vista, se muestra el proceso de desarrollo de cada sprint la cual está representada a través de un gráfico, como podemos observar tenemos dos líneas la cual la línea roja representa al tiempo ideal que se define para el desarrollo del sprint, de igual manera la línea azul representa el tiempo real que tomo el desarrollo del sprint. Ya teniendo bien definido las representaciones tenemos que concluir que si la línea azul se encuentra por debajo de la roja significa que se presentó un adelanto, de lo contrario si la línea azul está por encima es decir que hubo un retraso.

## ACTA DE REUNION DEL SPRINT N° 7

Siendo las 3 pm del día 20 de agosto del 2021, se reúne en la empresa AC Brands & Consulting E.I.R.L.


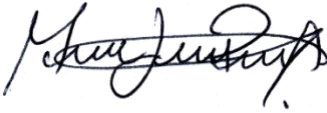


Presentes:

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

El encargado de la Empresa AC Brands & Consulting E.I.R.L., Isaac Abensur Vargas concluye la reunión con los Sres. Marzio Perez y Nicolás Segovia aclarando los últimos puntos sobre el sprint.

Para determinar las formas de trabajo y los requerimientos respectivos para esta interacción, también la fecha de entrega y firmar el acta para dar por terminado la reunión.

Los participantes manifestaron su aprobación respecto a todos los puntos definidos en la reunión, que fueron de ayuda a la planificación del Sprint, manifestando su compromiso para la entrega de los resultados en las fechas pactadas.

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza

## ACTA DE ENTREGA DEL SPRINT N° 7


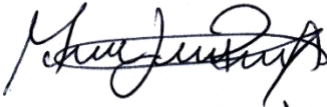


Siendo la 1 pm del día 31 de agosto del 2021 se reúne en la empresa AC Brands & Consulting E.I.R.L.

ROL	NOMBRE
Scrum Master	Isaac Abensur Vargas
Team Member	Marzio Perez Arbulú Nicolás Segovia Manrique
Product Owner	Juan Carlos Galagarza

Los Sres. Marzio Perez y Nicolás Segovia, especifican cada uno de los requerimientos desarrollados, también presentan las interfaces elaboradas y brindadas por el producto Owner.

Luego de ser verificadas las presentaciones realizadas por los Srs. Marzio Perez y Nicolás Segovia, se decidió de manera unánime, dar por aprobado el término del Sprint respectivo, del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Los presentes manifestarán su aprobación al informe presentado por los Sres. Marzio Perez y Nicolás Segovia del Sprint N° 7 terminado del proyecto “Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

			
Isaac Abensur Vargas	Marzio Perez Arbulú	Nicolás Segovia Manrique	Juan Carlos Galagarza



## RESUMEN DE LA REUNIÓN RETROSPECTIVA DE SPRINT N° 7

Información de la empresa y proyecto:

<b>Empresa / Organización</b>	AC Brands & Consulting E.I.R.L.
<b>Proyecto</b>	“Marketplace para el proceso de ventas en la empresa AC Brands & Consulting E.I.R.L.”

Información de la reunión:

<b>Lugar</b>	AC Brands & Consulting E.I.R.L.
<b>Fecha</b>	31/08/2021
<b>Número de iteración / Sprint</b>	Sprint 7
<b>Personas Convocadas a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza
<b>Persona que asistieron a la reunión</b>	Isaac Abensur Vargas Marzio Perez Arbulú Nicolás Segovia Manrique Juan Carlos Galagarza

Formulario de reunión retrospectiva

<b>¿Qué salió bien en la Iteración? (Aciertos)</b>	<b>¿Qué no salió bien en la Iteración? (Errores)</b>
Se logró desarrollar el historial de pedidos, la vista de los productos. Se desarrolló todos los procesos de compra Se desarrollaron los indicadores de ventas y calidad de pedidos generados.	Se presentó problemas al momento de desarrollar los procesos de compra, sin embargo, se solucionó.



**UNIVERSIDAD CÉSAR VALLEJO**

**FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

### **Declaratoria de Autenticidad del Asesor**

Yo, PEREZ FARFAN IVAN MARTIN, docente de la FACULTAD DE INGENIERÍA Y ARQUITECTURA de la escuela profesional de INGENIERÍA DE SISTEMAS de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA NORTE, asesor de Tesis titulada: "MARKETPLACE PARA EL PROCESO DE VENTAS EN LA EMPRESA AC BRANDS & CONSULTING E.I.R.L.", cuyo autor es PEREZ ARBULU MARZIO JOSUE, constato que la investigación cumple con el índice de similitud establecido, y verificable en el reporte de originalidad del programa Turnitin, el cual ha sido realizado sin filtros, ni exclusiones.

He revisado dicho reporte y concluyo que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la Tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

En tal sentido, asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

LIMA, 14 de Diciembre del 2021

<b>Apellidos y Nombres del Asesor:</b>	<b>Firma</b>
PEREZ FARFAN IVAN MARTIN <b>DNI:</b> 08647541 <b>ORCID</b> 0000-0001-5833-9400	Firmado digitalmente por: IPEREZFA el 22-12-2021 23:53:05

Código documento Trilce: TRI - 0223890