



UNIVERSIDAD CÉSAR VALLEJO

**FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS**

Arquitectura de microservicios para el proceso de ventas de la
óptica Multi-len's, 2022.

TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:

Ingeniero de Sistemas

AUTOR:

Carranza Felix, Christian Bernardo (orcid.org/0000-0002-0599-1347)

ASESOR:

Dr. Necochea Chamorro, Jorge Isaac (orcid.org/0000-0002-3290-8975)

LÍNEA DE INVESTIGACIÓN:

Sistemas de Información y comunicaciones

LÍNEA DE RESPONSABILIDAD SOCIAL UNIVERSITARIA:

Desarrollo económico, empleo y emprendimiento

LIMA - PERÚ

2022

Dedicatoria

A mis padres y hermanos, por su amor incondicional, por su apoyo constante y por ser mi mayor inspiración en la vida. Cada logro que he alcanzado es gracias a su dedicación y esfuerzo en brindarme siempre lo mejor. Gracias por enseñarme el valor de la perseverancia, la honestidad y el trabajo duro. Les dedico este trabajo con todo mi corazón.

Agradecimiento

A mis profesores, por sus enseñanzas, consejos y orientaciones en el desarrollo de esta tesis.

A mis amigos y compañeros, por compartir conmigo momentos inolvidables y por ser parte fundamental en mi formación como persona y profesional. Finalmente, quiero expresar mi gratitud a todas las personas que, de alguna manera, contribuyeron a la realización de este trabajo. Su apoyo y colaboración fueron fundamentales para lograr los objetivos propuestos.

Índice de contenidos

Dedicatoria	ii
Agradecimiento	iii
Índice de contenidos	iv
Índice de tablas	v
Índice de figuras	vi
Resumen	viii
Abstract	ix
I. INTRODUCCIÓN	1
II. MARCO TEÓRICO	5
III. METODOLOGÍA	20
3.1. Tipo y diseño de investigación	20
3.2. Variables y operacionalización	21
3.3. Población, muestra y muestreo	23
3.4. Técnicas e instrumentos de recolección de datos	23
3.5. Procedimientos	26
3.6. Método de análisis de datos	27
3.7. Aspectos éticos	28
IV. RESULTADOS	29
V. DISCUSIÓN	39
VI. CONCLUSIONES	41
VII. RECOMENDACIONES	43
REFERENCIAS	45
ANEXOS	54

Índice de tablas

Tabla 1 Matriz de Operacionalización de Variables	22
Tabla 2 Determinación para la población	23
Tabla 3 Instrumentos de recolección de datos	24
Tabla 4 Confiabilidad para la Tasa de adquisición	25
Tabla 5 Confiabilidad para la Tasa de conversión	25
Tabla 6 Estadísticos descriptivos del indicador tasa de adquisición antes y después de implementar el sistema web basado en la arquitectura de microservicios	30
Tabla 7 Estadísticos descriptivos del indicador tasa de conversión antes y después de implementar el sistema web basado en la arquitectura de microservicios	30
Tabla 8 Prueba de normalidad tasa de adquisición	32
Tabla 9 Prueba de Normalidad Tasa de Conversión	34
Tabla 10 Product backlog	63
Tabla 11 Historia de Usuario 1	64
Tabla 12 Historia de Usuario 2	65
Tabla 13 Historia de Usuario 3	65
Tabla 14 Historia de Usuario 4	66
Tabla 15 Historia de Usuario 5	66
Tabla 16 Historia de Usuario 6	67
Tabla 17 Historia de Usuario 7	67
Tabla 18 Historia de Usuario 8	68
Tabla 19 Historia de Usuario 9	68
Tabla 20 Historia de Usuario 10	69
Tabla 21 Historia de Usuario 11	69
Tabla 22 Historia de Usuario 12	70
Tabla 23 Historia de Usuario 13	70
Tabla 24 Historia de Usuario 14	71
Tabla 25 Historia de Usuario 15	71
Tabla 26 Sprint backlog	72
Tabla 27 Actividades	73
Tabla 28 Sprint 0	77
Tabla 29 Acta de reunión	78
Tabla 30 Desarrollo de la reunión	79
Tabla 31 Conclusiones	79
Tabla 32 Revisión del sprint	87
Tabla 33 Sprint 1	87
Tabla 34 Acta de reunión sprint 1	88
Tabla 35 Revisión del sprint	101
Tabla 36 Sprint 2	102
Tabla 37 Revisión del sprint	110

Índice de figuras

Figura 1 Arquitectura monolítica y de microservicios	10
Figura 2 Integración mediante API Gateway	12
Figura 3 Etapas del proceso de venta	15
Figura 4 Scrum: principios	18
Figura 5 Roles de scrum y su relación.	19
Figura 6 Diseño Pre - Experimental	20
Figura 7 Prueba de normalidad de la tasa de adquisición antes de la implementación del sistema	33
Figura 8 Prueba de normalidad de la tasa de adquisición después de la implementación del sistema	33
Figura 9 Prueba de normalidad de la tasa de conversión antes de la implementación del sistema	34
Figura 10 Prueba de normalidad de la tasa de conversión después de la implementación del sistema	35
Figura 11 Ficha de registro tasa de adquisición - Pretest	57
Figura 12 Ficha de registro tasa de adquisición - PostTest	58
Figura 13 Ficha de registro tasa de conversión – Pretest	59
Figura 14 Ficha de registro tasa de conversión – PostTest	60
Figura 15 Autorización para la realización y difusión de resultados de la investigación	61
Figura 16 Consentimiento informado	62
Figura 17 Patrón CQRS en el proyecto	80
Figura 19 Patrón SAGA en el proyecto	80
Figura 20 Diseño de la arquitectura de microservicios del proyecto	82
Figura 21 Diagrama de base de datos del microservicio Usuarios.	83
Figura 22 Diagrama de base de datos del microservicio carritos.	84
Figura 23 Diagrama de base de datos del microservicio productos	85
Figura 24 Diagrama de base de datos del microservicio ventas	86
Figura 25 Desarrollo UI de registro de usuarios	89
Figura 26 Controller API registrar usuario	90
Figura 27 Service API registrar usuario	90
Figura 28 Prueba del endpoint registrar usuario	91
Figura 29 Prueba funcional de registrar usuario	92
Figura 30 Mensaje de confirmación de registro usuario	92
Figura 31 login cliente	93
Figura 32 UI nueva categoría	94
Figura 33 Controller API registrar categoría	94
Figura 34 Service API registrar categoría	95
Figura 35 Prueba del endpoint registrar categoría	95
Figura 36 Prueba funcional de registro de categoría	96
Figura 37 Mensaje de confirmación de registro de categorías	97
Figura 38 Listado de categorías	97
Figura 39 UI subcategoría	98
Figura 40 Controller API registrar subcategoría	98
Figura 41 Service API registrar subcategoría	99

Figura 42 Prueba del endpoint registrar subcategoría	99
Figura 43 Prueba funcional registro subcategoría	100
Figura 44 mensaje de confirmación de registro de subcategoría	100
Figura 45 listado de subcategorías	101
Figura 46 UI Visualizar categoría de productos	103
Figura 47 Controller API listado de categorías	103
Figura 48 Repository API listado de categorías	104
Figura 49 Service API listado de categorías	104
Figura 50 Envío de parámetros para el endpoint Visualizar categoría de productos	105
Figura 51 Prueba del endpoint Visualizar categoría de productos	105
Figura 52 Prueba funcional listado de categorías - administrador	106
Figura 53 Prueba funcional listado de categorías - cliente	106
Figura 54 Prueba funcional listado de producto	107
Figura 55 Controller API listado de productos	108
Figura 56 Service API listado de productos	108
Figura 57 Envío de parámetros para el endpoint Visualizar productos	109
Figura 58 Prueba del endpoint Visualizar productos	109
Figura 59 Prueba funcional listado de productos	110

Resumen

El presente estudio titulado “Arquitectura de microservicios para el proceso de ventas de la óptica Multi-len’s, 2022” propuso una solución para el problema encontrado en la organización. El cual tuvo como objetivo principal determinar la influencia de una arquitectura de microservicios en el proceso de ventas de la óptica Multi-len’s. La investigación fue de tipo de estudio aplicado, con diseño preexperimental y con un enfoque cuantitativo, la técnica de recolección de datos fue el fichaje y la ficha de registro como instrumento. Asimismo, se usó la metodología RUP y SCRUM para el desarrollo del proyecto. Por otra parte, se empearon servicios en la nube de AWS, Kubernetes como orquestador de contenedores, java como lenguaje de programación en la versión 11 y su framework Spring, todo esto como parte del backend del sistema y para el frontend se utilizó el framework angular, y MYSQL, MongoDB y Redis como base de datos.

Por otra parte, para el indicador de la tasa de adquisición y la tasa de conversión se tuvo una población de 14 registros de clientes en un periodo de 2 semanas. De la misma forma se llevó a cabo la prueba de normalidad empleando el método de Shapiro-Wilk, ya que la población es menor a 50; y se empleó el t-Student para aceptar o rechazar las hipótesis, ya que los datos encontrados tienen una distribución normal. Los resultados obtenidos luego de realizar el pretest y el PostTest para el indicador de tasa de adquisición demostraron un incremento de 67.72% con respecto a la situación anterior a la implementación de dicho sistema ya que en el pretest se obtuvo 1.89 y en el PostTest 3.17. Por otro lado, para la tasa de adquisición se obtuvo un incremento de 45.16% pues en el pretest se obtuvo 0.62 y en el PostTest 0.90.

Basándose en los resultados obtenidos de las pruebas se concluyó que una arquitectura de microservicios mejora el proceso de ventas de la óptica Multi-len’s

Palabras clave: MICROSERVICIOS, SISTEMA DE VENTAS, SCRUM, SERVICIOS EN LA NUBE.

Abstract

The present study titled 'Microservices Architecture for the Sales Process of Multi-len's Optics, 2022' proposed a solution for the problem encountered in the organization. The main objective was to determine the influence of a microservices architecture on the sales process of Multi-len's Optics. The research was of an applied study type, with pre-experimental design and a quantitative approach, using time card and registration form as data collection techniques. Furthermore, RUP and SCRUM methodologies were used for project development. On the other hand, AWS cloud services, Kubernetes as container orchestrator, Java programming language in version 11 and its Spring framework were used as part of the system's backend, while Angular framework was used for the frontend, and MYSQL, MongoDB, and Redis were used as databases.

For the acquisition rate and conversion rate indicator, a population of 14 customer records was taken over a period of 2 weeks. The normality test was performed using the Shapiro-Wilk method, as the population is less than 50, and the t-Student test was used to accept or reject hypotheses, as the data found has a normal distribution. The results obtained after conducting the Pretest and PostTest for the acquisition rate indicator showed an increase of 67.72% compared to the situation prior to the implementation of the system, as the Pretest obtained 1.89 and the PostTest obtained 3.17. On the other hand, for the acquisition rate, there was an increase of 45.16% as the Pretest obtained 0.62 and the PostTest obtained 0.90.

Based on the test results, it was concluded that a microservices architecture improves the sales process of Multi-len's Optics.

Keywords: MICROSERVICES, SALES SYSTEM, SCRUM, CLOUD SERVICES.

I. INTRODUCCIÓN

En los últimos años las tecnologías de información y comunicación han tenido un impacto significativo en las organizaciones, creando diferentes ventajas en todas las áreas de trabajo, quitando las limitaciones que se tenía en negocios tradicionales. Actualmente las empresas necesitan adaptarse a un mercado exigente y globalizado para seguir siendo competitivo, por lo que el futuro de cada industria depende de la innovación y la tecnología (Bentaher y Rajaa, 2022). ya que el mercado actual tiene una rivalidad feroz y demandas dinámicas. Por lo que las empresas que no puedan estar al día con los desarrollos tecnológicos están condenadas al descenso (Firmansyah et al. 2019). Siendo así la tecnología un elemento indispensable e importante para el desarrollo de las actividades tanto en empresas pequeñas como grandes.

Por otro lado, las ventas son las bases para la economía en general, razón por la cual las organizaciones nacionales e internacionales buscan soluciones tecnológicas que puedan ayudarlos en mejorar dicho proceso, de tal forma que cumplan con sus metas de manera eficiente. Los países extranjeros vienen adoptando los distintos tipos de herramientas tecnológicas como los sistemas de escritorio, web y móviles, las cuales favorecen en la venta de servicios y productos. Dado que los clientes quieren realizar sus compras a gusto, además con la nueva forma de hacer negocio hay una mayor competencia (Yulianto y Fauzi, 2020).

Asimismo, Soegoto D. y Cica C. (2018). Mencionan que las ventas son el proceso principal de las organizaciones al momento de generar ingresos, tanto en empresas grandes como pequeñas. Además, estos son los objetivos del mercadeo, ya que es allí donde se definen los precios, las negociaciones y las formas de pago, para llegar a un punto de satisfacción. Una de las formas de ventas son las que se dan por internet las cuales pueden aumentar la cantidad de ganancias.

En un contexto nacional, según Zafra (2020). Las empresas peruanas cada día tienen un mayor crecimiento, especialmente las que forman parte del sector de servicios y ventas. Lo cual hace indispensable el contar con control de las ventas para el conocimiento del beneficio obtenido. Al mismo tiempo tener el

conocimiento de la cantidad de ventas realizadas, conservando la calidad en el servicio.

Por otra parte, para realizar los sistemas de información existen varios patrones de arquitectura de software, uno de los más comunes que antiguamente se usaban eran las arquitecturas monolíticas, la cual hoy en día ya no cubre las necesidades de las organizaciones. Dado que este no permite la escalabilidad de los sistemas, además de que son más complejos y costosos mantenerlos (Bucchiarone et al. 2018). A raíz de estos problemas nacen otros tipos de arquitectura como los microservicios, la cual se fundamenta en la capacidad de modularización. Es decir que descompone el sistema en pequeños servicios, la cual lo hace mantenible y escalable, contrastando así con la arquitectura monolítica tradicional la cual encapsula todas las lógicas comerciales en una sola pieza y, por lo tanto, hace que el sistema solo pueda implementarse como una sola unidad (Liu, Jurdana, Lopac, Wakabayashi. 2022).

La presente investigación se llevó a cabo en la óptica Multi-len's ubicada en la ciudad de Lima, lo cual ofrece servicios optométricos, además del cuidado de la imagen integral y la venta de productos para la salud visual, de igual forma la reparación y mantenimiento de monturas. En los últimos años la empresa ha tenido un gran crecimiento, contando actualmente con varias sedes en los distintos distritos de Lima, lo que le ha permitido llegar a más clientes, haciendo que la demanda de sus servicios aumente, por lo que atender a las peticiones de forma rápida es crucial tanto para fidelizar y atraer.

Sin embargo, la empresa no cuenta con un sistema que le pueda ayudar en sus procesos, y esto ha hecho que muchas veces tanto los pedidos, las citas y las ventas se tarden o se cancelen, ya que además a la falta de un sistema actualmente realizan sus registros de manera tradicional a través de herramientas no automatizadas tales como los registros en cuadernos, en formularios y entre otros. Las cuales son usadas posteriormente para gestionar el historial, las ganancias, las ventas, pagos, etc. En consecuencia, esta forma de administrar dichos datos genera un problema, ya que son susceptibles a pérdidas y además no se puede administrar correctamente tales procesos, impidiendo que se le pueda dar un seguimiento, de igual forma cuando ocurren

problemas con las ventas, la búsqueda de la transacción lleva mucho tiempo poder encontrarlos y esto a la vez aumenta la posibilidad de que un cliente ya no quiera los servicios de la empresa.

Asimismo, entre los problemas esta la falta de centralización de la información que maneja cada una de las sedes de la óptica. Dado que todos ellos se administran por separado y con una información heterogénea. Por otra parte, el inventariado se lleva a cabo a mediante archivos de Excel, en donde se registra todos los datos de los productos, el stock y la rotación entre las sedes, y esto hace que se pierda tiempo tanto en llenarlo como obtener la información cuando se requiere, resultando así un problema poder mantener un control correcto de estos. Asimismo, la actual forma de administrar la información trae problemas con el cliente, debido a la velocidad de atención, ya que muchas veces la consulta de alguna información puede tomar tiempos excesivos y aún más en ciertas horas del día cuando hay una mayor demanda de servicios.

Además, otro de los problemas principales en la cual se centra la investigación es en la manera en cómo lleva a cabo la promoción y venta por internet la empresa, ya que actualmente está usando las redes sociales como Facebook para realizar dichos procesos, la cual limita el alcance de las ventas sólo a las personas que utilizan esta red social, lo que además impide que la empresa pueda tener un control sobre los productos vendidos o las preferencias de los clientes. De igual forma para dar a conocer los servicios y los productos usa las redes sociales y folletos, la cual no es tan eficiente si se requiere llegar a los clientes

Luego de realizar el análisis actual de la óptica Multi-len's, se presentó la siguiente problemática general: ¿Cómo influye una arquitectura de micros servicios en el proceso de ventas de la óptica Multi-len's? Y como primera problemática específica: ¿Cómo influye una arquitectura de micros servicios en la tasa de adquisición para el proceso de ventas de la óptica Multi-len's? Y como última pregunta específica: ¿Cómo influye una arquitectura de micros servicios en la tasa de conversión para el proceso de venta de la óptica Multi-len's?

La justificación del estudio se da mediante la relevancia social, dado que ayudará en la perspectiva que tienen los clientes de la empresa, y esto atraerá más comensales con una visión complacido por el sistema de ventas, que pasará a la automatización, lo cual generará una mayor interactividad y aceptación de los que realizan una compra en la óptica.

La justificación de valor teórico es dado que se va a hacer uso de las tecnologías recientes, la cual le traerá distintas ventajas a la empresa que lo utilice. Por otra parte, también está el respaldo con el que se basa la investigación. Finalmente, el estudio brindará conocimiento sobre las nuevas maneras de desarrollar un sistema de información, mostrando tanto el diseño, el desarrollo e implementación de la misma.

Ante todo, lo investigado se presentó como objetivo general lo siguiente: Determinar la influencia de una arquitectura de microservicios en el proceso de ventas de la óptica Multi-len's. Y como primer objetivo específico: Determinar la influencia de una arquitectura de microservicios en la tasa de adquisición del proceso de ventas de la óptica Multi-len's. Y como segundo objetivo específico: Determinar la influencia de una arquitectura de microservicios en la tasa de conversión para el proceso de venta de la óptica Multi-len's.

Los objetivos mencionados permitieron presentar las siguientes hipótesis: Una arquitectura de microservicios mejora el proceso de ventas de la óptica Multi-len's: Y las otras hipótesis específicas, la primera es: Una arquitectura de microservicios mejora la tasa de adquisición en el proceso de ventas de la óptica Multi-len's. La segunda es: Una arquitectura de microservicios aumenta la tasa de conversión en la gestión de ventas de la óptica Multi-len's.

II. MARCO TEÓRICO

Para respaldar el presente trabajo se buscó varios antecedentes que se relacionen con la pregunta de investigación y el tema en general. Dichos trabajos abarcan tanto a nivel nacional como internacional, los cuales se detallan a continuación.

Mora M., Noraimar Y. (2019), en su tesis de licenciatura “Arquitectura basada en microservicios para sistemas de E-commerce de la empresa LCC Opentech, C.A.”, la cual tuvo como objetivo desarrollar un e-commerce escalable y de alta disponibilidad que se base en la arquitectura de microservicios para dicha empresa. Esta investigación es proyectiva. Además, la investigación se realizó siguiendo un enfoque basado en un modelo de desarrollo incremental. Al final, como resultado de someter la arquitectura de software a muchas pruebas, la arquitectura desarrollada satisface las necesidades de la organización, por lo que se adapta perfectamente al tipo de sistema que la empresa desea construir.

Cahyo y Candiwan (2020) en su artículo titulado “Analysis and Design of Sales Information System on Web-Based E-Commerce in Yoga Farm Catfish Breeding Business Using UML”. Mencionan los problemas que la empresa tenía cuando usaban un sistema manual para la venta y dar a conocer sus productos, por lo cual la investigación tenía el objetivo de diseñar, desarrollar e implementar un software para el proceso de ventas que además le ayude a los clientes la obtención de la información de productos en tiempo real, asimismo que sean conocidos ampliamente de tal forma que pueda aumentar la clientela, y para este sistema se usó el lenguaje de modelado unificado para el diseño. El método de la investigación fue cualitativo. Como conclusión luego de implementar el sistema web, se observó que los productos fueran conocidos por una cantidad mayor de personas, haciendo que los clientes aumenten y esto a la vez ayudó a que las ganancias incrementaran. De igual forma con la adopción del sistema de ventas se pudo reducir el error humano al momento de realizar informes de las ventas e inventario. Por el lado del cliente, este sistema le ayudó a la obtención de la información de productos rápidamente y las transacciones eran más fáciles.

Ezeonwumelu, Eunice y Ezenugu (2017). Realizó una investigación llamada “Development of Sales and Inventory Workflow Management

Information System Web Portal for Petrospan Integrated Services, Eket, Akwa Ibom State, Nigeria Adanna”. En la cual desarrolló un aplicativo Web para la administración de ventas e inventariado de la organización, usando la metodología Rapid Application Development (RAP) para la construcción de dicho sistema. Además, se empleó una arquitectura de tres niveles que se basa en la configuración del servidor WAMP. Por otra parte, el sistema de información desarrollado contaba con cinco módulos tanto para la administración, inventariado, clientes, oficina de ventas y la información de la empresa. Finalmente, el sistema implementado permitió realizar los registros de las transacciones para que posteriormente puedan darle seguimiento y auditoría de las actividades diarias de inventario y ventas.

Djatkiko et al. (2017) en su trabajo de investigación titulado “Analyze and Record a Series of Corporate Sales Transactions On Web Based Accounting Online System”. Nos habla acerca de la importancia del proceso de venta en una organización, en la cual también menciona que todo el flujo de datos que se mueve en dicho proceso debe registrarse y analizarse en tiempo real, en cualquier momento y lugar, de tal forma que se pueda mitigar los delitos financieros es decir el robo de las ganancias que a la larga puede perjudicar a la organización. Dado el problema mencionado es que, en la investigación, el autor propuso un sistema web para el proceso de ventas que pueda solventar dichos inconvenientes. Y como resultado de dicha implementación se vio una mejora en el registro y análisis de ventas gracias a que dicha aplicación agiliza el proceso, además de que cada incidente de ventas fue registrado, reduciendo así las pérdidas para la empresa.

Zafra (2020) en su tesis “E-commerce basado en microservicios para el proceso de ventas en la empresa COMATPERU S.A.C.”, en la que desarrolló como objetivo principal establecer el cómo influye un ecommerce desarrollado con una arquitectura de microservicios en la empresa COMATPERU S.A.C. El tipo de investigación fue de tipo aplicada, de diseño preexperimental, de enfoque cuantitativo, con una población de 100 ventas de las cuales se tomó una muestra de 80 ventas. Además, se usó la metodología SCRUM para desarrollar el software propuesto. Los resultados del estudio mostraron que el porcentaje de ventas tuvo un crecimiento de un 16.45% a 21.90% gracias al Ecommerce

implementado, de igual forma la tasa de conversión tuvo un aumento de un 43.95% a 67.25%. Concluyendo así que un ecommerce desarrollado en base a la arquitectura de microservicios mejora la administración de ventas en la empresa COMATPERU S.A.C. De esta investigación se tomará en consideración las dimensiones de la variable dependiente proceso de venta, la cual es importante para el proyecto.

Delgado y Cieza (2020) en el artículo que publicaron “Design of a Web System for Sales Processes in a Microenterprise in Peru”. Hicieron una propuesta del diseño de un sistema web de ventas que le permita a la empresa ubicada en Puente Piedra administrar la información de ventas diarias, así como también el control de inventarios e información de sus clientes. Esto debido a que la organización venía presentando problemas en el control óptimo sobre sus ventas. Por otra parte, para llevar a cabo el proyecto se usó la metodología SCRUM la cual le permitió agilizar el desarrollo. Finalmente, el diseño del sistema web se completó, de esta manera la implementación de ello mejorará el proceso de ventas, eliminando los errores humanos en los cálculos para la elaboración de tickets o el control de inventarios.

Katherin (2020), en su tesis de titulación “Sistema de Gestión de Contenidos para el Proceso de Ventas en la MYPE D’Tudo Digital S.A.C.” Desarrolló como objetivo principal el establecer los efectos de este sistema al implementarlo en el proceso mencionado de la organización. El tipo de investigación fue la experimental aplicada y preexperimental como diseño. Tomando como población 20 fichas de registro las cuales se llevaron a cabo en un mes. Como resultado luego de la implementación de la aplicación web se obtuvo para el indicador eficiencia un resultado del 76% de un 50%, teniendo así un incremento del 26%. Por otra parte, para el indicador de ventas, se obtuvo un 57% antes de la implementación y luego de la implementación se observó un incremento 31% es decir un resultado del 80%. Finalmente se logró determinar que el sistema propuesto tuvo mejoras en el proceso de ventas, demostrando así el logro de los objetivos. De esta tesis se tomará definiciones de la variable venta.

Ruelas (2017), en su investigación titulada “Modelo de composición de microservicios para la implementación de una aplicación web de comercio electrónico utilizando kubernetes”, desarrolló como objetivo principal el plantear un modelo de composición de microservicios para el despliegue de una E-commerce usando Kubernetes. Su método de investigación fue de tipo experimental con una población de todas las peticiones que recibía la aplicación web de la cual tomó como muestra las 20 peticiones por segundo. En esta investigación se usó la metodología XP para el desarrollo de la misma. Como resultado se obtuvo que la aplicación tuvo mejoras significativas en los indicadores de disponibilidad, tiempo de respuesta y rendimiento, aumentado hasta un 104%. Por lo cual se concluyó que el modelo presentado sostiene un significativo funcionamiento.

Robinson y Romero (2017), en su investigación titulada “Sistema web para el proceso de ventas en la empresa Rysoft” la cual tenía como objetivo general precisar la influencia de un sistema web en el proceso de ventas. Su tipo de investigación es experimental - aplicada, en donde se usó la metodología RUP para el diseño y desarrollo del sistema propuesto. La población estaba compuesta por 8 registros de ventas, que a la vez también fue su muestra. Finalmente se obtuvo como resultado en el pretest 0.50 en el índice de calidad de ventas calificando como muy alto a lo esperado y 50.05 en el índice de fiabilidad calificando, así como muy bajo a lo esperado. Luego de implementar el sistema propuesto, se realizó un PostTest en la cual se obtuvo como resultado un 0.16 en la calidad de ventas y un 80.71 en la fiabilidad de entregas. Por lo que se concluyó que la aplicación web aumenta el índice de fiabilidad, pero reduce la calidad de ventas.

A fin de tener un mejor respaldo del presente trabajo se han tomado teorías que se relacionen con el tema. Para ponernos en contexto comenzaremos por definir la arquitectura de software.

Para esto, es importante mencionar que no existe un consenso para definir exactamente la arquitectura de software. Según Medel et al (2021). Es una descripción de los componentes y subsistemas de un software y las conexiones entre estos (p.6). Es decir, este establece la estructura de un sistema. Asimismo, la arquitectura de software es uno de los elementos más importantes de las aplicaciones, ya que define los requisitos operativos y técnicos, además se encarga de optimizar todos los atributos del sistema, tales como la escalabilidad, confiabilidad, manejabilidad y otros aspectos más. Siendo así de suma importancia la adecuada elección en la fase de desarrollo del software (Zabierowski, 2020).

Actualmente existen varios patrones de arquitectura de software, pero entre los más conocidos para el desarrollo de sistemas empresariales modernos están las arquitecturas monolíticas y de microservicios. La primera tiene un enfoque tradicional en la cual las aplicaciones tienen única base de código la cual incluye múltiples servicios que se comunican con otros sistemas consumidores o externos mediante varios interfaces como REST API, servicios web o páginas HTML (Al-Debagy y Martinek, 2018). Además, en este no se puede ejecutar de manera independiente ya que integra tanto el acceso de datos como la lógica del programa en un mismo desplegable. Lo cual hace que mantenerlo y actualizarlo sea más difícil, por lo que este tipo de arquitectura ya no es suficiente para las nuevas necesidades de las organizaciones. (Bucchiarone et al. 2018).

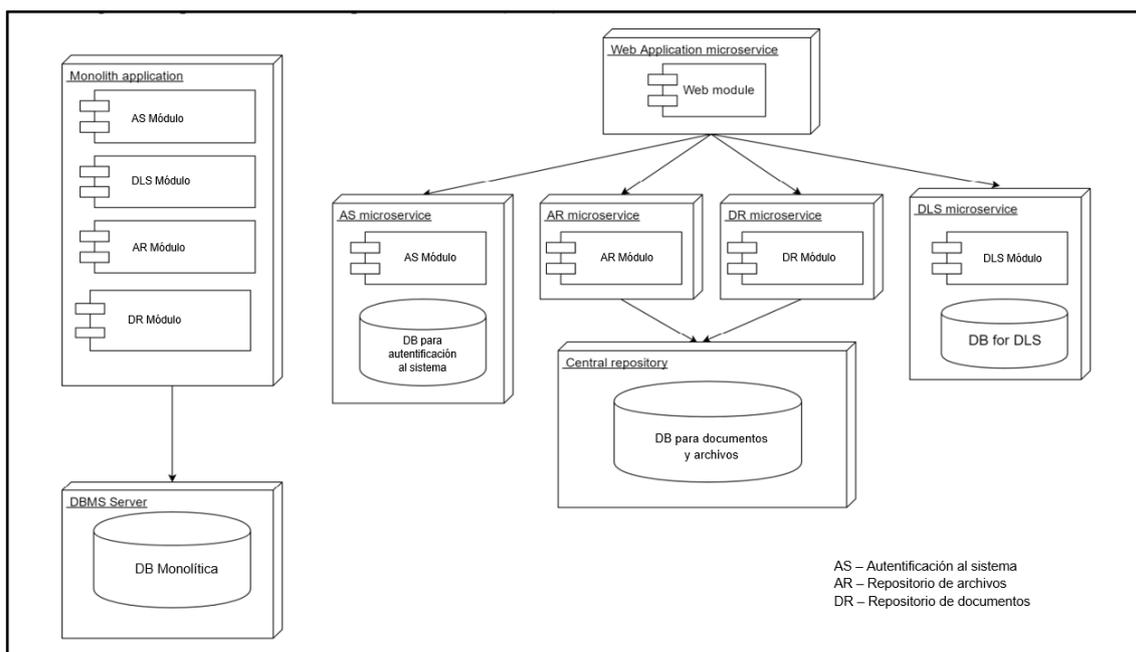
De igual forma, Kuryazov, Jabborov y Khujamuratov (2020). Nos dicen que tanto a los desarrolladores, arquitectos y diseñadores se les hace muy complicado el mantener bajo control el sistema y tener una visión detallada de las partes del sistema, lo cual además impide que no se puedan adaptar nuevos cambios de los usuarios o variar en las tecnologías que se usan.

Por otra parte, la arquitectura de microservicios (**variable independiente**). Son aplicaciones pequeñas que tienen una sola responsabilidad, la cual a diferencia de las monolíticas se pueden implementar, probar y escalar de manera independiente. Estos sistemas pueden comunicarse a través de APIs basados en servicios web RESTful o RPC. Esta arquitectura les permite a las empresas lograr un mejor tiempo de comercialización gracias a las entregas continuas y rápidas. Además, se puede hacer un desarrollo más ágil, puesto que, los microservicios se pueden desarrollar de manera autónoma. (Larrucea et al. 2018).

Por otro lado, Bucchiarone et al. Menciona que los microservicios nacen a partir de las arquitecturas orientadas a servicios (SOA) con el fin de dividir sistemas grandes en pequeñas partes, la cual permite tratar los problemas de forma independiente. (2018). Este elimina varios de los problemas que presentan las SOA, la cual está más enfocado al código. Además, SOA se fundamenta en el concepto de reutilización más rápida, es decir compartir todo lo que se pueda, en cambio los microservicios son todo lo contrario a esa idea.

En la figura 1 se muestra una representación de las arquitecturas monolíticas y de microservicios.

Figura 1 Arquitectura monolítica y de microservicios



Fuente: Berkunskyj et al. 2017.

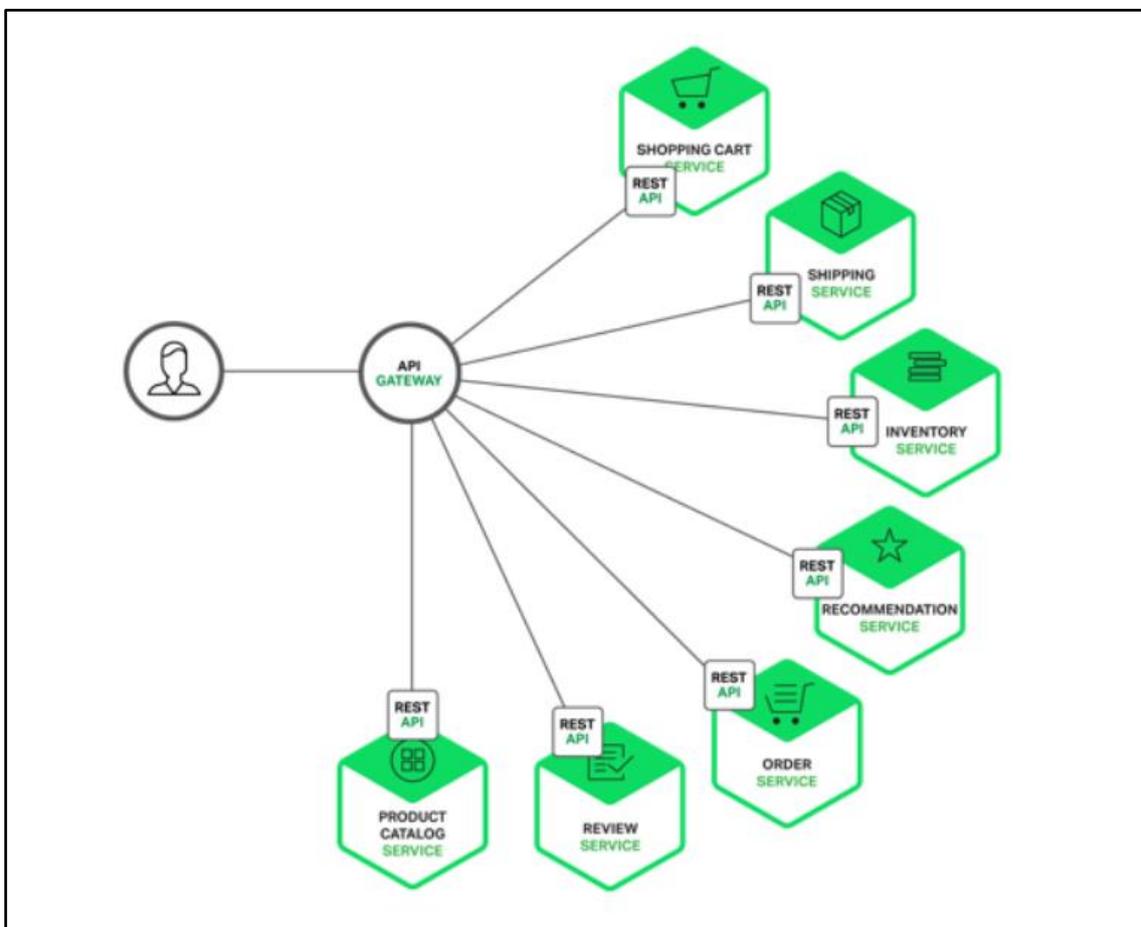
Entre las ventajas de implementar los microservicios según Raj y Ravichandra (2018), son: el acoplamiento suelto, debido a que según la definición los servicios tienen que ser livianos y pequeños, además que estos solo deben construirse para cumplir solo un requisito comercial. En la cual cada microservicio posee su propia base de datos. Reduciendo así el acoplamiento entre consumidores y servidores. Por otra parte, uno de los grandes beneficios que ofrece son la escalabilidad de los sistemas, ya que nos permite escalar solo los componentes que se requieran del código o los servicios, lo cual reduce los problemas de rendimiento. Asimismo, los servicios que fueron construidos con la arquitectura de microservicios son reutilizables. Por otro lado, ninguno de los servicios cuenta con estado, es decir no almacenan nada, debido a que utilizan protocolos de transporte como HTTP.

Igualmente, Indrasiri y Siriwardena (2018). Nos mencionan los beneficios de los microservicios tales como: El despliegue ágil y rápido, esto gracias a que los servicios desarrollados son autónomos, lo cual facilita su despliegue a diferencia a las arquitecturas convencionales como las monolíticas en donde pasar las funcionalidades a producción requiere de mucho tiempo, fundamentalmente por el tamaño de la aplicación y dependencias, y en los microservicios solo se requiere focalizar en la funcionalidad y la interfaz de un servicio y no en toda la aplicación. Además, estos son reemplazables, nuevamente gracias a su naturaleza autónoma, ya que este comparte información vía APIs permitiéndonos así reemplazarlo fácilmente por una mejor implementación. Por otro lado, como uno de los principales beneficios está la tolerancia a fallos, ya que, si uno de los servicios presenta problemas, los demás seguirán funcionando con normalidad, en cambio en las arquitecturas tradicionales ese fallo puede significar la caída de todo el sistema.

Por otra parte, Wolff (2016). Nos menciona acerca de los motivos por la cual implementar los microservicios. Entre las razones técnicas está el desarrollo sostenible, ya que en una arquitectura tradicional con el paso del tiempo puede ser complejo mantenerlos, en cambio con los microservicios pueden ser reemplazados cuando se requiera. Otra razón es que nos permite la entrega continua, es decir la aplicación puede estar en producción rápidamente gracias a su modularidad. Además, permite que la aplicación tenga una robustez ya que

este tiene mecanismos para prever las fallas. Y finalmente, una arquitectura de microservicios nos permite desarrollar con un stack diverso, lo cual posibilita al equipo trabajar cada microservicio en las tecnologías que mejor manejen. Y desde el punto de vista del negocio, pues está el poder trabajar en historias en paralelo. Es decir, los equipos pueden desarrollar distintos microservicios y si uno de ellos se retrasa o tiene dificultades para avanzar, este no afectaría a otros. Lo cual también reduce los riesgos del proyecto. La razón organizacional, es que gracias a que nos permite separar proyectos grandes en pequeños, cada uno independientes del otro, la coordinación también se descentraliza. Y esto nos permite tener estimaciones más exactas por servicio.

Figura 2 Integración mediante API Gateway



Fuente: Richardson y Smith (2016).

En la figura 2 se señala la integración de los microservicios mediante API Gateway. Según Song, Zhang y Haihong, es un servidor a través del cual los clientes de los microservicios pueden entrar a los servicios de cada uno de ellos

por un único punto de entrada. Este servicio encapsula la arquitectura interna de la aplicación para invocar adecuadamente los servicios backend. Dado que este tiene una sola entrada puede tener más responsabilidades como los servicios de autenticación, almacenamiento en caché, balanceo de carga, monitoreo, puntuar las peticiones por servicio. La cual también es responsable de exponer los servicios. (2018).

Por otro lado, el microservicio que se va a desarrollar será desplegado utilizando Docker como contenedor. Según Klinbua y Vatanawood (2017). Docker es una plataforma de contenedores de software que es empleado por desarrolladores, usuarios, empresas e investigadores para múltiples fines. Cada contenedor crea fragmentos del componente SaaS, la cual es ejecutada en paquetes aislados de los demás contenedores y en entorno seguro. Desde el punto de vista de un desarrollador, este les facilita en la creación de entornos, así como también la automatización para desplegar sus aplicaciones. Además, les permite compartir plantillas para entornos o contenedores iguales según la necesidad. Por otra parte, al emplear docker se pueden reducir los problemas de las dependencias.

El objetivo de los contenedores docker es empaquetar las aplicaciones en unidades estandarizadas que contengan los elementos importantes para su funcionamiento, incluidos las herramientas del sistema, tiempo de ejecución, bibliotecas y código.

Por otra parte, el lenguaje de programación empleado para el desarrollo de los microservicios del proyecto es java. Según Jabri (2021). Es un lenguaje de programación que está orientado a objetos de alto nivel y de propósito general. La cual nació con el objetivo principal de permitir a los usuarios escribir código una sola vez y poder ejecutar en todas partes.

Asimismo, se usará su framework Spring. Según Carnell y Huaylupo (2021). Es uno de los framework de desarrollo más populares para crear aplicaciones basadas en java. Spring se basa en el concepto de inversión de control, la cual permite administrar de forma eficiente proyectos grandes de java,

gracias a que externaliza las relaciones entre los objetos dentro del sistema por medio de anotaciones.

De igual forma para la puesta en producción del sistema se hará el despliegue de los microservicios en Amazon Web Service (AWS) el cual según Neela et al. (2021). Es una plataforma de computación en la nube que ofrece muchos servicios que son rentables, confiables, rápidos y fáciles de usar. Amazon Web Service es un proveedor de servicios en la nube proporcionado por Amazon. Además, es una combinación de tres servicios: infraestructura como servicio (IaaS), plataforma como servicio (PaaS) y software como servicio (SaaS). Además, es la plataforma en la nube más común, ampliamente adoptada y utilizada a nivel mundial que proporciona muchos servicios de extremo a extremo a nivel global y local, como Amazon Services desde centros de datos. Muchos clientes y usuarios, incluidas nuevas empresas, grandes empresas y otros gobiernos utilizan AWS para reducir costos, aumentar la velocidad informática.

Para el despliegue también será necesario emplear Kubernetes el cual es una plataforma de administración de contenedores de código abierto que está basado en Docker el cual implementa contenedores en múltiples nodos de host, además de planificar y escalar dichos contenedores en un clúster. Los pods son el recurso operativo principal para Kubernetes y consisten en uno o más recursos compartidos de red, almacenamiento y contenedores que contienen especificaciones de contenedores activos. El escalado automático horizontal de Kubernetes es un aumento o disminución en la cantidad de copias de grupo que se puede lograr mediante la función HPA (escalado automático de grupo horizontal). HPA recopila la cantidad de carga de la cápsula y la compara con el umbral establecido de la cápsula. Cuando la carga supere el umbral, HPA creará un nuevo Pod para garantizar la estabilidad del servicio. A medida que cae la carga, HPA libera el exceso de reserva para utilizar los recursos del sistema (Hu y Wang 2021).

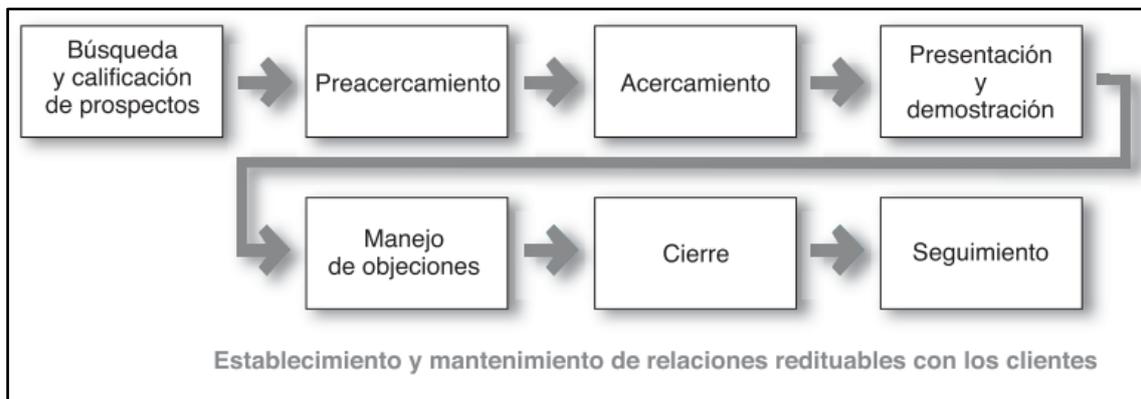
Por otro lado, el proceso de ventas (**variable dependiente**). Es definido por los siguientes autores:

Según Nilsson. Un proceso de ventas es una serie de pasos predecibles o fases requeridas para vender un servicio o producto. (2022). En la cual cada venta es realizada por un equipo o un solo vendedor, donde deben seguir ciertos pasos que son importantes tanto para la administración de ventas como para el monitoreo del canal de ventas (Parsalis, 2019).

Igualmente, Kotler y Armstrong (2012). Menciona que es el proceso determinado por una serie de pasos que deben cumplir los vendedores, debido a que tienen el fin de atraer más clientes y a la vez lograr que puedan concretar la compra.

En la siguiente figura 3 se muestran los pasos que forman el proceso de ventas.

Figura 3 Etapas del proceso de venta



Fuente: Kotler y Armstrong (2012).

Por otra parte, como nuestra **dimensión e indicadores** de la variable dependiente tenemos lo siguiente: La dimensión que se escogió para llevar a cabo la investigación es **la dimensión de prospección** del proceso de ventas. La cual es el proceso de reconocimiento de clientes potenciales, tanto empresas como personas que cumplan con la descripción de cliente ideal de cada negocio.

Y como primer indicador de dicha dimensión tenemos **la tasa de adquisición**. La cual, según Domínguez (2007). Nos permite saber la cantidad de clientes nuevos en un periodo establecido.

Fórmula:

$$TA = \frac{CN}{NC}$$

Donde:

CN: Número de clientes nuevos en el periodo.

NC: Número de clientes en el periodo

Y como segundo indicador de dicha dimensión tenemos **la tasa de conversión**. La cual, según Gabir, Karrar (2018). Se puede definir como el porcentaje de visitas a un sitio web de comercio electrónico que incluyen una transacción de compra. También se puede denominar como el porcentaje de visitantes del sitio web que se transforman en ventas potenciales al proporcionar sus datos de contacto al completar un formulario web. La tasa de conversión también se definió como el porcentaje del total de usuarios que compran un producto del total de visitas únicas en el sitio web. Es decir, permite saber la cantidad de clientes que se interesaron por el producto o servicio finalmente compraron. Domínguez (2007).

Fórmula:

$$TC = \frac{NCN}{PPC}$$

Donde:

NCN: Número de clientes nuevos

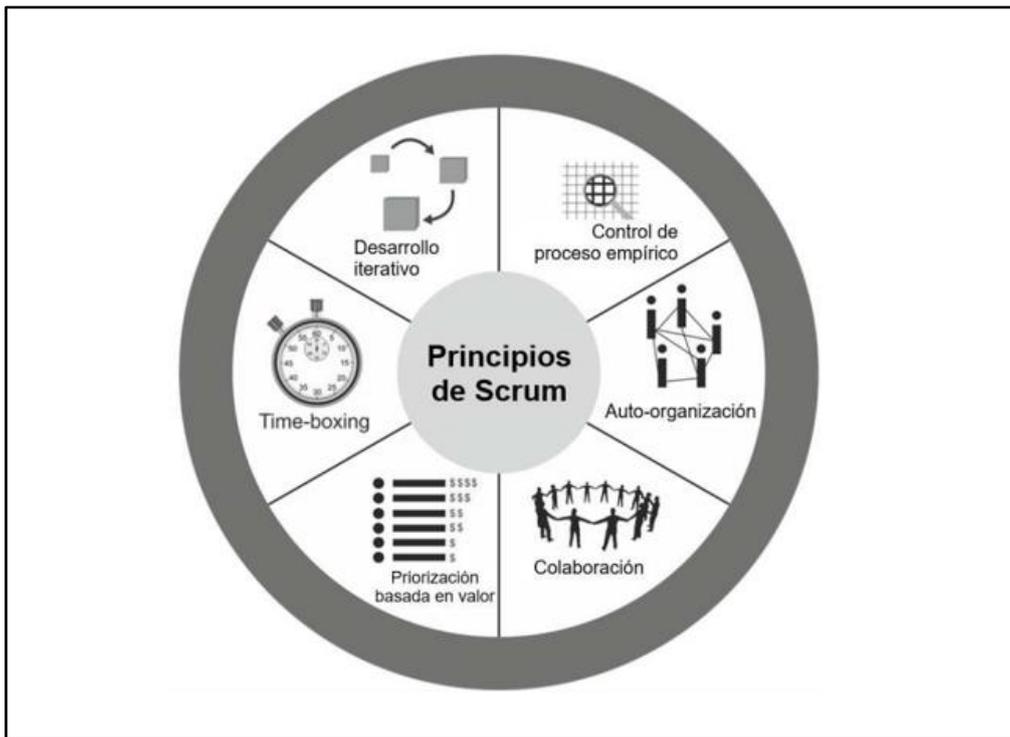
PPC: Peticiones prospectivas recibidas

Asimismo, dado que el desarrollo de un microservicio va siempre de la mano con las metodologías ágiles para llegar al éxito, se ha escogido una metodología para el desarrollo de ello. A pesar de que hay varios tipos de metodología tales como el Proceso Unificado Racional (RUP) y Programación extrema (XP). Para este proyecto se ha considerado usar la metodología Scrum luego de que se realizarán las comparaciones respectivas con las demás metodologías, ya que esta metodología es de las más usadas para realizar el análisis, desarrollo, implementación y documentación de software orientado a objetos.

Scrum es una metodología ágil, iterativa e incremental para la gestión del desarrollo de sistemas (Shafiee et al. 2020). Este puede ser empleado en casi cualquier proyecto, pero comúnmente es más utilizado en el desarrollo de software.

Entre sus principios ágiles están la colaboración activa entre los desarrolladores de la aplicación con los expertos, la satisfacción del cliente y como su principio clave está la volatilidad de los requisitos, es decir aceptar cambios en el proceso de desarrollo, ya que es consciente del hecho que los clientes pueden sugerir nuevos cambios o simplemente cambian la manera en cómo lo querían o necesitaban la aplicación inicialmente. Y estas modificaciones no previstas son difíciles de abordar con enfoques tradicionales, por ende, es una ventaja de esta metodología el poder adaptarse a los cambios. Además, tiene un enfoque empírico, dado que es consciente de que no puede definir por completo el problema, por lo cual se enfoca en atender los requerimientos nuevos y adaptarse a las tecnologías actuales, y en las nuevas condiciones del mercado actual. Además, scrum utiliza en tiempo real procesos de toma de decisiones en base a información y eventos reales, lo que demanda equipos con capacidad de autogestión, especializados, toma de decisión y comunicación (Sachdeva, 2016). En la siguiente figura 4 se muestran dichos principios mencionados, que son obligatorios para emplear el framework Scrum en todos los proyectos.

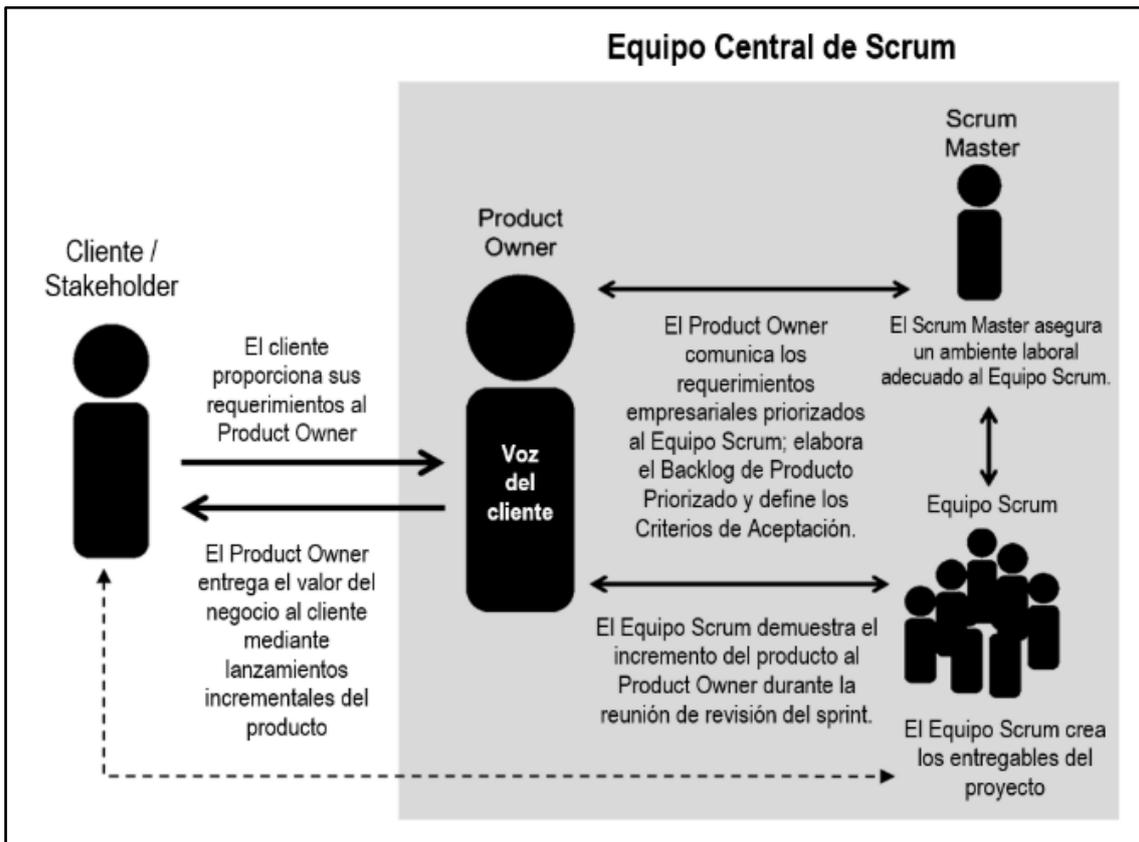
Figura 4 Scrum: principios



Fuente: Scrumstudy (2017).

Por otra parte, en comparación con las metodologías tradicionales para la gestión de proyectos, el scrum no requiere y no cuenta con un administrador de productos o tareas o un líder de equipo, ya que en scrum la composición del equipo tiene ligeras diferencias de un proyecto en cascada común, pues este cuenta con tres roles específicos, como el product Owner, Development team y el scrum Master. Los tres roles son iguales, pero con ciertas responsabilidades. El Product Owner es el encargado de la visión del proyecto, el control del presupuesto y la priorización y levantamiento de requisitos, igualmente actúa como el mediador entre el dueño del producto y los desarrolladores, siendo así fundamental su papel en el equipo para el éxito de scrum y entregar mayor valor al negocio. Por otro lado, el Scrum Master se encarga de los problemas, ver que las responsabilidades, las prácticas, los valores y las reglas del scrum se sigan adecuadamente y además se ocupa de entrenar al equipo. Finalmente, el Development team, es el conjunto de personas que están autoorganizadas y que se encargan del desarrollo del producto. Y dado que un equipo en scrum es multifuncional, pues están integrados tanto por arquitectos de software, diseñadores, analistas y desarrolladores (Sancheva, 2016).

Figura 5 Roles de scrum y su relación.



Fuente: Scrumstudy (2017).

De igual forma, el flujo de trabajo de scrum se basa en una colaboración estrecha del propietario del producto con el equipo de scrum sobre las iteraciones continuas de la aplicación en evolución. El proceso scrum involucra los tres roles mencionados anteriormente. El bloque más pequeño del scrum es denominado como sprint, el cual cuenta con un equipo pequeño que realiza una tarea asignada en un tiempo limitado, estas tareas son decididas por la acumulación de sprint. Por otro lado, el Sprint backlog es el documento que contiene los requisitos necesarios para trabajar en el sprint. El Product Backlog es el documento que contiene la lista de requisitos y funcionalidades definidas por el dueño del producto, así como también las historias de usuario. Al finalizar cada día se realiza un scrum para evaluar los avances del proyecto designado del día. La meta de cada sprint es entregar un producto final. Y al finalizar dicho sprint se realiza una revisión con el dueño del producto (Srivastava, Bhardwaj y Saraswat, 2017).

III. METODOLOGÍA

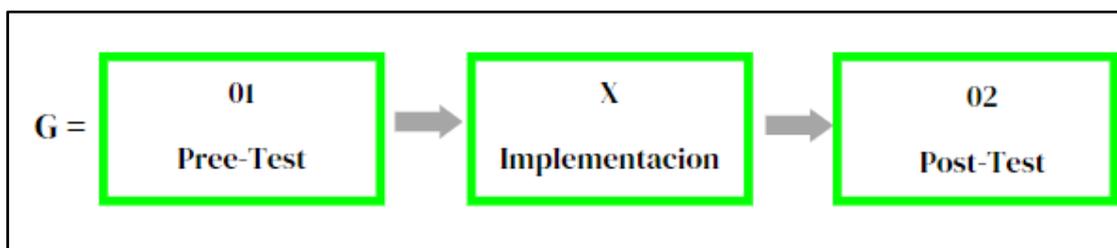
3.1. Tipo y diseño de investigación

El tipo de investigación del proyecto es aplicada, definido por Schubert et al. Como el estudio que surgió a partir de la práctica social, la cual crea resultados aplicables. Pero no siempre terminan en producción, y esto debido a los costos del proyecto. Sin embargo, dichos resultados son importantes para aplicaciones prácticas (2017, p.5). Esta forma de investigar se basa en la usabilidad directa y más no en la creación de teorías. Por ello, se escogió el tipo de investigación aplicada para el proyecto, dado que no se desarrollaron teorías nuevas, sino la implementación de estándares establecidos y herramientas para resolver la problemática en el proceso de venta de la organización.

Asimismo, para el diseño de investigación se eligió el diseño experimental pre-experimental, la cual se basa en el método del pretest y posttest tal como se observa en la figura 4, para Marquis et al. Este diseño de investigación es el que evalúa la situación actual mediante el pretest para luego evidenciar la diferencia con el resultado del posttest, el cual se da gracias a la implementación. Asimismo, este tiene el propósito de validar los objetivos e hipótesis.

Primeramente, se va a realizar la prueba pretest a las ventas y los clientes que se llevará a cabo en fichas de registros las cuales están en el anexo N°4 y N°5, posteriormente se implementará el sistema web basado en la arquitectura de microservicios para finalmente realizar la prueba post test y comparar los resultados de ambas pruebas.

Figura 6 Diseño Pre - Experimental



Fuente: Creación propia.

Donde:

Grupo experimental (G): Es el grupo a la cual se le aplicó la medida para investigar la tasa de adquisición y la tasa de conversión.

Experimento (X): Es la solución que se va a desarrollar que en este caso es un sistema web basado en microservicios.

Pre-Test (O1): Es la prueba al grupo antes de aplicar la solución del sistema web basado en microservicios.

Post-Test (O2): Es la prueba al grupo después de la aplicación de la solución.

3.2. Variables y operacionalización

La actual investigación cuenta con dos variables que son: La arquitectura de microservicios, la cual es la variable independiente y el proceso de ventas que es la variable dependiente, que cuenta con una dimensión (prospección) y está a la vez tiene dos indicadores la tasa de adquisición y la tasa de conversión. En la tabla 1 se muestra la operacionalización de variables. Y para mayor detalle de los indicadores ver anexo N° 2 operacionalización de dimensiones.

Tabla 1 Matriz de Operacionalización de Variables

Variables	Definiciones Conceptuales	Definiciones Operacionales	Dimensión	Indicador	Instrumento	Medición
<p>INDEPENDIENTE</p> <p>Arquitectura de microservicios</p>	<p>Ortiz et al (2022) menciona que la arquitectura de microservicio se basa en el concepto de construir una aplicación como un conjunto de pequeños servicios interconectados, que se comunican a través de protocolos ligeros” (p. 1).</p>	<p>Se realizará una prueba de carga para medir el tiempo de respuesta y la disponibilidad del sistema, además de la prueba de resistencia para comprobar la tolerancia a fallas.</p>				Razón
<p>DEPENDIENTE</p> <p>Proceso de venta</p>	<p>Se entiende como una serie de pasos o etapas que es llevada a cabo por el vendedor al tratar con el cliente, y que tiene como meta es lograr una venta efectiva del servicio o producto de la organización. (Carrasco, 2014, p. 15).</p>	<p>Se registran las ventas y los clientes en fichas de registros para medir los indicadores de la tasa de adquisición y la tasa de conversión respectivamente.</p>	<p>Prospección</p>	<p>Tasa de Adquisición (Domínguez y Hermo 2007)</p>	Ficha de Registro	
			<p>Prospección</p>	<p>Tasa de conversión (Domínguez y Hermo 2007)</p>		

Fuente: Creación Propia

3.3. Población, muestra y muestreo

La población es definida como el grupo de cosas o personas, quienes cuentan con una característica en común. A la cual se les aplica una prueba que pueden ser de varias formas y tipos, con el fin de reunir información que ayude a la investigación (Boddy 2016, p.432).

De acuerdo con lo definido, en nuestra investigación llevada a cabo en la óptica Multi-len's para el proceso de ventas. Para el indicador de la tasa de adquisición y la tasa de conversión se consideró una población de 14 registros de clientes en un periodo de 3 semanas en las cuales se registraron en fichas de registro por día.

Dado que la población de esta investigación es menor a 50 no se realizará el muestreo por lo cual se quedará con la misma cantidad

Tabla 2 Determinación para la población

Indicador	Tipo de población	Tamaño para la población	tiempo
tasa de adquisición	Clientes	14 registros de clientes	Dos semanas
Tasa de conversión	Clientes	14 registros de clientes	Dos semanas

Fuente: Creación Propia

3.4. Técnicas e instrumentos de recolección de datos

En esta investigación se ha establecido usar el fichaje como la técnica de recolección de datos para los dos indicadores, tanto la tasa de adquisición como la tasa de conversión

De igual manera como instrumento de datos se estableció la ficha de registro en donde se recolectó el performance de cada módulo, usando las fichas tomadas

del indicador de la tasa de conversión y la tasa de adquisición. Este instrumento permitirá hacer los registros necesarios para la medición de indicadores de esta investigación en el periodo de 2 semanas respectivamente.

Tabla 3 Instrumentos de recolección de datos

Indicador	Instrumento	Informante
tasa de adquisición	Ficha de Registro	Área de ventas
Tasa de conversión	Ficha de Registro	Área de ventas

Fuente: Creación Propia

Asimismo, se han tomado las fichas de otras investigaciones que utilizaron indicadores iguales, en la cual se van a recolectar los datos en el periodo del estudio, para una adecuada recolección de información, de tal forma que se puedan abastecer las fichas de registro para la medición de las variables.

- **Anexo N° 4:** Ficha de registro “Tasa de adquisición” (Domínguez y Hermo 2007)
- **Anexo N° 5:** Ficha de registro “Tasa de conversión” (Domínguez y Hermo 2007)

A lo largo del desarrollo de la investigación, se llevaron a cabo tanto reuniones a través de videollamadas como visitas presenciales a la organización, con el objetivo de recolectar información.

Por otra parte, en esta investigación se presentó también la confiabilidad del estudio medidos a través de la prueba del test y retest la cual se muestra en las siguientes figuras para cada indicador.

Coeficiente en la correlación de Pearson:

Tabla 4 Confiabilidad para la Tasa de adquisición

		Tasa de Adquisición - Test	Tasa de Adquisición - Retest
Tasa de Adquisición - Test	Correlación de Pearson	1	,813**
	Sig. (bilateral)		,000
	N	14	14
Tasa de Adquisición - Retest	Correlación de Pearson	,813**	1
	Sig. (bilateral)	,000	
	N	14	14

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

Fuente: Creación Propia

Como se aprecia en la Tabla 5, se tiene una confiabilidad de 0.813 del Test y Re-Test, por lo cual el grado de confiabilidad en el instrumento para la tasa de adquisición es de correlación positiva fuerte.

Tabla 5 Confiabilidad para la Tasa de conversión

		Tasa de conversión test	Tasa de conversión retes
Tasa de conversión test	Correlación de Pearson	1	,976**
	Sig. (bilateral)		,000
	N	14	14
Tasa de conversión retes	Correlación de Pearson	,976**	1
	Sig. (bilateral)	,000	
	N	14	14

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

Fuente: Creación Propia

Como se aprecia en la Tabla 6, se tiene una confiabilidad de 0.976 del Test y Re-Test, por lo cual el grado de confiabilidad en el instrumento para la tasa de conversión es de correlación positiva fuerte.

3.5. Procedimientos

En el presente trabajo se realizó una evaluación de la problemática actual de la empresa Multi-len's, en donde se vio que el proceso de venta se llevaba a cabo en medios no automatizados, por lo cual se planteó como variable dependiente de esta investigación. Después de determinar el enfoque se investigaron situaciones semejantes en otras organizaciones tanto nacionales como internacionales, con el fin de estudiar las soluciones que se dieron a los problemas hallados en cada empresa. Asimismo, se realizaron reuniones virtuales como presenciales para la explicación del flujo de proceso, los problemas y los requerimientos que se van a tratar en la solución. Finalmente, luego de tener toda la información, se planteó como solución un software para automatizar el proceso de venta que esté basado en una arquitectura de microservicios, la cual es además es nuestra variable independiente.

También se ha planteado realizar una investigación a detalle de las variables que se emplearon, razón por la cual se recabó información de distintas fuentes de investigación, consultando tanto libros, artículos y tesis, en donde se encontraron problemáticas similares y las soluciones planteadas. De igual forma, gracias a dichas fuentes, se obtuvieron las bases teóricas para el respaldo de este trabajo y los indicadores y dimensiones. Con la información encontrada se propone el tipo de investigación aplicada preexperimental, dado que en el proyecto se va a realizar un producto que será implementado con lo cual se podrá realizar un contraste del pretest y postest de los indicadores empleados. Luego de tener claro las variables y el ambiente se planteó la población para el estudio, de la cual también se sacó la muestra y se determinó el muestreo que se va a emplear y técnica de recolección de datos. De igual forma, se determinó se empleará el software SPSS como el método con el cual se va a analizar los datos de la investigación.

Finalmente, se estableció los aspectos administrativos del trabajo, en la cual se definirán el presupuesto y los recursos necesarios, asimismo el financiamiento para el proyecto y el cronograma para la ejecución del proyecto.

3.6. Método de análisis de datos

En esta investigación se llevará a cabo el análisis descriptivo de las dos variables, en donde se determinará la influencia de la arquitectura de microservicios (variable independiente) en la tasa de adquisición y la tasa de conversión en el proceso de ventas (variable dependiente); para ello se realizará un pretest que permita contemplar el contexto actual de los indicadores y luego de ello se llevará a cabo un post test después de la intervención del sistema implementado.

Además, efectuará el análisis inferencial basándose en la prueba de normalidad a los indicadores de la tasa de adquisición y la tasa de conversión, lo cual se llevará a cabo con el método Shapiro-Wilk dado que la población para ambos indicadores es de 14 registros de clientes, siendo esta la más recomendable para dicha cantidad

Según los resultados que se tengan después de aplicar el método Shapiro-Wilk se establecerá si la distribución es no normal o normal, en el caso de ser normal se emplea la prueba paramétrica T – Student de lo contrario se usará la prueba no paramétrica.

Por otra parte, se aplicará una prueba de hipótesis, para ello se va usar variables para representar ciertas definiciones: TcAs, que significa la tasa de conversión antes de implementar el sistema basado en la arquitectura de microservicios y TcDs que significa la tasa de conversión después de implementar la aplicación basado en la arquitectura de microservicios; : TaAs, que significa la tasa de adquisición antes de implementar el sistema basado en la arquitectura de microservicios y TaDs que significara la tasa de adquisición después de implementar el sistema basado en la arquitectura de microservicios. En el siguiente punto se describen las hipótesis estadísticas.

HE1: Una arquitectura de microservicios mejora la tasa de adquisición en el proceso de ventas de la óptica Multi-len's.

Hipótesis Nula H0: Una arquitectura de microservicios no mejora la tasa de adquisición en el proceso de ventas de la óptica Multi-len's.

$$H_0: TaAs \geq TaDs$$

Hipótesis Alternativa Ha: Una arquitectura de microservicios mejora la tasa de adquisición en el proceso de ventas de la óptica Multi-len's.

$$H_a: TaAs < TaDs$$

HE2: Una arquitectura de microservicios aumenta la tasa de conversión en la gestión de ventas de la óptica Multi-len's.

Hipótesis Nula H0: Una arquitectura de microservicios no aumenta la tasa de conversión en el proceso de ventas de la óptica Multi-len's.

$$H_0: TcAs \geq TcDs$$

Hipótesis Alternativa Ha: Una arquitectura de microservicios aumenta la tasa de conversión en el proceso de ventas de la óptica Multi-len's.

$$H_a: TcAs < TcDs$$

Nivel de Significancia

$\alpha = 5\%$ Error.

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

3.7. Aspectos éticos

Esta investigación se realizó siguiendo las normas establecidas por la universidad en la cual se definen las pautas que se deben seguir para el desarrollo de la investigación, además la información recolectada es completamente profesional, respetando la propiedad intelectual mediante las referencias realizadas de los distintos autores, esto siguiendo los lineamientos

éticos del investigador. Además, se respeta la privacidad de los datos brindados por la empresa, para lo cual también se pidió un consentimiento informado para la divulgación de los resultados obtenidos en esta investigación haciendo el uso correcto de ello. De igual forma se le pidió a la organización el permiso para realizar la investigación en la empresa. Asimismo, no se manipulará la información recolectada.

IV. RESULTADOS

Esta investigación se realizó en dos fases, para establecer las hipótesis en base al diseño Preexperimental. En la primera etapa se realizó una prueba de pretest y los cálculos de cada indicador para conocer el estado actual del indicador, esto antes de poner en marcha el sistema web basado en la arquitectura de microservicios, en la segunda etapa se vuelve a realizar los cálculos, pero esta vez con los datos obtenidos del post test luego de la implementación, de esta forma se hace una comparación de los datos de cada fase.

Los resultados se procesarán mediante el software estadístico SPSS, con el fin de obtener la prueba de normalidad, dependiendo del tamaño de la muestra y así determinar el rechazo o la aceptación de las hipótesis propuestas. La siguiente tabla señala los resultados descriptivos.

Análisis descriptivo

Indicador: Tasa de Adquisición

Para este indicador se encontraron los siguientes resultados, en la cual se analizaron e interpretaron dichos resultados de la siguiente forma:

Tabla 6 Estadísticos descriptivos del indicador tasa de adquisición antes y después de implementar el sistema web basado en la arquitectura de microservicios

		Estadísticos	
		Tasa de Adquisición - PreTest	Tasa de Adquisición - PostTest
N	Válido	14	14
	Perdidos	0	0
Media		1,8929	2,4164
Desv. Desviación		1,11026	1,29410
Mínimo		,67	1,00
Máximo		4,00	5,00

Fuente: Creación Propia

Tal como se muestra en la tabla 6, en relación con el indicador de la tasa de adquisición, el valor en la media muestral del pretest es de 1,8929, la desviación estándar es de 1,10026 lo cual muestra una dispersión alta en los datos, el valor mínimo es de 0,67 es decir hubo registros de clientes en la cual los clientes nuevos no superaron la cantidad de clientes antiguos y el valor máximo es de 4,00 lo cual muestra que hubo más clientes nuevos que antiguos en los registros. Por otro lado, el valor en la media del PostTest es 2.416429 la desviación estándar es de 1,29410 lo cual muestra una dispersión alta en los datos, el valor mínimo es 1.00 67 es decir hubo registros de clientes en la cual la cantidad de los clientes nuevos era igual a la cantidad de clientes antiguos y el valor máximo es 5.00 lo cual muestra que hubo más clientes nuevos que antiguos en los registros. Lo que muestra una diferencia del después y el antes de la implementación del sistema basado en arquitectura de microservicios.

Indicador: Tasa de Conversión

Para este indicador se encontraron los siguientes resultados, en la cual se analizaron e interpretaron dichos resultados de la siguiente forma:

Tabla 7 Estadísticos descriptivos del indicador tasa de conversión antes y después de implementar el sistema web basado en la arquitectura de microservicios

Estadísticos

		Tasa de Conversión - PreTest	Tasa de Conversión - PostTest
N	Válido	14	14
	Perdidos	0	0
Media		,5821	,8336
Desv. Estandar		,27451	,12983
Mínimo		,20	,67
Máximo		1,00	1,00

Fuente: Creación Propia

Por otro lado, para el nivel de conversión indicado en la Tabla 7, se puede observar que el valor en la media muestral del pretest es 0.5821 la desviación estándar es de 0,27451 lo cual muestra una dispersión en los datos, el valor mínimo es de 0.20 es decir hubo registros de clientes en la cual la cantidad de los clientes nuevos es menor a la cantidad de peticiones prospectivas recibidas y el máximo es de 1.00 es decir la cantidad de clientes nuevos es igual a la cantidad de peticiones prospectivas recibidas. Por otro lado, el valor medio del post-test es 0.8957 la desviación estándar es de 0,12983 lo cual muestra una dispersión en los datos, el valor mínimo es 0.67 lo cual es mayor al valor mínimo obtenido en PreTest y el valor máximo es 1.00 es decir la cantidad de clientes nuevos es igual a la cantidad de peticiones prospectivas recibidas. Lo que muestra una diferencia del después y el antes de la implementación del sistema basado en la arquitectura de microservicios.

Análisis inferencial

Prueba de Normalidad

Se realizó la prueba de normalidad para los dos indicadores, tanto la tasa de adquisición como la tasa de conversión a partir del método Shapiro-Wilk, dado que la muestra es menor de 50. Los datos obtenidos de cada indicador se procesarán en el SPSS.

Si:

Sig < 0.05 es una distribución no normal

Sig \geq 0.05 es una distribución normal

Donde:

Sig.: P – Es el mérito o el nivel del contraste crítico

Los datos obtenidos de cada indicador son los siguientes:

Indicador: Tasa de Adquisición

Con el fin de escoger la prueba idónea para las hipótesis; se comprobaron los datos para determinar si la naturalidad de su distribución es normal o no normal.

Tabla 8 Prueba de normalidad tasa de adquisición

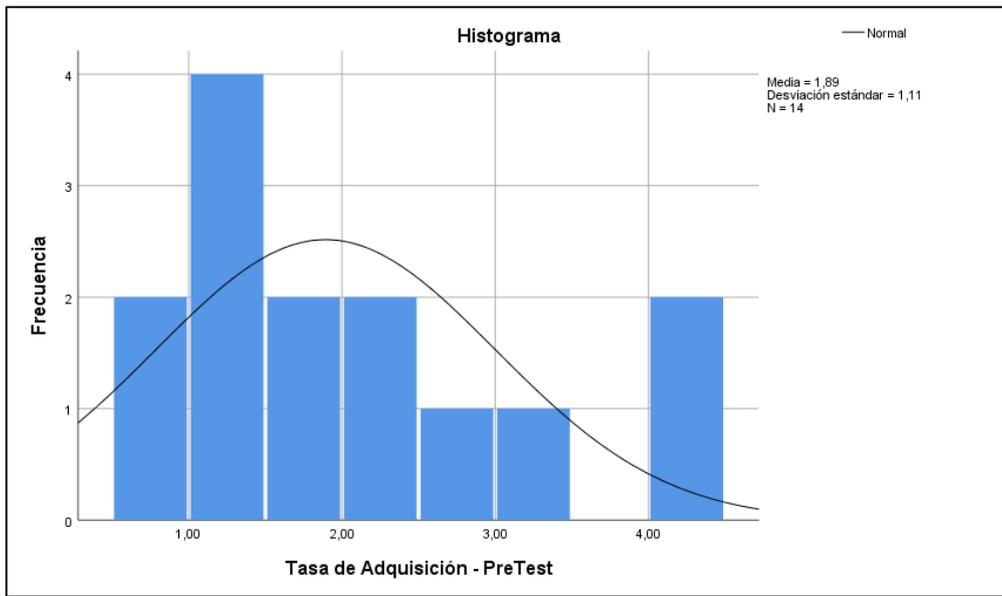
	Shapiro-Wilk		
	Estadístico	gl	Sig.
Tasa de Adquisición - PreTest	,876	14	,052
Tasa de Adquisición - PostTest	,892	14	,086

Fuente: Creación Propia

En los resultados de la prueba que se muestran en la tabla 8, indica que el sig. de la tasa de adquisición en el pretest es de 0.052 la cual es mayor a 0.05, siendo así una distribución normal. Por otro lado, las pruebas del post test nos muestran un valor del sig. del 0.086 siendo este mayor a 0.05, por lo que la tasa de adquisición se distribuye normalmente. Lo que confirma que la distribución presentada en los datos comparados de la muestra es de tipo normal.

En la siguiente figura, se muestra el pretest, en la cual la media es 1.89 y la desviación estándar es de 1,11. Tal como se explicó en el análisis descriptivo.

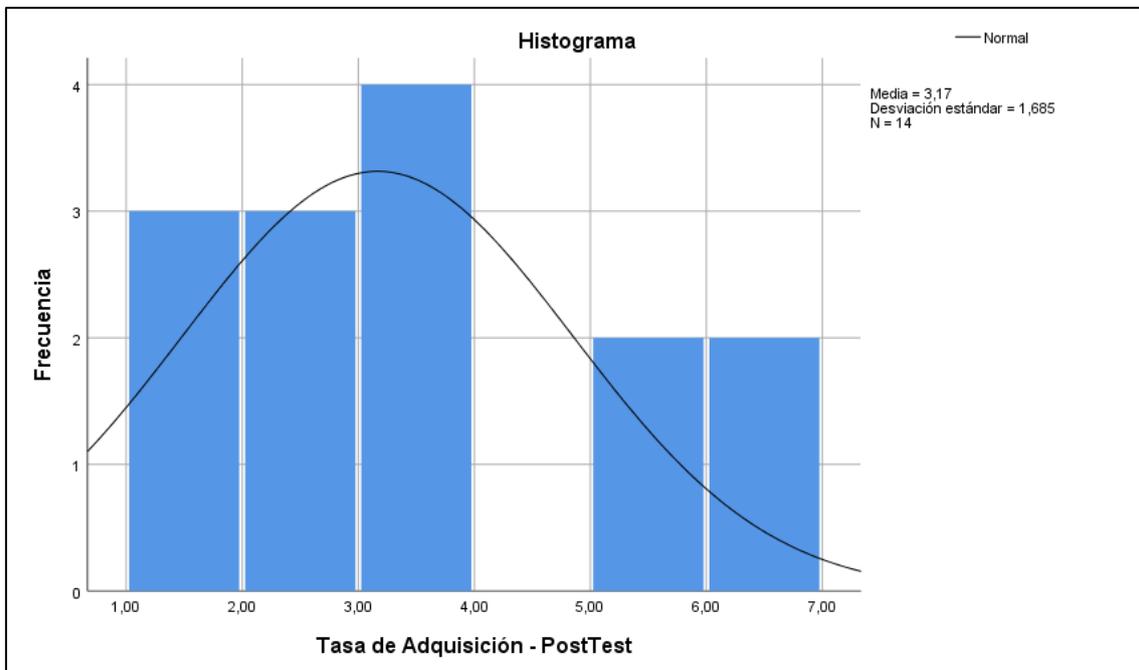
Figura 7 Prueba de normalidad de la tasa de adquisición antes de la implementación del sistema



Fuente: Creación Propia

En la siguiente figura, se muestra el PostTest de la tasa de adquisición, en la cual la media es 3.17 y la desviación estándar es de 1,685. Tal como se explicó en el análisis descriptivo.

Figura 8 Prueba de normalidad de la tasa de adquisición después de la implementación del sistema



Fuente: Creación Propia

Indicador: Tasa de Conversión

Con el objetivo de escoger la prueba idónea para las hipótesis; se comprobaron los datos hallados de la tasa de conversión para determinar si la naturalidad de su distribución es normal o no normal.

Tabla 9 Prueba de Normalidad Tasa de Conversión

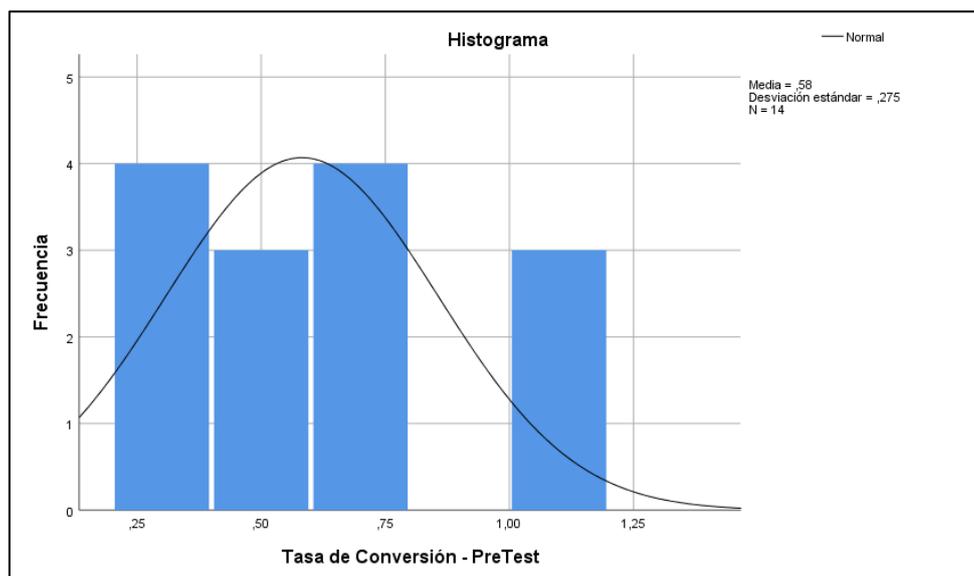
	Pruebas de normalidad		
		Shapiro-Wilk	
Tasa de Conversión - PreTest	,909	14	,153
Tasa de Conversión - PostTest	,876	14	,051

Fuente: Creación Propia

En los resultados de la prueba que se muestran en la tabla 9, indica que el sig. de la tasa de conversión en el pretest es de 0.153 la cual es mayor a 0.05, siendo así una distribución normal. Por otro lado, las pruebas del post test nos muestran un valor del sig. del 0.051 siendo este mayor a 0.05, por lo que la tasa de adquisición se distribuye normalmente. Lo que confirma que la distribución presentada en los datos comparados de la muestra es de tipo normal.

En la siguiente figura, se muestra el pretest, en la cual la media es 0.58 y la desviación estándar es de 0.275. Tal como se explicó en el análisis descriptivo.

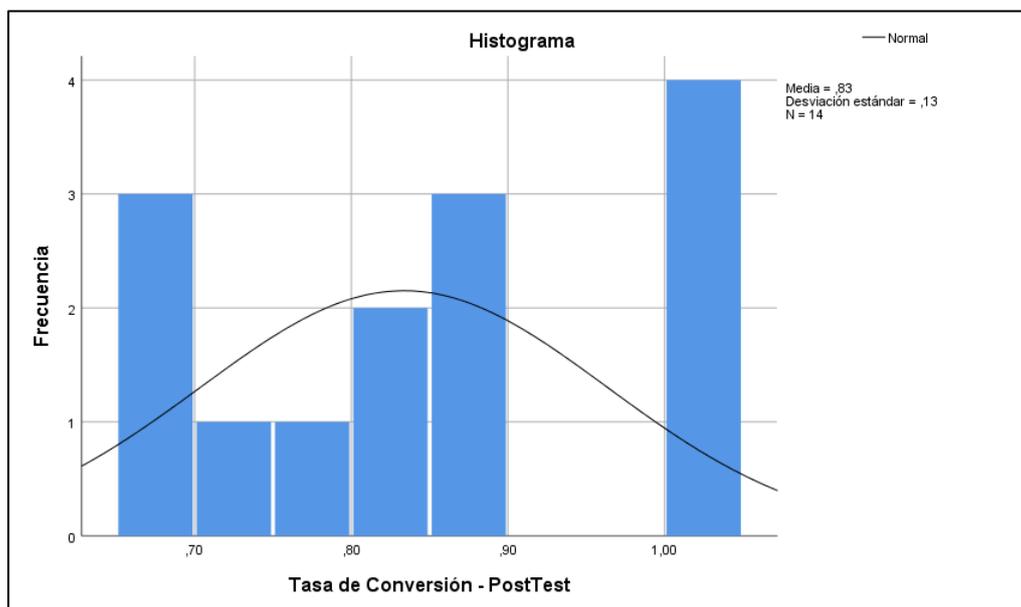
Figura 9 Prueba de normalidad de la tasa de conversión antes de la implementación del sistema



Fuente: Creación Propia

En la siguiente figura, se muestra el PostTest, en la cual la media es 0.83 y la desviación estándar es de 0.13. Tal como se explicó en el análisis descriptivo.

Figura 10 Prueba de normalidad de la tasa de conversión después de la implementación del sistema



Fuente: Creación Propia

Prueba de Hipótesis

Hipótesis de Investigación 1:

- **H1:** Una arquitectura de microservicios mejora la tasa de adquisición en el proceso de ventas de la óptica Multi-len's.
- **Indicador:** Tasa de Adquisición

Hipótesis Estadísticas

Definición de variables:

NEa: Tasa de Adquisición antes de utilizar el sistema web basado en la arquitectura de microservicios.

NEp: Tasa de Adquisición después de utilizar el sistema web basado en la arquitectura de microservicios.

- **Hipótesis Nula H0:** Una arquitectura de microservicios no mejora la tasa de adquisición en el proceso de ventas de la óptica Multi-len's.

$$H_0: NE_a \leq NE_p$$

El indicador actual del sistema es preferible a que el indicador con el sistema

- **Hipótesis Alternativa Ha:** Una arquitectura de microservicios mejora la tasa de adquisición en el proceso de ventas de la óptica Multi-len's.

$$H_a: NE_a > NE_p$$

El indicador con la aplicación web es mejor que el indicador sin la aplicación web

Para la verificación en la hipótesis se empleará la prueba de T-Student dado que para este indicador se tiene una distribución normal, siendo así paramétrica. En la siguiente tabla se muestra dicha prueba.

Prueba de T-Student

		Media	t	gl	Sig. (bilateral)
Par 1	Tasa de Adquisición - PreTest - Tasa de Adquisición - PostTest	-1,27357	-3,200	13	,007

Fuente: Creación Propia

El valor de Sig. es de 0,007 siendo esto menor que 0.05. Por lo cual es rechazado la hipótesis nula y se acepta la hipótesis alterna. Es decir, una arquitectura de microservicios mejora la tasa de adquisición en el proceso de ventas de la óptica Multi-len's.

Hipótesis de Investigación 2:

- **H2:** Una arquitectura de microservicios aumenta la tasa de conversión en el proceso de ventas de la óptica Multi-len's.
- **Indicador:** Tasa de Conversión

Hipótesis Estadísticas

Definición de variables:

NEa: Tasa de Conversión antes de utilizar el sistema web basado en la arquitectura de microservicios.

NEp: Tasa de Conversión después de utilizar el sistema web basado en la arquitectura de microservicios.

- **Hipótesis Nula H0:** Una arquitectura de microservicios no aumenta la tasa de conversión en el proceso de ventas de la óptica Multi-len's.

$$H_0: NEa \leq NEp$$

El indicador actual es preferible a que el indicador con el sistema

- **Hipótesis Alternativa Ha:** Una arquitectura de microservicios aumenta la tasa de conversión en el proceso de ventas de la óptica Multi-len's.

$$H_a: NEa > NEp$$

El indicador con la aplicación web es mejor que el indicador sin la aplicación web.

Para la verificación en la hipótesis se empleará la prueba de T-Student dado que para este indicador se tiene una distribución normal, siendo así paramétrica. En la siguiente tabla se muestra dicha prueba.

Prueba de T-Student

		Media	t	gl	Sig. (bilateral)
Par 1	Tasa de Conversión - PreTest - Tasa de Conversión - PostTest	-,25143	-3,696	13	,003

Fuente: Creación Propia

El valor de Sig. es de 0,003 siendo esto menor que 0.05. Por lo cual es rechazado la hipótesis nula y se acepta la hipótesis alterna. Es decir, una arquitectura de

microservicios aumenta la tasa de conversión en el proceso de ventas de la óptica Multi-len's.

Tal como se pudo observar las dos hipótesis específicas planteadas mejoran y aumentan la tasa de adquisición y la tasa de conversión respectivamente. Por ende, se confirma la hipótesis general de que una arquitectura de microservicios mejora el proceso de ventas de la óptica Multi-len's:

V. DISCUSIÓN

Con la implementación del sistema, se pudo observar las mejoras en los indicadores de esta investigación con respecto a los resultados obtenidos en el pretest.

En el presente trabajo de investigación se tuvo como resultado que una arquitectura de microservicios mejoró la tasa de adquisición en el proceso de ventas, dado que en el PreTest se obtuvo 1.89 y luego de ser implementado el sistema se alcanzó valor de 3.17 en el PostTest. Lo cual significa un incremento del 67.72% con respecto a la situación anterior. Y esto concuerda con los resultados que obtuvo el autor Zafra (2020), en su investigación E-commerce basado en microservicios para el proceso de ventas en la empresa COMATPERU S.A.C. en donde concluyó que el e-commerce tuvo un impacto en la tasa de adquisición en la cual se mostró un aumento de un 43.95% a 67.25%. Asimismo, en la investigación de Mora M., Noraimar Y. (2019). Se desarrolló un e-commerce escalable, con una alta disponibilidad y rendimiento, para lo cual emplearon la arquitectura de microservicios, en donde los resultados luego de someter a la arquitectura a una serie de pruebas de carga, de funcionalidad y de resistencia, demostraron que dicha arquitectura era lo ideal para el tipo de sistema que la empresa quería construir como el sistema de ventas.

Por otra parte, para el segundo objetivo se obtuvo que una arquitectura de microservicios aumenta la tasa de conversión en el proceso de ventas, dado que se mostró un incremento de un 43.10%, pues en el pretest se obtuvo un resultado de 0.58 y en el PostTest 0.83. De igual manera, Martínez J. (2020) en su investigación Sistema web para el proceso de ventas en la Empresa Wilmer venta de repuestos de sistema eléctrico en Santa Anita. En la cual llegó a la conclusión de que un sistema web tiene un impacto en la tasa de conversión pues se mostró un aumento de un 0.29 a 0.77.

En relación con los dos indicadores, Cahyo y Candiwan (2020) en su artículo científico, pudieron contrastar que la implementación del sistema ayudo a

que los productos fueran conocidos por una cantidad mayor de personas, lo cual mejoró la tasa de conversión y la tasa de adquisición de la organización.

Los resultados presentados en la investigación que se llevó a cabo evidencian que la implementación de un sistema web basado en la arquitectura de microservicios permiten llevar a cabo el proceso de ventas de manera eficiente, lo cual aumenta tanto la tasa de adquisición como la tasa de conversión. Por ende, se puede afirmar que una arquitectura de microservicios mejora el proceso de ventas. Y con este resultado concuerdan, Djatmiko et al. (2017), pues en su investigación luego de la implementación del sistema se mostró una mejora en el registro y análisis de ventas gracias a que dicha aplicación agiliza el proceso, además de que cada incidente de ventas fue registrado, reduciendo así las pérdidas para la empresa. De igual forma Katherin (2020), luego de evaluar sus indicadores menciona que la aplicación web tuvo mejoras en el proceso de ventas, ya que obtuvo un aumento del 31%.

VI. CONCLUSIONES

A partir de los resultados obtenidos en esta investigación, se pudo llegar a las siguientes conclusiones:

Para el objetivo general de la investigación el cual fue determinar la influencia de una arquitectura de microservicios en el proceso de ventas de la óptica Multi-len's, en donde se pudo contrastar con los resultados de otras investigaciones tales como el de Djatmiko et al. (2017) y Katherin (2020). Los cuales nos llevan a concluir que hay una influencia positiva en cuanto a la arquitectura de microservicios en el proceso de ventas, pues dicha arquitectura permitió construir un sistema escalable, con un rendimiento y disponibilidad alto el cual permitió administrar las ventas de forma eficiente, pues la información estaba donde y cuando se necesitaba tanto para los clientes como para los dueños del negocio, eliminando así muchas de las tareas manuales que se realizaban en la empresa y que consumía tiempo y recurso.

En cuanto al primer objetivo específico el cual fue determinar la influencia de una arquitectura de microservicios en la tasa de adquisición del proceso de ventas de la óptica Multi-len's, se pudo contrastar los resultados con investigaciones de diferentes autores tales como el de Mora M., Noraimar Y. (2019) y Zafra (2020) cuyos resultados coinciden con lo hallado en esta investigación, por lo cual se llega a la conclusión de que un sistema web desarrollado en una arquitectura de microservicios mejora la tasa de adquisición en el proceso de ventas de la óptica Multi-len's. Teniendo un aumento de 67.72% con respecto a la situación anterior a la implementación de dicho sistema.

Finalmente, con respecto al segundo objetivo específico el cual es determinar la influencia de una arquitectura de microservicios en la tasa de conversión para el proceso de venta de la óptica Multi-len's, se logró contrastar con los resultados obtenidos por Cahyo y Candiwan (2020) y Martínez J. (2020), quienes también coinciden con el resultado de esta investigación. Por ende, se puede concluir que un sistema web desarrollado en una arquitectura de microservicios mejora la tasa de conversión en el

proceso de ventas de la óptica Multi-len's. Teniendo un aumento de 45.16% con respecto a la situación anterior a la implementación de dicho sistema. Esto gracias a las ventajas que ofrece una arquitectura de microservicios.

VII. RECOMENDACIONES

Se recomienda investigar a mayor detalle sobre la arquitectura de microservicios y los patrones de diseños de software que se pueden aplicar a este tipo de sistemas ya que es fundamental que se tomen en cuenta dichos patrones en el desarrollo de la aplicación, lo cual permita cumplir con todos los requerimientos y a la vez tener una mejora continua en el sistema para el apoyo del proceso de ventas.

Se recomienda a la empresa que adopte la cultura DevOps, dado que al ser un proyecto basado en una arquitectura de microservicios es importante emplear las herramientas y los estándares que tienen realización con la entrega continua y la integración continua (CD/CI). Por lo cual es importante implementar la filosofía DevOps pues este le permitirá tener agilidad en el despliegue y además mejorar la calidad del software. Dado que entre los principales desafíos que las empresas deben enfrentar esta el uso de las metodologías ágiles, ya que muchas de las organizaciones tienen una forma de trabajo tradicional en la cual separan los equipos de trabajo. Por lo que es importante que adopten dichas metodologías, pues el desarrollo de un microservicio va siempre de la mano con estas metodologías para llegar al éxito.

Se recomienda evaluar si es que es necesario emplear la arquitectura de microservicios ya que no es para todos los sistemas. Debido a los costos, por lo que realizar una migración o desarrollo de sistemas pequeños que no manejen una transacción de datos grande puede generar más gastos que ganancias o beneficios, dado que necesitan de equipos especializados en temas de microservicios lo cual también lleva a los altos costos y no solo en el personal sino también en los equipos de software que son necesarios para desarrollar e implementar microservicios. Por ende, es recomendable el uso de este tipo de arquitectura para sistemas que reciban varias peticiones al mismo tiempo. Así mismo para realizar la descomposición del sistema en componentes o servicios más pequeños es necesario que el sistema tenga un tamaño considerable o que esté pensado en crecer a tal punto que se pueda dividir en varios módulos.

Se recomienda desarrollar las APIs implementando HATEOS para cumplir con todos los niveles de madurez, de tal forma que se consideren APIs RESTful.

REFERENCIAS

LIU, H., JURDANA, I., LOPAC, N. y WAKABAYASHI, N., 2022. BlueNavi: A Microservices Architecture-Styled Platform Providing Maritime Information Sustainability [en línea], vol. 14, no. 4, pp. 2173. ISSN 2071-1050. DOI 10.3390/su14042173. Disponible en: <https://www.mdpi.com/2071-1050/14/4/2173>.

BUCCHIARONE, A., DRAGONI, N., DUSTDAR, S., LARSEN, S.T. y MAZZARA, M., 2018. From Monolithic to Microservices: An Experience Report from the Banking Domain. IEEE Software [en línea], vol. 35, no. 3, pp. 50-55. ISSN 0740-7459. DOI 10.1109/MS.2018.2141026. Disponible en: <https://ieeexplore.ieee.org/document/8354415/>.

SOEGOTO, D.S. y CICA, C., 2018. Design of Web-based Sales Information System on Fashion Shop in Bandung, Indonesia. IOP Conference Series: Materials Science and Engineering [en línea], vol. 407, no. 1, pp. 012023. ISSN 1757-899X. DOI 10.1088/1757-899X/407/1/012023. Disponible en: <https://iopscience.iop.org/article/10.1088/1757-899X/407/1/012023>.

YULIANTO, H.D. y FAUZI, R., 2020. Design of Web-based Online Sales Information System. IOP Conference Series: Materials Science and Engineering [en línea], vol. 879, no. 1, pp. 012007. ISSN 1757-8981. DOI 10.1088/1757-899X/879/1/012007. Disponible en: <https://iopscience.iop.org/article/10.1088/1757-899X/879/1/012007>.

DODY FIRMANSYAH, M., BACHTIAR, S., SFENRIANTO, S. y ROBERT KABURUAN, E., 2019. Sales Information System Using Web for Small Business (Case Study: Cv. Tanaka Service). International Journal of Mechanical Engineering and Technology (IJMET) [en línea], vol. 10, no. 3, pp. 1696-1702. ISSN 0976-6359. Disponible en: <http://www.iaeme.com/IJMET/index.asp1696http://www.iaeme.com/ijmet/issues.asp?JType=IJMET&VType=10&IType=3http://www.iaeme.com/IJMET/issues.asp?JType=IJMET&VType=10&IType=3>.

DJATMIKO, M.B., HUSAIN, A., MAULANI, G. y NIRMALASARI, L., 2017. Analyze and Record a Series of Corporate Sales Transactions On Web Based Accounting Online System. *Aptisi Transactions on Management (ATM)* [en línea], vol. 1, no. 2, pp. 103-115. ISSN 2622-6804. DOI 10.33050/atm.v1i2.691. Disponible en: <http://ijc.ilearning.co/index.php/ATM/article/view/691>.

EZEONWUMELU, A., EUNICE, A., EZENUGU, I. 2017. Development of Sales and Inventory Workflow Management Information System Web Portal for Petrospan Integrated Services, Eket, Akwa Ibom State, Nigeria. *Mathematical and Software Engineering* [en línea], vol. 3, no. 1, pp. 57-66.

Disponible en:

https://explore.openaire.eu/search/publication?articleId=dedup_wf_001::2fdb0e147d02fe857ba0fb2a07c681c5%0A.

CAHYO, M.R.D. y CANDIWAN, C., 2020. Analysis and Design of Sales Information System on Web-Based E-Commerce in Yoga Farm Catfish Breeding Business Using UML. *JURNAL MEDIA INFORMATIKA BUDIDARMA* [en línea], vol. 4, no. 3, pp. 683. ISSN 2548-8368. DOI 10.30865/mib.v4i3.2106.

Disponible en:

<https://ejurnal.stmik-budidarma.ac.id/index.php/mib/article/view/2106>.

NORAIMAR, Y., MORA, M., 2019. *Arquitectura Basada en Microservicios para Sistemas de E-commerce de la Empresa LCC OpenTech, C.A.* [en línea]. S.l.: Universidad Católica Andrés Bello.

Disponible en: <http://catalogo-gy.ucab.edu.ve/documentos/tesis/35583.pdf>.

DOMINGUEZ, A. y HERMO, S., 2007. Métricas del marketing [en línea]. Madrid: s.n. ISBN 9788473565219.

Disponible en:

https://www.esic.edu/documentos/esicpress/archivos_download/metricas_del_marketing.pdf

MEDEL, Y., CASTRO, F.E., FIGUEREDO, A.E., LEYVA, A.R. y ALMAGUEL, A., 2021. Sistema informático para la distribución de uniforme escolar. Caso de estudio: provincia de Granma, Cuba. *Revista de Investigación en Tecnologías de la Información* [en línea], vol. 9, no. 19, pp. 1-15. ISSN 23870893. DOI 10.36825/RITI.09.19.001.

Disponible en: <https://www.riti.es/ojs2018/inicio/index.php/riti/article/view/360>.

GOS, K. y ZABIEROWSKI, W., 2020. The Comparison of Microservice and Monolithic Architecture. 2020 IEEE XVIth International Conference on the Perspective

Technologies and Methods in MEMS Design (MEMSTECH) [en línea], pp. 150-153. ISSN 25735373. DOI 10.1109/MEMSTECH49584.2020.9109514.

Disponible en: <https://ieeexplore.ieee.org/document/9109514/>.

AL-DEBAGY, O. y MARTINEK, P., 2018. A Comparative Review of Microservices and Monolithic Architectures. 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI) [en línea]. Hungría: IEEE, pp. 000149-000154. ISBN 978-1-7281-1117-9. DOI 10.1109/CINTI.2018.8928192. Disponible en: <https://ieeexplore.ieee.org/document/8928192/>.

BUCCHIARONE, A., DRAGONI, N., DUSTDAR, S., LARSEN, S.T. y MAZZARA, M., 2018. From Monolithic to Microservices: An Experience Report from the Banking Domain. IEEE Software [en línea], vol. 35, no. 3, pp. 50-55. ISSN 0740-7459. DOI 10.1109/MS.2018.2141026.

Disponible en: <https://ieeexplore.ieee.org/document/8354415/>.

LARRUCEA, X., SANTAMARIA, I., COLOMO-PALACIOS, R. y EBERT, C., 2018. Microservices. IEEE Software [en línea], vol. 35, no. 3, pp. 96-100. ISSN 07407459. DOI 10.1109/MS.2018.2141030.

Disponible en: <https://ieeexplore.ieee.org/document/8354423/>.

KURYAZOV, D., JABBOROV, D. y KHUJAMURATOV, B., 2020. Towards Decomposing Monolithic Applications into Microservices. 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT) [en línea], pp. 1-4. DOI 10.1109/AICT50176.2020.9368571. Disponible en: <https://ieeexplore.ieee.org/document/9368571/>.

RAJ, V. y RAVICHANDRA, S., 2018. Microservices: A perfect SOA based solution for Enterprise Applications compared to Web Services. 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) [en línea]. S.I.: IEEE, pp. 1531-1536. ISBN 978-1-5386-2440-1. DOI 10.1109/RTEICT42901.2018.9012140. Disponible en: <https://ieeexplore.ieee.org/document/9012140/>.

BERKUNSKYI, Y., KNYRIK, K., FARIONOVA, T. y SMYKODUB, T., 2017. Using microservices in educational applications of IT-company. 2017 IEEE First Ukraine

Conference on Electrical and Computer Engineering (UKRCON) [en línea]. S.I.: IEEE, pp. 1208-1211. ISBN 978-1-5090-3006-4. DOI 10.1109/UKRCON.2017.8100443. Disponible en: <http://ieeexplore.ieee.org/document/8100443/>.

INDRASIRI, K. y SIRIWARDENA, P., 2018. Microservices for the Enterprise [en línea]. Berkeley, CA: Apress. ISBN 978-1-4842-3857-8. Disponible en: <http://link.springer.com/10.1007/978-1-4842-3858-5>.

WOLFF, E., 2016. Microservices - Flexible Software Architecture [en línea]. Addison We. USA: s.n. ISBN 9780134602417. Disponible en: [https://sd.blackball.lv/library/Microservices_Flexible_Software_Architectures_\(2016\).pdf](https://sd.blackball.lv/library/Microservices_Flexible_Software_Architectures_(2016).pdf).

RICHARDSON, C. y SMITH, F., 2016. Microservices - From Design to Deployment [en línea]. USA: s.n. ISBN 978-0-134-60241-7. Disponible en: <https://www.nginx.com/blog/microservices-from-design-to-deployment-ebook-nginx/>.

SONG, M., ZHANG, C. y HAIHONG, E., 2018. An Auto Scaling System for API Gateway Based on Kubernetes. 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS) [en línea]. S.I.: IEEE, pp. 109-112. ISBN 978-1-5386-6565-7. DOI 10.1109/ICSESS.2018.8663784. Disponible en: <https://ieeexplore.ieee.org/document/8663784/>.

Nilsson, D. (2022). How to Design a Sales Process for B2B Sales. [online] Daniel-one.com. Disponible en: <https://daniel-one.com/design-sales-process-funnel-b2b/>.

PARSALIS, A., 2019. Sales process optimization, in Business-to-Business (B2B) organizations with solution selling methodology [en línea]. S.I.: Selinus University of Science and literature. Disponible en: https://www.uniselinus.education/sites/default/files/2021-06/Tesi_Parsalis.pdf.

KOTLER, P. y ARMSTRONG, G., 2012. Marketing [en línea]. Decimocuar. S.I.: s.n. ISBN 9786073214209. Disponible en: [https://frrq.cvg.utn.edu.ar/pluginfile.php/14584/mod_resource/content/1/Fundamentos del Marketing-Kotler.pdf](https://frrq.cvg.utn.edu.ar/pluginfile.php/14584/mod_resource/content/1/Fundamentos_del_Marketing-Kotler.pdf).

KLINBUA, K. y VATANAWOOD, W., 2017. Translating TOSCA into docker-compose YAML file using ANTLR. 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS) [en línea]. S.I.: IEEE, pp. 145-148. ISBN 978-1-5386-0497-7. DOI 10.1109/ICSESS.2017.8342884. Disponible en: <http://ieeexplore.ieee.org/document/8342884/>.

JABRI, M., 2021. Java Programming Language Report. Database and Network Journal [en línea], vol. 35, no. 5, pp. 11-15. ISSN 02654490. DOI 10.13140/RG.2.2.33652.48005. Disponible en: <https://www.researchgate.net/publication/351088042>.

CARNELL, J. y HUAYLUPO, I., 2021. Spring Microservices in Action, Second Edition [en línea]. 2. Estados Unidos: s.n. ISBN 9781617293986. Disponible en: <https://www.manning.com/books/spring-microservices-in-action-second-edition>.

SCHUBERT, O.T., RÖST, H.L., COLLINS, B.C., ROSENBERGER, G. y AEBERSOLD, R., 2017. Quantitative proteomics: challenges and opportunities in basic and applied research. Nature Protocols [en línea], vol. 12, no. 7, pp. 1289-1294. ISSN 1754-2189. DOI 10.1038/nprot.2017.040. Disponible en: <http://www.nature.com/articles/nprot.2017.040>.

MARQUIS, N., LARIVÉE, P., SAEY, D., DUBOIS, M.-F. y TOUSIGNANT, M., 2015. In-Home Pulmonary Telerehabilitation for Patients with Chronic Obstructive Pulmonary Disease: A Pre-experimental Study on Effectiveness, Satisfaction, and Adherence. Telemedicine and e-Health [en línea], vol. 21, no. 11, pp. 870-879. ISSN 1530-5627. DOI 10.1089/tmj.2014.0198. Disponible en: <https://www.liebertpub.com/doi/10.1089/tmj.2014.0198>.

BODDY, C.R., 2016. Sample size for qualitative research. Qualitative Market Research: An International Journal [en línea], vol. 19, no. 4, pp. 426-432. ISSN 1352-2752. DOI 10.1108/QMR-06-2016-0053. Disponible en: <https://www.emerald.com/insight/content/doi/10.1108/QMR-06-2016-0053/full/html>.

BALAKRISHNAN, S., WAINWRIGHT, M.J. y YU, B., 2017. Statistical guarantees for the EM algorithm: From population to sample-based analysis. The Annals of Statistics [en línea], vol. 45, no. 1, pp. 77-120. ISSN 0090-5364. DOI 10.1214/16-AOS1435.

Disponible en: <https://projecteuclid.org/journals/annals-of-statistics/volume-45/issue-1/Statistical-guarantees-for-the-EM-algorithm--From-population-to/10.1214/16-AOS1435.full>.

SHAFIEE, S., WAUTELET, Y., HVAM, L., SANDRIN, E. y FORZA, C., 2020. Scrum versus Rational Unified Process in facing the main challenges of product configuration systems development. Journal of Systems and Software [en línea], vol. 170, pp. 110732. ISSN 01641212. DOI 10.1016/j.jss.2020.110732. Disponible en: <https://doi.org/10.1016/j.jss.2020.110732>.

SACHDEVA, S., 2016. Scrum Methodology. International Journal Of Engineering And Computer Science [en línea], vol. 5, no. 16792, pp. 16792-16800. ISSN 23197242. DOI 10.18535/ijecs/v5i6.11. Disponible en: http://ijecs.in/issue/v5-i6/11_ijecs.pdf.

SCRUMSTUDY, 2017. La Guía para el Cuerpo de Conocimiento de Scrum (Guía SBOKTM) [en línea]. 3. S.l.: s.n. ISBN 9780989925204. Disponible en: <https://books.apple.com/es/book/una-guía-para-el-cuerpo-de-conocimiento-de-scrum/id1478177119>.

SACHDEVA, S., 2016. Scrum Methodology. International Journal Of Engineering And Computer Science [en línea], vol. 5, no. 16792, pp. 16792-16800. ISSN 23197242. DOI 10.18535/ijecs/v5i6.11. Disponible en: http://ijecs.in/issue/v5-i6/11_ijecs.pdf.

SRIVASTAVA, A., BHARDWAJ, S. y SARASWAT, S., 2017. SCRUM model for agile methodology. 2017 International Conference on Computing, Communication and Automation (ICCCA) [en línea]. S.l.: IEEE, pp. 864-869. ISBN 978-1-5090-6471-7. DOI 10.1109/CCAA.2017.8229928. Disponible en: <http://ieeexplore.ieee.org/document/8229928/>.

RAJ, V. y RAVICHANDRA, S., 2018. Microservices: A perfect SOA based solution for Enterprise Applications compared to Web Services. 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) [en línea]. S.l.: IEEE, pp. 1531-1536. ISBN 978-1-5386-2440-1. DOI 10.1109/RTEICT42901.2018.9012140.

RADEMACHER, F., SACHWEH, S. y ZUNDORF, A., 2017. Differences between Model-Driven Development of Service-Oriented and Microservice Architecture. 2017 IEEE International Conference on Software Architecture Workshops (ICSAW) [en línea]. S.l.: IEEE, pp. 38-45. ISBN 978-1-5090-4793-2. DOI 10.1109/ICSAW.2017.32.

KOSCHEL, A., ASTROVA, I. y DOTTERL, J., 2017. Making the move to microservice architecture. 2017 International Conference on Information Society (i-Society) [en línea]. S.l.: IEEE, pp. 74-79. ISBN 978-1-908320-80-3. DOI 10.23919/i-Society.2017.8354675.

TRIHINAS, D., TRYFONOS, A., DIKAIKOS, M.D. y PALLIS, G., 2018. DevOps as a Service: Pushing the Boundaries of Microservice Adoption. IEEE Internet Computing [en línea], vol. 22, no. 3, pp. 65-71. ISSN 1089-7801. DOI 10.1109/MIC.2018.032501519.

AHMED, Shahbaz, RAZZAQ, A., ULLAH, S. y AHMED, Salman, 2018. Matrix Clustering based Migration of System Application to Microservices Architecture. International Journal of Advanced Computer Science and Applications [en línea], vol. 9, no. 1, pp. 284-296. ISSN 21565570. DOI 10.14569/IJACSA.2018.090139.

RAJ, V. y SADAM, R., 2021. Performance and complexity comparison of service oriented architecture and microservices architecture. International Journal of Communication Networks and Distributed Systems [en línea], vol. 27, no. 1, pp. 100. ISSN 1754-3916. DOI 10.1504/IJCND.2021.116463.

BAŠKARADA, S., NGUYEN, V. y KORONIOS, A., 2020. Architecting Microservices: Practical Opportunities and Challenges. Journal of Computer Information Systems [en línea], vol. 60, no. 5, pp. 428-436. ISSN 0887-4417. DOI 10.1080/08874417.2018.1520056.

BOGNER, J., FRITZSCH, J., WAGNER, S. y ZIMMERMANN, A., 2021. Industry practices and challenges for the evolvability assurance of microservices. Empirical Software Engineering [en línea], vol. 26, no. 5, pp. 104. ISSN 1382-3256. DOI 10.1007/s10664-021-09999-9.

CHEN, R., LI, S. y LI, Z., 2017. From Monolith to Microservices: A Dataflow-Driven Approach. 2017 24th Asia-Pacific Software Engineering (APSEC) [en línea], vol. 2017-Decem, no. February, pp. 466-475. ISSN 15301362. DOI 10.1109/APSEC.2017.53.

BAKSHI, K., 2017. Microservices-based software architecture and approaches. 2017 IEEE Aerospace [en línea], pp. 1-8. ISSN 1095323X. DOI 10.1109/AERO.2017.7943959.

CHEN, L., 2018. Microservices: Architecting for Continuous Delivery and DevOps. 2018 IEEE Software Architecture (ICSA) [en línea]. S.l.: IEEE, pp. 15. ISBN 978-1-5386-6398-1. DOI 10.1109/ICSA.2018.00013.

GRANCHELLI, G., CARDARELLI, M., DI FRANCESCO, P., MALAVOLTA, I., IOVINO, L. y DI SALLE, A., 2017. Towards Recovering the Software Architecture of Microservice-Based Systems. 2017 IEEE International Software Architecture Workshops (ICSAW) [en línea]. S.l.: IEEE, pp. 46-53. ISBN 978-1-5090-4793-2. DOI 10.1109/ICSAW.2017.48.

KALSKE, M., MÄKITALO, N. y MIKKONEN, T., 2018. Challenges When Moving from Monolith to Microservice Architecture. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) [en línea]. S.l.: s.n., pp. 32-47. ISBN 9783319744322.

STRNNER, H., 2020. Microservice Decomposition Migration in the Automotive Industry. DOI 10.34726/hss.2020.81104

ADERALDO, C.M., MENDONCA, N.C., PAHL, C. y JAMSHIDI, P., 2017. Benchmark Requirements for Microservices Architecture Research. 2017 IEEE/ACM 1st International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering (ECASE) [en línea]. S.l.: IEEE, pp. 8-13. ISBN 978-1-5386-0417-5. DOI 10.1109/ECASE.2017.4.

TERZIĆ, B., DIMITRIESKI, V., KORDIĆ, S., MILOSAVLJEVIĆ, G. y LUKOVIĆ, I., 2018. Development and evaluation of MicroBuilder: a Model-Driven tool for the specification of REST Microservice Software Architectures. Enterprise Information

Systems [en línea], vol. 12, no. 8-9, pp. 1034-1057. ISSN 1751-7575. DOI 10.1080/17517575.2018.1460766.

Ghofrani, J., Bozorgmehr, A. (2019). Migration to Microservices: Barriers and Solutions. In: Florez, H., Leon, M., Diaz-Nafria, J., Belli, S. (eds) Applied Informatics. ICAI 2019. Communications in Computer and Information Science, vol 1051. Springer, Cham. DOI 10.1007/978-3-030-32475-9_20.

AL-DEBAGY, O. y MARTINEK, P., 2018. A Comparative Review of Microservices and Monolithic Architectures. 18th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2018 - Proceedings, pp. 149-154. DOI 10.1109/CINTI.2018.8928192.

LENARDUZZI, V., LOMIO, F., SAARIMÄKI, N. y TAIBI, D., 2019. Does Migrate a Monolithic System to Microservices Decrease the Technical Debt? Journal of Systems and Software [en línea], vol. 169, no. July. ISSN 01641212. DOI 10.1016/j.jss.2020.110710.

RAMIN, F., MATTHIES, C. y TEUSNER, R., 2020. More than Code. Proceedings of the IEEE/ACM 42nd Software Engineering Workshops [en línea]. New York, NY, USA: ACM, pp. 137-140. ISBN 9781450379632. DOI 10.1145/3387940.3392241.

NEELA, S., NEYYALA, Y., PENDEM, V., PERYALA, K. y KUMAR, V.V., 2021. Cloud Computing Based Learning Web Application Through Amazon Web Services. 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS) [en línea]. S.l.: IEEE, pp. 472-475. ISBN 978-1-6654-0520-1. DOI 10.1109/ICACCS51430.2021.9441974. Disponible en: <https://ieeexplore.ieee.org/document/9441974/>.

HU, T. y WANG, Y., 2021. A Kubernetes Autoscaler Based on Pod Replicas Prediction. 2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS) [en línea]. S.l.: IEEE, pp. 238-241. ISBN 978-1-6654-1538-5. DOI 10.1109/ACCTCS52002.2021.00053. Disponible en: <https://ieeexplore.ieee.org/document/9407757/>.

ANEXOS

ANEXO N° 01: Matriz de Consistencia

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLE	DIMENSIONES	INDICADOR
Principal	General	General	Independiente		
PG: ¿Cómo influye una arquitectura de microservicios en el proceso de ventas de la óptica Multi-len's?	Determinar la influencia de una arquitectura de microservicios en el proceso de ventas de la óptica Multi-len's	Una arquitectura de microservicios mejora el proceso de ventas de la óptica Multi-len's:	arquitectura de microservicios		
Secundarios	Específicos	Específicos	Dependiente		
PE1: ¿Cómo influye una arquitectura de microservicios en la tasa de adquisición para el proceso de ventas de la óptica Multi-len's?	Determinar la influencia de una arquitectura de microservicios en la tasa de adquisición del proceso de ventas de la óptica Multi-len's	Una arquitectura de microservicios mejora la tasa de adquisición en el proceso de ventas de la óptica Multi-len's.	Proceso de venta	Prospección	Tasa de Adquisición
PE2: ¿Cómo influye una arquitectura de microservicios en la tasa de conversión para el proceso de venta de la óptica Multi-len's?	Determinar la influencia de una arquitectura de microservicios en la tasa de conversión para el proceso de venta de la óptica Multi-len's	Una arquitectura de microservicios aumenta la tasa de conversión en la gestión de ventas de la óptica Multi-len's.		Prospección	Tasa de conversión

Fuente: Creación Propia

ANEXO N° 02: Tabla de Indicadores

DIMENSIÓN	INDICADOR	DESCRIPCIÓN	TÉCNICA	INSTRUMENTO	U. DE MEDIDA	FÓRMULA
Prospección	Tasa de Adquisición	Permite saber la cantidad de clientes nuevos en un lapso determinado de tiempo (Domínguez y Hermo, 2007, p.76).	Fichaje	Ficha de Registro	Proporción	$Tasa\ de\ adquisición = \frac{\text{número de clientes nuevos en el periodo}}{\text{número de clientes en el periodo}}$
Prospección	Tasa de conversión	La tasa de conversión se puede definir como el porcentaje de visitas a un sitio web de comercio electrónico que incluyen una transacción de compra. También se puede denominar como el porcentaje de visitantes del sitio web que se transforman en ventas potenciales al proporcionar sus datos de contacto al completar un formulario web. La tasa de conversión también se definió como el porcentaje del total de usuarios que compran un producto del total de visitas únicas en el sitio web. (Gabir, H.H. y Karrar, A.Z., 2018).	Fichaje	Ficha de Registro	Proporción	$Tasa\ de\ conversión = \frac{\text{Número de ventas}}{\text{Número total de visitas}}$

Fuente: Creación Propia

ANEXO N° 03: Operacionalización de la Variable

Variables	Definiciones Conceptuales	Definiciones Operacionales	Dimensión	Indicador	Instrumento	Medición
<p>INDEPENDIENTE</p> <p>Arquitectura de microservicios</p>	<p>Ortiz et al (2022) menciona que la arquitectura de microservicio se basa en el concepto de construir una aplicación como un conjunto de pequeños servicios interconectados, que se comunican a través de protocolos ligeros” (p. 1).</p>	<p>Se realizará una prueba de carga para medir el tiempo de respuesta y la disponibilidad del sistema, además de la prueba de resistencia para comprobar la tolerancia a fallas.</p>				Razón
<p>DEPENDIENTE</p> <p>Proceso de venta</p>	<p>Se entiende como una serie de pasos o etapas que es llevada a cabo por el vendedor al tratar con el cliente, y que como objetivo es lograr una venta efectiva del servicio o producto de la organización. (Carrasco, 2014, p. 15).</p>	<p>Se registran las ventas y los clientes en fichas de registros para medir los indicadores de la tasa de adquisición y la tasa de conversión respectivamente.</p>	<p>Prospección</p>	<p>Tasa de Adquisición (Domínguez y Hermo 2007)</p>	Ficha de Registro	
			<p>Prospección</p>	<p>Tasa de conversión (Domínguez y Hermo 2007)</p>		

Fuente: Creación Propia

ANEXO 04: Ficha de registro tasa de adquisición

Figura 11 Ficha de registro tasa de adquisición - Pretest

Ficha de Registro			
Investigador	Carranza Felix Christian	Tipo de prueba	Pre-Test
Empresa Investigada	Óptica Multi-len's	Indicador	Tasa de adquisición
Fórmula	$\text{Tasa de adquisición} = \frac{\text{número de clientes nuevos en el periodo}}{\text{número de clientes en el periodo}}$		

Ítem	Fecha	número de clientes nuevos en el periodo	número de clientes en el periodo	Tasa de Adquisición
1	22/08/2022	4	3	1.33
2	23/08/2022	4	2	2.00
3	24/08/2022	2	2	1.00
4	25/08/2022	2	2	1.00
5	26/08/2022	3	2	1.50
6	27/08/2022	4	3	1.33
7	28/08/2022	2	1	2.00
8	29/08/2022	2	3	0.67
9	30/08/2022	6	2	3.00
10	31/08/2022	2	3	0.67
11	1/09/2022	4	1	4.00
12	2/09/2022	5	2	2.50
13	3/09/2022	3	2	1.50
14	4/09/2022	4	1	4.00


 DNI N° 25572167
 Gerente General

Fuente: Creación Propia

Figura 12 Ficha de registro tasa de adquisición - PostTest

Ficha de Registro			
Investigador	Carranza Felix Christian	Tipo de prueba	Post -Test
Empresa Investigada	Óptica Multi-len's	Indicador	Tasa de adquisición
Fórmula	$\text{Tasa de adquisición} = \frac{\text{número de clientes nuevos en el periodo}}{\text{número de clientes en el periodo}}$		

Ítem	Fecha	número de clientes nuevos en el periodo	número de clientes en el periodo	Tasa de Adquisición
1	7/10/2022	4	2	2.00
2	8/10/2022	5	1	5.00
3	9/10/2022	3	2	1.50
4	10/10/2022	6	1	6.00
5	11/10/2022	5	2	2.50
6	12/10/2022	6	2	3.00
7	13/10/2022	3	1	3.00
8	14/10/2022	4	3	1.33
9	15/10/2022	6	2	3.00
10	16/10/2022	3	3	1.00
11	17/10/2022	3	1	3.00
12	18/10/2022	5	1	5.00
13	19/10/2022	4	2	2.00
14	20/10/2022	6	1	6.00



DNI N° 25572167
Gerente General

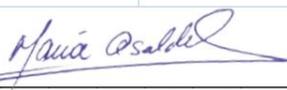
Fuente: Creación Propia

ANEXO 05: Ficha de registro tasa de conversión

Figura 13 Ficha de registro tasa de conversión – Pretest

Ficha de Registro				
Investigador	Carranza Felix Christian	Tipo de prueba	Pre-Test	
Empresa Investigada	Óptica Multi-len's	Indicador	Tasa de conversión	
Fórmula	$\text{Tasa de conversión} = \frac{\text{Número de clientes nuevos}}{\text{Peticiónes prospectivas recibidas}}$			

Ítem	fecha	Número de clientes nuevos	Peticiones prospectivas recibidas	Tasa de conversión
1	26/08/2022	4	7	0.57
2	27/08/2022	4	5	0.80
3	28/08/2022	2	9	0.22
4	29/08/2022	2	5	0.40
5	30/08/2022	3	5	0.60
6	31/08/2022	4	6	0.67
7	1/09/2022	2	7	0.29
8	2/09/2022	2	8	0.25
9	3/09/2022	4	4	1.00
10	4/09/2022	2	6	0.33
11	5/09/2022	4	8	0.50
12	6/09/2022	5	5	1.00
13	7/09/2022	3	5	0.60
14	8/09/2022	4	4	1.00



DNI N° 25572167
Gerente General

Fuente: Creación Propia

Figura 14 Ficha de registro tasa de conversión – PostTest

Ficha de Registro			
Investigador	Carranza Felix Christian	Tipo de prueba	Post-Test
Empresa Investigada	Óptica Multi-len's	Indicador	Tasa de conversión
Fórmula	$\text{Tasa de conversión} = \frac{\text{Número de clientes nuevos}}{\text{Peticiónes prospectivas recibidas}}$		

Ítem	fecha	Número de clientes nuevos	Peticiónes prospectivas recibidas	Tasa de conversión
1	7/10/2022	6	9	0.67
2	8/10/2022	5	7	0.71
3	9/10/2022	4	5	0.80
4	10/10/2022	6	7	0.86
5	11/10/2022	5	5	1.00
6	12/10/2022	4	6	0.67
7	13/10/2022	4	5	0.80
8	14/10/2022	5	5	1.00
9	15/10/2022	6	7	0.86
10	16/10/2022	4	6	0.67
11	17/10/2022	6	8	0.75
12	18/10/2022	7	8	0.88
13	19/10/2022	6	6	1.00
14	20/10/2022	5	5	1.00



 DNI N° 25572167
 Gerente General

Fuente: Creación Propia

ANEXO 06: Autorización para la realización y difusión de resultados de la investigación

Figura 15 Autorización para la realización y difusión de resultados de la investigación

Anexo 6: Autorización para la realización y difusión de resultados de la investigación

AUTORIZACIÓN PARA LA REALIZACIÓN Y DIFUSIÓN DE RESULTADOS DE LA INVESTIGACIÓN

Por medio del presente documento, Yo María Isabel, Asalde Roque identificado con DNI N° 25572167 y representante legal de Óptica Multi-len's autorizo a Christian Bernardo Carranza Felix identificado con DNI N° 76420854 a realizar la investigación titulada: "Arquitectura de microservicios para el proceso de ventas de la óptica Multi-len's, 2022." y a difundir los resultados de la investigación utilizando el nombre de Óptica Multi-len's

Lima, 25 de octubre de 2022



María Isabel, Asalde Roque

DNI N° 25572167

Gerente General

Óptica Multi-len's

ANEXO 07: Consentimiento informado

Figura 16 Consentimiento informado

Anexo 7: Consentimiento informado

CONSENTIMIENTO INFORMADO

Yo María Isabel, Asalde Roque identificado con DNI N.º 25572167 he sido informada sobre el procedimiento de la investigación titulada " Arquitectura de microservicios para el proceso de ventas de la óptica Multi-len's, 2022.", cuyo autor es Carranza Felix Christian con DNI N° 76420854 y se me ha entregado una copia de este consentimiento informado, fechado y firmado.

Además, se me ha explicado las características y el objetivo del estudio, así como los posibles beneficios de este. He contado con el tiempo y la oportunidad para realizar preguntas y plantear las dudas que poseía. Todas las preguntas fueron respondidas a mi entera satisfacción.

Se me ha asegurado que se mantendrá la confidencialidad de mis datos. Mi consentimiento lo otorgo de manera voluntaria y sé que soy libre de retirarme del estudio en cualquier momento, por cualquier razón de fuerza mayor. Por lo tanto, en forma consciente y voluntaria doy mi consentimiento para ser parte de esta investigación.

Lima, 2022

Asalde Roque, María Isabel		
Apellidos y nombres	Firma	Huella
25572167	52	F
DNI	Edad	(F:Femenino / M:Masculino)

ANEXO 08: Desarrollo de la metodología

A través de entrevistas realizadas a los dueños del negocio, y la revisión de documentos, se establecieron los requerimientos no funcionales y funcionales que el sistema debe cumplir.

A continuación, se muestra el Product Backlog que determina los requerimientos necesarios para cada entregable.

Tabla 10 Product backlog

Product backlog		
N°	Requerimiento	Descripción
1	Visualizar categoría de productos	El sistema debe permitir mostrar el listado de categorías del producto registrado.
2	Visualizar productos	El sistema permitirá ver el listado de productos registrados.
3	Visualizar precios	El sistema debe mostrar los precios de los productos.
4	Registrar pedidos	El sistema debe permitir registrar las compras de los clientes
5	Registrar visitas al sistema	El sistema debe permitir registrar la cantidad de clientes que visitan la página.
6	Registrarse al sistema	El sistema debe permitir registrarse a los clientes para realizar las compras
7	Tasa de conversión	El sistema debe tener la opción de calcular la tasa de conversión.
8	Tasa de adquisición	El sistema debe tener la opción de calcular la tasa de adquisición.
9	Registro de productos	El sistema debe permitir registrar nuevos productos
10	Actualización de inventario	El sistema debe permitir actualizar el stock de cada producto según la compra del cliente

11	Registro de categoría de productos	El sistema debe permitir registrar los productos por categoría
12	Registro de subcategorías	El sistema debe permitir registrar las subcategorías de cada producto.
13	Generar carrito de compras	El sistema debe permitir crear carrito de compras para que el cliente pueda almacenar los productos que quiera comprar.
14	Generar pedido	El sistema debe permitir realizar la compra de un producto.

Fuente: Creación Propia

Historias de usuario

Tabla 11 Historia de Usuario 1

Historia de Usuario	
Numero: 1	Nombre: Visualizar categoría de productos
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Bajo (Alto/Media/Baja)
Iteración Asignada	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir mostrar el listado de categorías del producto registrados.	
Observaciones:	

Fuente: Creación Propia

Tabla 12 Historia de Usuario 2

Historia de Usuario	
Numero: 2	Nombre: Visualizar subcategoría de productos
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Baja (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir mostrar el listado de subcategorías de la categoría registrada.	
Observaciones:	

Fuente: Creación Propia

Tabla 13 Historia de Usuario 3

Historia de Usuario	
Numero: 3	Nombre: Visualizar productos
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Bajo (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema permitirá ver el listado de productos registrados.	
Observaciones:	

Fuente: Creación Propia

Tabla 14 Historia de Usuario 4

Historia de Usuario	
Numero: 4	Nombre: Visualizar precios
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Bajo (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe mostrar los precios de los productos.	
Observaciones:	

Fuente: Creación Propia

Tabla 15 Historia de Usuario 5

Historia de Usuario	
Numero: 5	Nombre: Registrar pedidos
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Alto (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir registrar las compras de los clientes	
Observaciones:	

Fuente: Creación Propia

Tabla 16 Historia de Usuario 6

Historia de Usuario	
Numero: 6	Nombre: Registrar visitas al sistema
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Media (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir registrar la cantidad de clientes que visitan la página.	
Observaciones:	

Fuente: Creación Propia

Tabla 17 Historia de Usuario 7

Historia de Usuario	
Numero: 7	Nombre: Registrarse al sistema
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Alta (Alto/Media/Baja)
Iteración Asignad:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir registrarse a los clientes para realizar las compras.	
Observaciones:	

Fuente: Creación Propia

Tabla 18 Historia de Usuario 8

Historia de Usuario	
Numero: 8	Nombre: Tasa de conversión
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Alta (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe tener la opción de calcular la tasa de conversión.	
Observaciones:	

Fuente: Creación Propia

Tabla 19 Historia de Usuario 9

Historia de Usuario	
Numero: 9	Nombre: Tasa de adquisición
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Alta (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe tener la opción de calcular la tasa de adquisición.	
Observaciones:	

Fuente: Creación Propia

Tabla 20 Historia de Usuario 10

Historia de Usuario	
Numero: 10	Nombre: Registro de productos
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Alto (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir registrar nuevos productos	
Observaciones:	

Fuente: Creación Propia

Tabla 21 Historia de Usuario 11

Historia de Usuario	
Numero: 11	Nombre: Actualización de inventario
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Baja (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir actualizar el stock de cada producto según la compra del cliente.	
Observaciones:	

Fuente: Creación Propia

Tabla 22 Historia de Usuario 12

Historia de Usuario	
Numero: 12	Nombre: Registro de categoría de productos
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Alto (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir registrar los productos por categoría.	
Observaciones:	

Fuente: Creación Propia

Tabla 23 Historia de Usuario 13

Historia de Usuario	
Numero: 13	Nombre: Registro de subcategorías
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Baja (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir registrar las subcategorías de cada producto.	
Observaciones:	

Fuente: Creación Propia

Tabla 24 Historia de Usuario 14

Historia de Usuario	
Numero: 14	Nombre: Generar carrito de compras
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Media (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir crear carrito de compras para que el cliente pueda almacenar los productos que quiera comprar.	
Observaciones:	

Fuente: Creación Propia

Tabla 25 Historia de Usuario 15

Historia de Usuario	
Numero: 15	Nombre: Generar pedido
Usuario: Administrador	
Prioridad en Negocio: Alta (Alto/Media/Baja)	Riesgo en Desarrollo: Alta (Alto/Media/Baja)
Iteración Asignada:	Puntos Estimados:
Programador Responsable: Carranza Christian	
Descripción: El sistema debe permitir realizar la compra de un producto.	
Observaciones:	

Fuente: Creación Propia

Sprint backlog

En esta sección se procedió a definir los sprints que se llevaran a cabo, los cuales están conformados por historias de usuarios que deberán ser desarrollados en el periodo determinado.

Tabla 26 Sprint backlog

Sprint backlog			
N* Sprint	Historia de Usuario	Descripción	Estimación (# días)
0	-	Se realizará la arquitectura del sistema, además se llevará a cabo el diseño de base de datos junto con la descripción de los microservicios de negocio	20
1	H6, H9, H11, H12	Se llevará a cabo el desarrollo del mantenimiento de categorías, subcategorías, productos y además del registro de usuarios del sistema.	17
2	H1, H2, H3, H5	Se llevará a cabo el desarrollo de la vista del listado de categorías, productos con sus respectivos precios y la visualización de la cantidad de visitantes a la página.	12
3	H4, H10, H13, H14, H15	Se llevará a cabo el desarrollo del mantenimiento de carrito de compras, el registro de las ventas, la finalización de la compra.	18

4	H7, H8	Se llevará a cabo el desarrollo del cálculo y la visualización de la tasa de adquisición y la tasa de conversión.	4
5	-	Se llevará acabo la contenerización de los microservicios para el despliegue.	12

Fuente: Creación Propia

Tabla 27 Actividades

N° SPRINT	HISTORIA DE USUARIO	ACTIVIDADES	ESTIMACIÓN (#DÍAS)	TOTAL (#DÍAS)
0		Reunión de planificación	1	20
		Definición de patrones a usar	3	
		Definición de la arquitectura del software	7	
		Descripción de los microservicios y la creación de la base de datos	8	
		Reunión de sprint	1	
1	Registrarse al sistema	Reunión de planificación	1	17
		Desarrollar UI	1	
		Desarrollo del API	1	

		Pruebas Unitarias	1			
		Pruebas funcionales	1			
		Reunión de planificación	1			
	Registro de categoría de productos	Desarrollar UI	2			
		Desarrollo del API	1			
		Pruebas Unitarias	1			
		Pruebas funcionales	1			
		Reunión de planificación	1			
	Registro de subcategorías	Desarrollar UI	2			
		Desarrollo del API	1			
		Pruebas Unitarias	1			
		Pruebas funcionales	1			
	2		Reunión de planificación		1	12
		Visualizar categoría de productos	Desarrollar UI		2	
Desarrollo del API			1			
Pruebas Unitarias			1			
Pruebas funcionales			1			
		Reunión de planificación	1			

		Desarrollar UI	2	
	Visualizar productos	Desarrollo del API	1	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
		Reunión de planificación	1	
		Desarrollar UI	2	
	Visualizar precios	Desarrollo del API	1	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
		Reunión de planificación	1	
		Desarrollar UI	2	
	Registrar visitas al sistema	Desarrollo del API	1	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
3		Reunión de planificación	1	18
		Desarrollar UI	2	
	Registrar pedidos	Desarrollo del API	1	
		Pruebas Unitarias	1	

		Pruebas funcionales	1	
		Reunión de planificación	1	
	Actualización de inventario	Desarrollar UI	2	
		Desarrollo del API	1	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
		Reunión de planificación	1	
	Generar carrito de compras	Desarrollar UI	2	
		Desarrollo del API	1	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
		Reunión de planificación	1	
	Generar pedido	Desarrollar UI	2	
		Desarrollo del API	1	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
4		Reunión de planificación	1	4
	Tasa de conversión	Desarrollo del API	1	

		Pruebas Unitarias	1	
		Pruebas funcionales	1	
		Reunión de planificación	1	
	Tasa de adquisición	Desarrollo del API	1	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
5	Crear archivos de despliegue y configuración de Clúster de Kubernetes	Reunión de planificación	1	12
		Crear los archivos Dockerfile	4	
		Crear los archivos yml	5	
		Configuración de Clúster de Kubernetes	2	

Fuente: Creación Propia

Sprint 0

Tabla 28 Sprint 0

N° SPRINT	HISTORIA DE USUARIO	ACTIVIDADES	ESTIMACIÓN (#DÍAS)	TOTAL (#DÍAS)
0		Reunión de planificación	1	20
		Definición de patrones a usar	3	
		Definición de la arquitectura del software	7	

		Descripción de los microservicios y la creación de la base de datos	8	
		Reunión de sprint	1	

Fuente: Creación Propia

○ **Reunión de planificación**

Esta reunión se realizó con el fin de planificar las tareas que se realizaran en cada entregable del producto.

Tabla 29 Acta de reunión

ACTA DE REUNIÓN	
Comité o Grupo: Equipo Scrum	Acta No: 1
Citada por: -	Fecha: 10-07-2022
Coordinador: Carranza Felix Christian	Hora inicio: 5:00 pm Fin: 8:30 pm
Secretario: -	Lugar: Google Meet.

Fuente: Creación Propia

Tabla 30 Desarrollo de la reunión

DESARROLLO DE LA REUNIÓN	
1.	<p>El dueño del negocio detallo los puntos importantes para el cumplimiento de los entregables y el control que se va a llevar de cada uno de ellos.</p> <p>El gerente resaltó que es importante tener una comunicación constante con su persona para mantenerlo al tanto de cada avance y ver el estado de los entregables, de tal forma que pueda dar un seguimiento. De esta manera, cualquier duda que se tenga con respecto a la lógica del negocio también será resuelto por el gerente.</p>
2.	Se detalló las tareas a realizar para desarrollar los entregables definidos del Sprint 0.
3.	
Observaciones:	

Fuente: Creación Propia

Tabla 31 Conclusiones

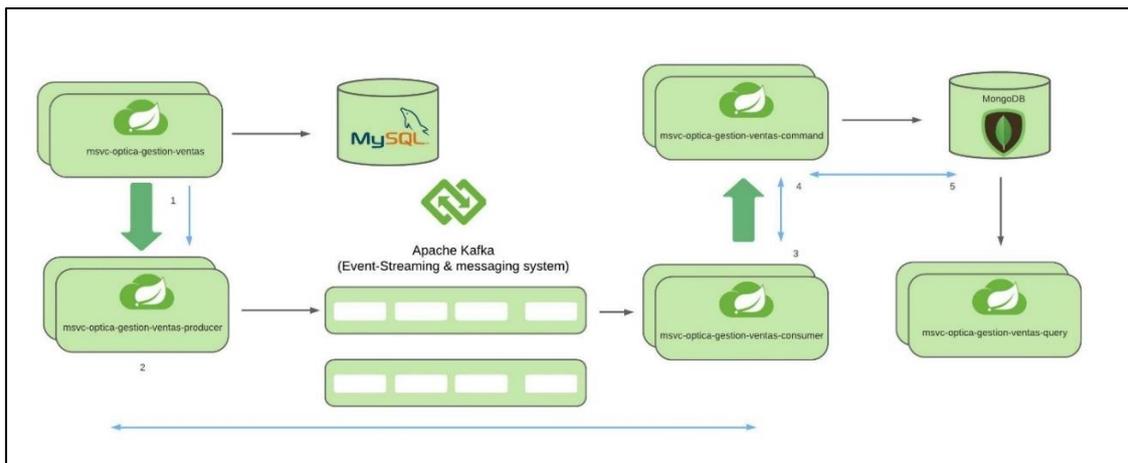
CONCLUSIONES			
No	Tarea	Período de cumplimiento	Observaciones
1	Definición de patrones a usar	3 días	
2	Definición de la arquitectura del software	7 días	
3	Descripción de los microservicios y la creación de la base de datos	8 días	

Fuente: Creación Propia

- **Definición de patrones a usar**

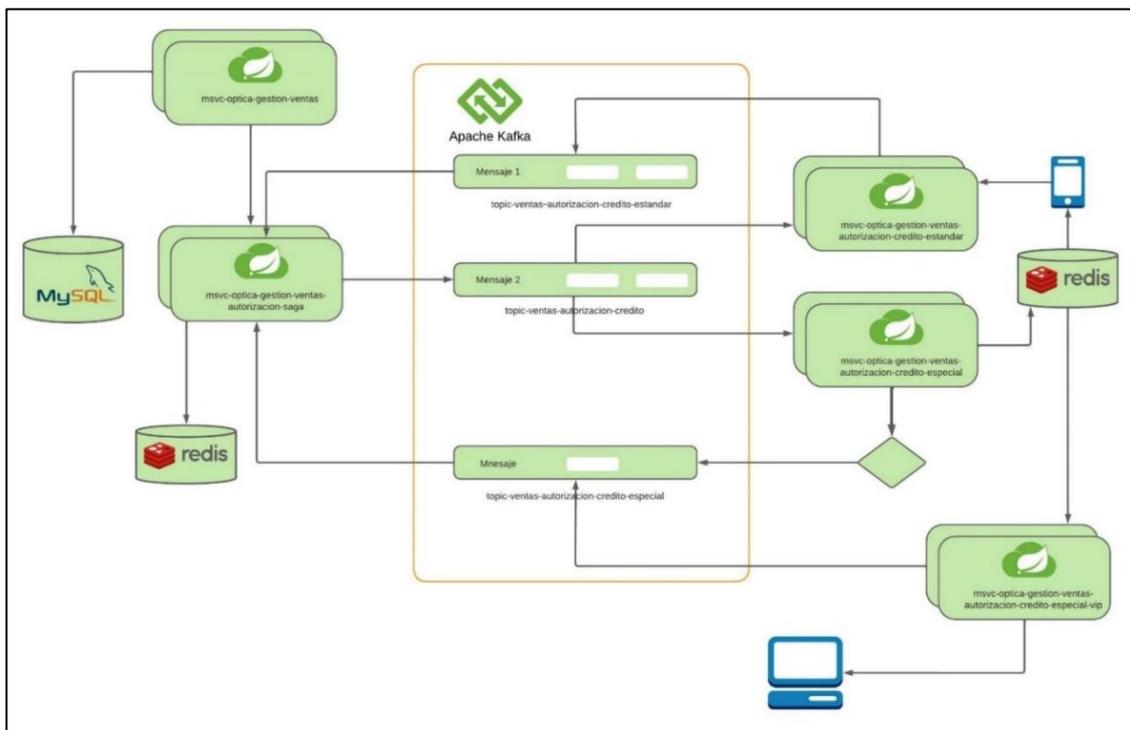
Para este proyecto se emplearon varios patrones a fin de dar un nivel de madurez mayor, además de que son necesario para ser considerado una arquitectura de microservicios para lo cual se está empleando los principales patrones de diseño de software como el CQRS y SAGA PATTERN, además del patrón de base de datos por servicio.

Figura 17 Patrón CQRS en el proyecto



Fuente: Creación Propia

Figura 18 Patrón SAGA en el proyecto

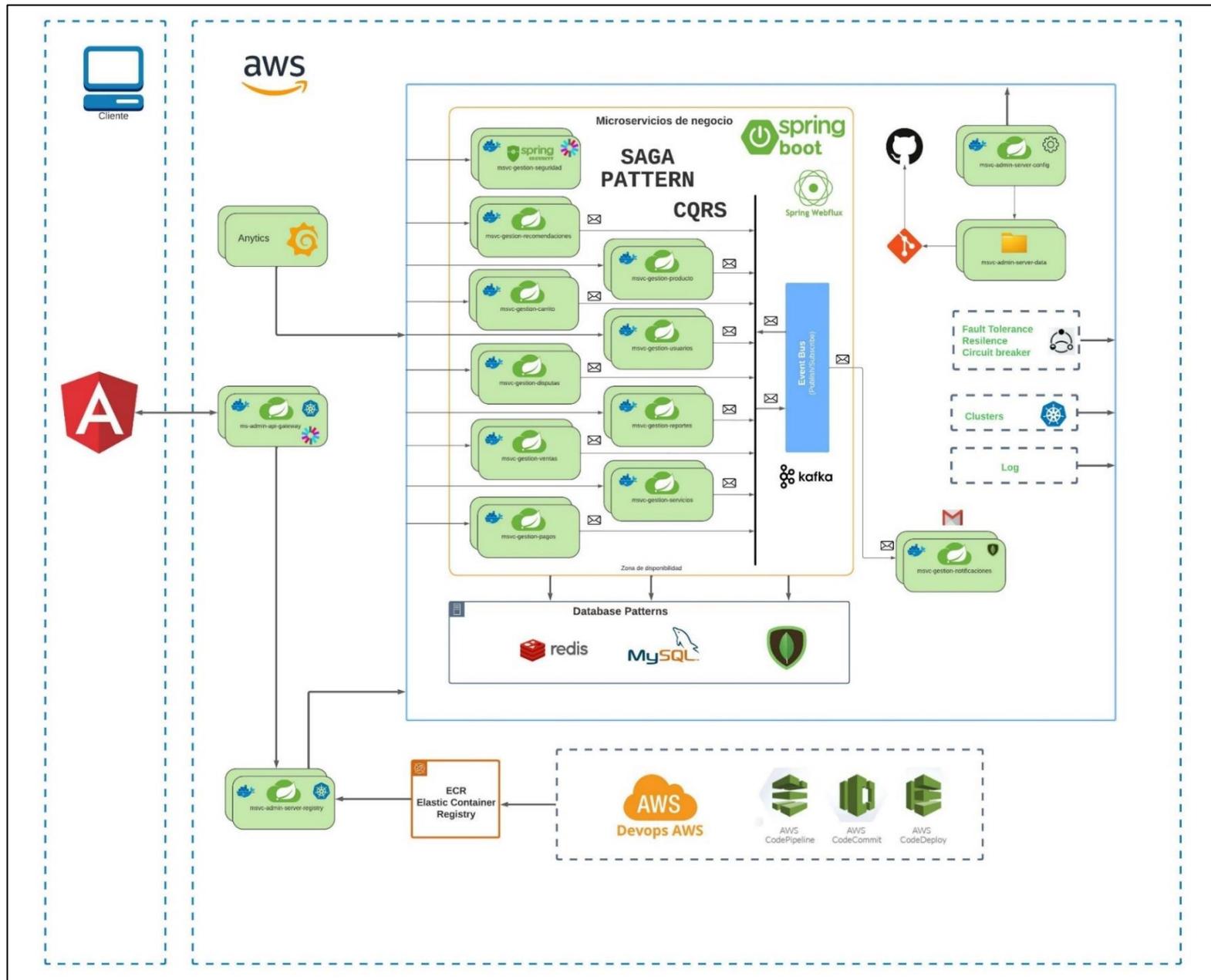


Fuente: Creación Propia

- **Definición de la arquitectura del software**

En la siguiente figura se muestra el diseño de la arquitectura, en donde se observan las relaciones de cada componente. Asimismo, se puede apreciar que dichos componentes están contenidos en clúster de Kubernetes, el cual ayudara en la automatización del despliegue de la aplicación como del escalado de los contenedores.

Figura 19 Diseño de la arquitectura de microservicios del proyecto



Fuente: Creación Propia

- **Descripción de los microservicios y creación de la base de datos**

Microservicios de negocio

Como se pudo observar en la figura anterior se identificaron los microservicios que componen el sistema. A continuación, se detallará las funciones de cada uno de los microservicios de negocio desarrollados en esta investigación, ya que por el tiempo corto no se podrá desarrollar todos los microservicios planteados en la arquitectura. Cabe mencionar que como parte de la arquitectura se ha decidido trabajar con el patrón de base de datos por servicio, esto con el fin de tener el bajo acoplamiento del sistema.

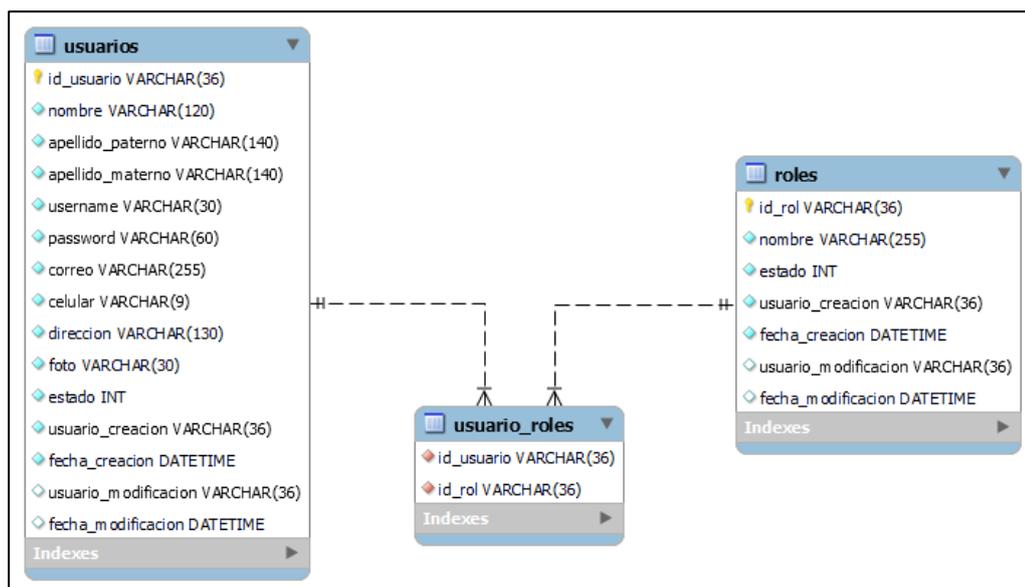
Promociones

Este microservicio permitirá gestionar las promociones que luego serán asignadas a los productos, además se encarga de mostrar los productos que tengan una fecha de promoción por terminar el inicio de la página.

Usuarios

Este microservicio permitirá realizar la gestión de los clientes y los administradores del sistema. A continuación, se presenta el diagrama de base de datos que pertenece al microservicio Usuarios.

Figura 20 Diagrama de base de datos del microservicio Usuarios.

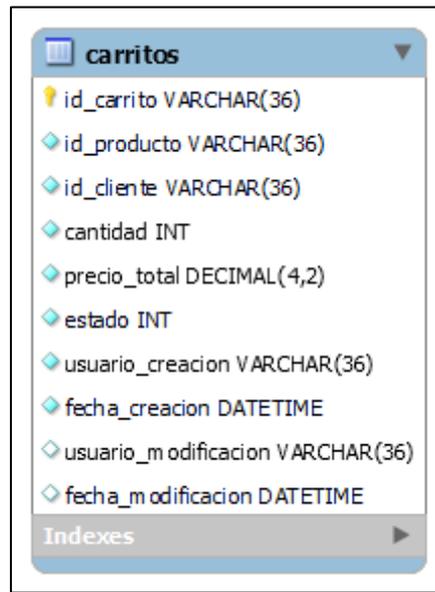


Fuente: Creación Propia

Carrito

Este microservicio permitirá realizar la gestión de los productos que seleccionará el cliente para su compra. A continuación, se presenta el diagrama de base de datos que pertenece al microservicio carrito.

Figura 21 Diagrama de base de datos del microservicio carritos.



Fuente: Creación Propia

Reportes

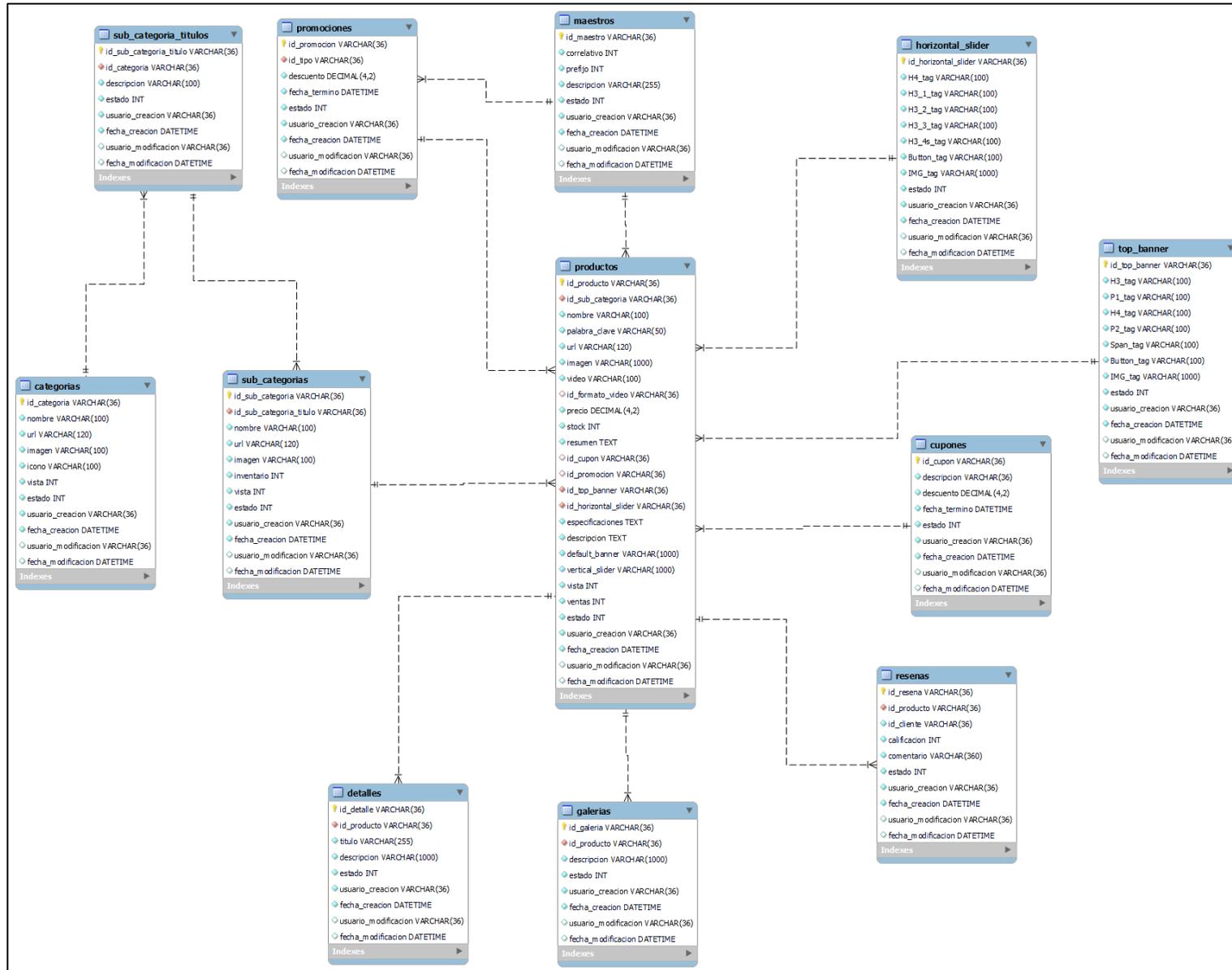
Este microservicio permitirá generar reporte de las distintas informaciones que se maneja en el sistema.

Productos

Este microservicio permitirá realizar la gestión de los productos con todos sus detalles, las categorías y las subcategorías a las que pertenecen, asimismo la galería de imágenes de cada producto como también de las disputas de los usuarios sobre el producto

A continuación, se presenta el diagrama de base de datos que pertenece al microservicio productos.

Figura 22 Diagrama de base de datos del microservicio productos

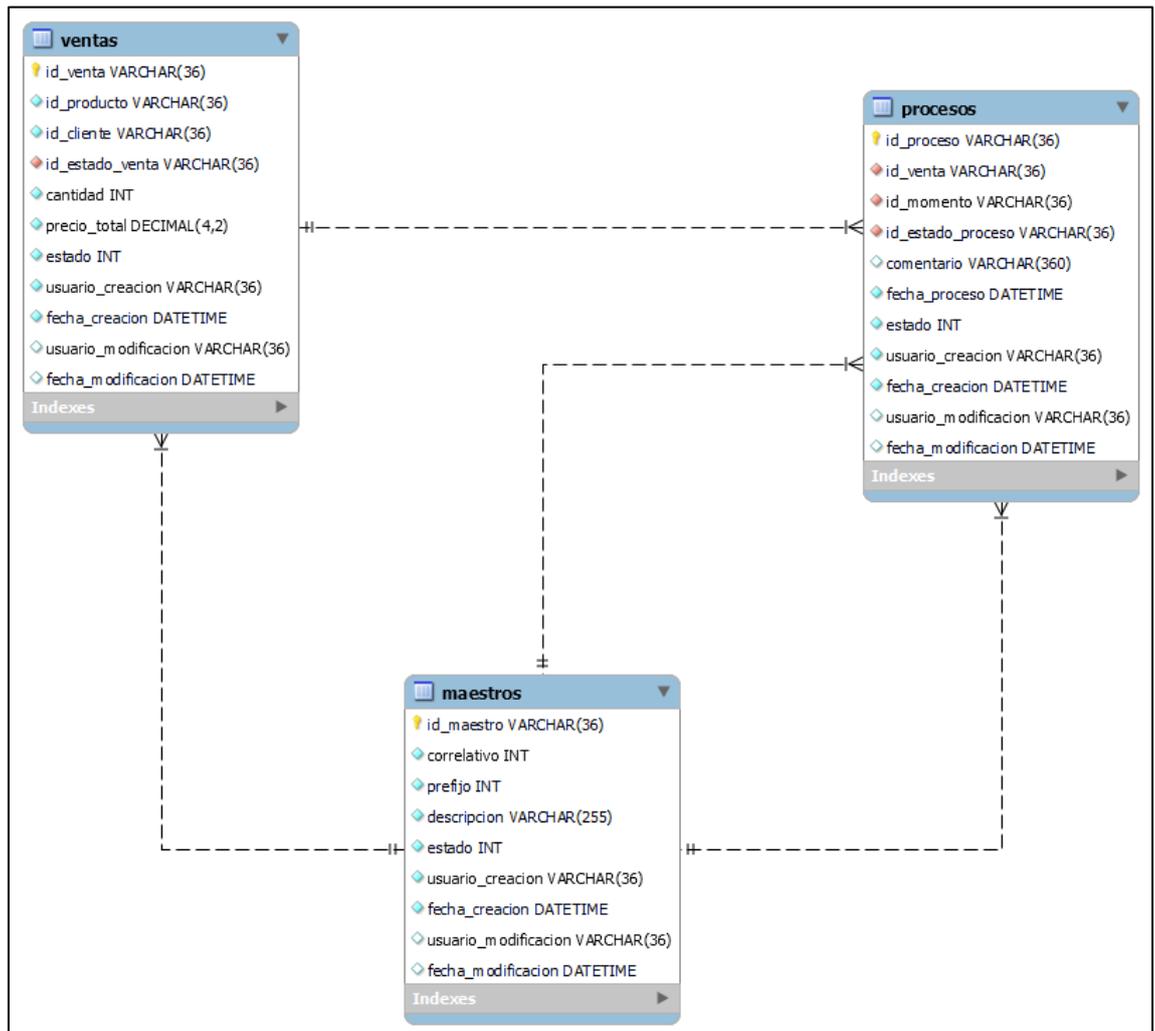


Fuente: Creación Propia

Ventas

Este microservicio permitirá realizar la gestión de la compra de los productos. A continuación, se presenta el diagrama de base de datos que pertenece al microservicio ventas.

Figura 23 Diagrama de base de datos del microservicio ventas



Fuente: Creación Propia

Recomendaciones

Este microservicio permitirá mostrar al cliente las recomendaciones según el producto más vendido o productos similares a sus compras anteriores.

Pagos

Este microservicio permitirá realizar la gestión de los pagos que realiza el cliente, además de los distintos métodos de pago que podría tener.

- **Revisión del sprint**

Tabla 32 Revisión del sprint

N°	ACTIVIDADES	DURACIÓN (#DÍAS)	ESTADO	OBSERVACIONES	DURACIÓN FINAL (#DÍAS)
0	Reunión de planificación	1	Completado		1
	Definición de patrones a usar	3	Completado		3
	Definición de la arquitectura del software	7	Completado		7
	Creación de la base de datos	8	Completado		8
	Revisión de sprint	1	Completado		1

Fuente: Creación Propia

Sprint 1

Tabla 33 Sprint 1

N° SPRINT	HISTORIA DE USUARIO	ACTIVIDADES	ESTIMACIÓN (#DÍAS)	TOTAL (#DÍAS)
1	Registrarse al sistema	Reunión de planificación	1	12
		Desarrollar UI	2	
		Desarrollo del API	1	
		Pruebas funcionales	1	
	Registro de categoría de productos	Reunión de planificación	1	
		Desarrollar UI	1	
		Desarrollo del API	2	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
	Registro de subcategorías	Reunión de planificación	1	
		Desarrollar UI	2	
		Desarrollo del API	1	

	Pruebas Unitarias	1	
	Pruebas funcionales	1	

Fuente: Creación Propia

○ **Reunión de planificación**

Se llevo a cabo la reunión con todo el equipo para el desarrollo del Sprint 1, a fin de planificar todas las actividades para completar los entregables.

✓ **Acta de reunión**

Tabla 34 Acta de reunión sprint 1

ACTA DE REUNIÓN	
Comité o Grupo: Equipo Scrum	Acta No: 1
Citada por: Carranza Felix Christian	Fecha: 01-08-2022
Coordinador: Carranza Felix Christian	Hora inicio: 10:00 am Fin: 11:30 am
Secretario: -	Lugar: Google Meet.

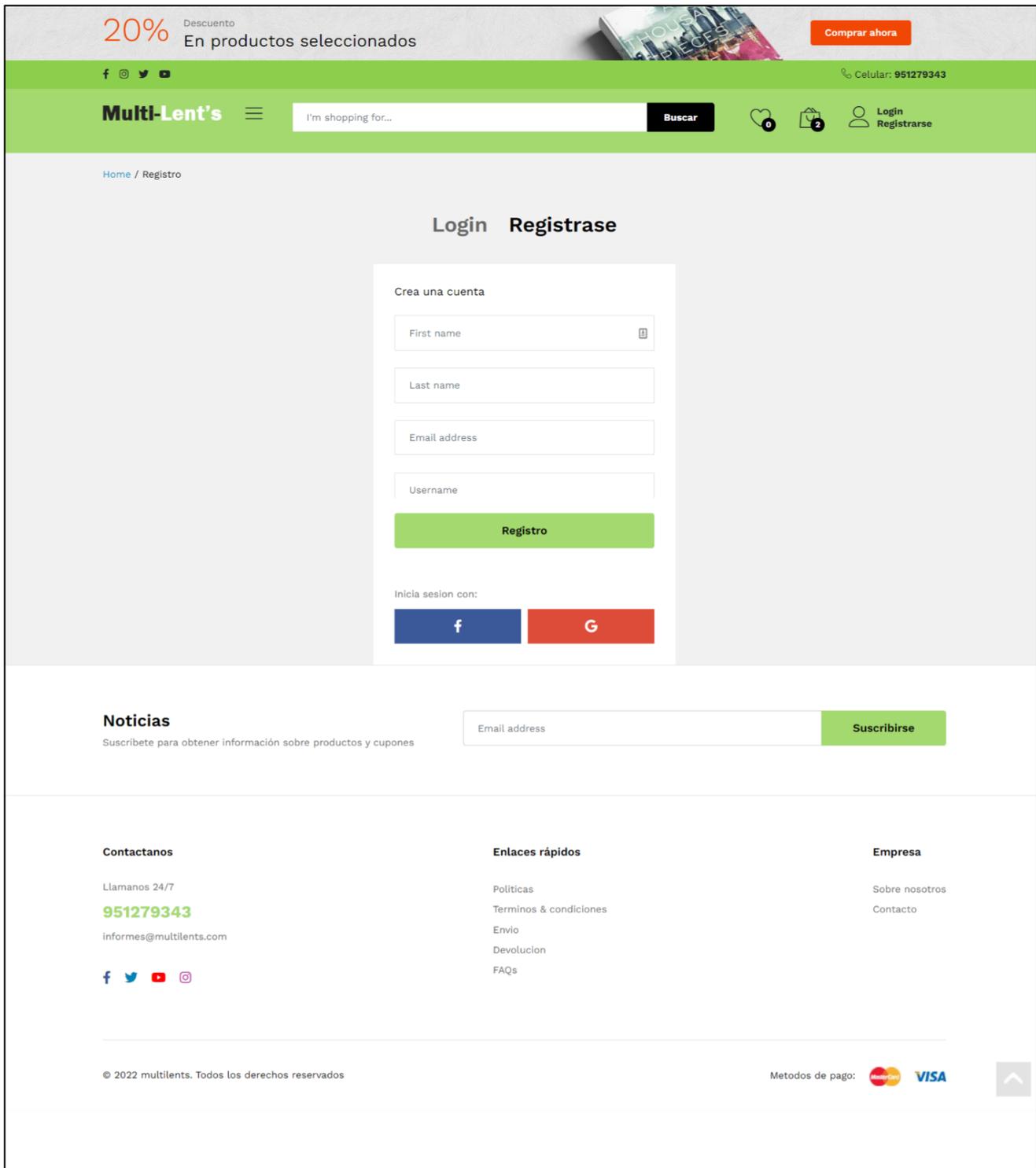
Fuente: Creación Propia

1) REGISTRARSE AL SISTEMA

✓ **Desarrollo UI**

Esta opción solo estará disponible para el cliente.

Figura 24 Desarrollo UI de registro de usuarios



Fuente: Creación Propia

✓ Desarrollo del API

A continuación, se muestra el desarrollo del API para el registro de usuarios al sistema.

Controller

Figura 25 Controller API registrar usuario

```
@PostMapping
public ResponseEntity<CustomResponse> save(@Valid @RequestBody UsuarioDTO usuarioDTO){
    UsuarioDTO usuario = this.usuarioService.save(usuarioDTO);
    CustomResponse r = new CustomResponse(String.valueOf(CodeEnum.SUCCESS),
        usuario, message: "Usuario registrado correctamente.");
    return new ResponseEntity<>(r, HttpStatus.CREATED);
}
```

Fuente: Creación Propia

ServiceImpl

Figura 26 Service API registrar usuario

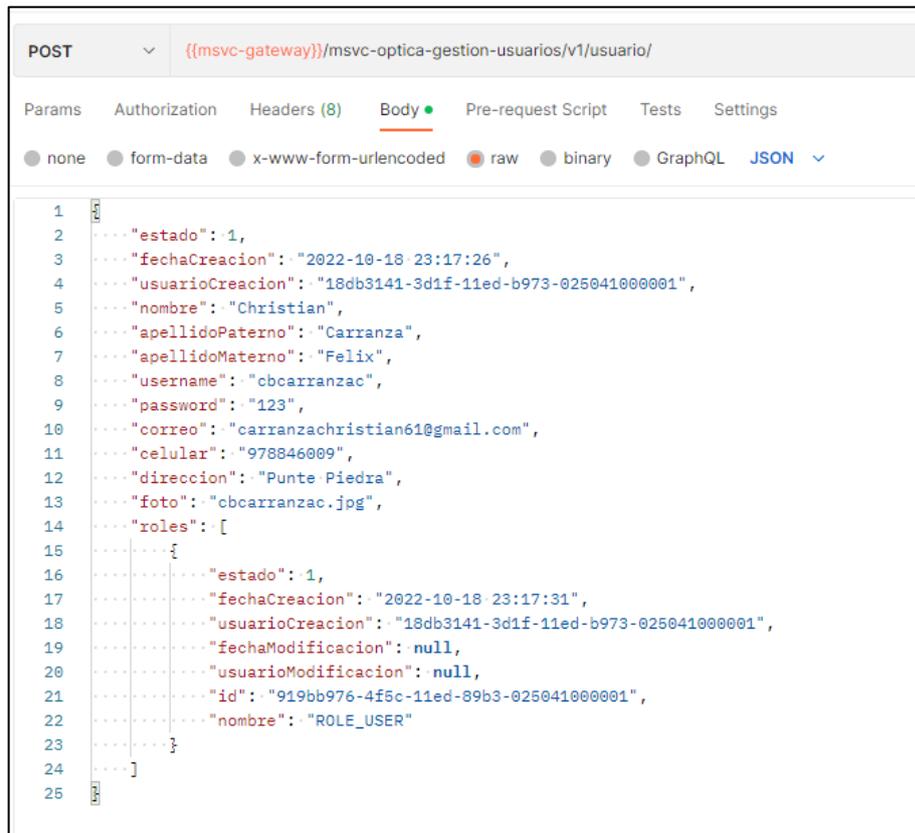
```
@Override
public UsuarioDTO save(UsuarioDTO usuarioDTO) {
    var usuarioEntity :UsuarioEntity = usuarioMapper.postDtoToEntity(usuarioDTO);
    var save :UsuarioEntity = this.usuarioRepository.save(usuarioEntity);
    return usuarioMapper.entityToGetDto(save);
}
```

Fuente: Creación Propia

URL: {{msvc-gateway}}/msvc-optica-gestion-usuarios/v1/usuario/

Descripción: Permite la creación de usuarios del sistema

Figura 27 Prueba del endpoint registrar usuario



```
POST {{msvc-gateway}}/msvc-optica-gestion-usuarios/v1/usuario/

Params Authorization Headers (8) Body Pre-request Script Tests Settings
● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON

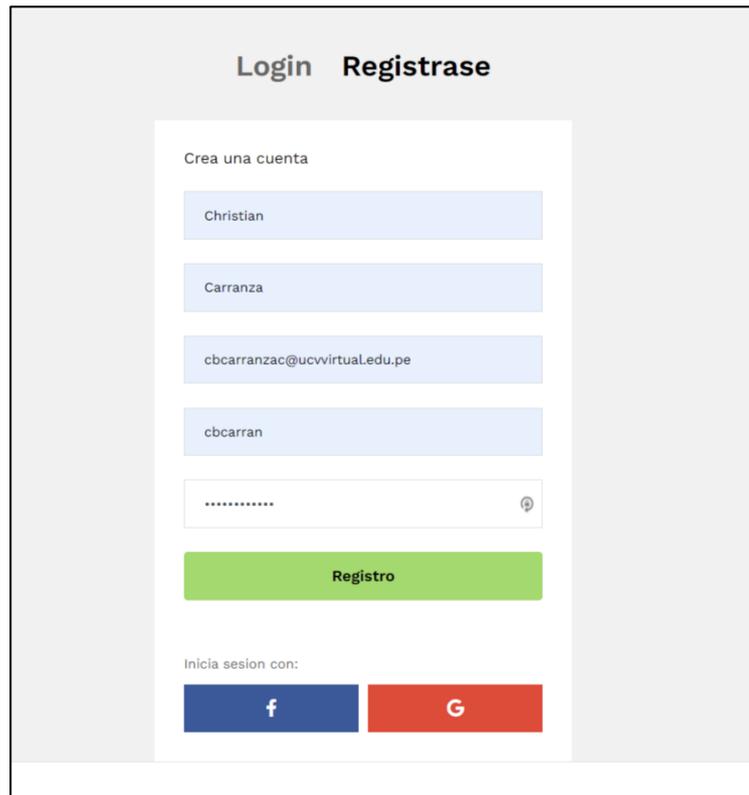
1 [
2   .... "estado": 1,
3   .... "fechaCreacion": "2022-10-18 23:17:26",
4   .... "usuarioCreacion": "18db3141-3d1f-11ed-b973-025041000001",
5   .... "nombre": "Christian",
6   .... "apellidoPaterno": "Carranza",
7   .... "apellidoMaterno": "Felix",
8   .... "username": "cbcarranzac",
9   .... "password": "123",
10  .... "correo": "carranzachristian61@gmail.com",
11  .... "celular": "978846009",
12  .... "direccion": "Punte Piedra",
13  .... "foto": "cbcarranzac.jpg",
14  .... "roles": [
15  ..... {
16  .....   "estado": 1,
17  .....   "fechaCreacion": "2022-10-18 23:17:31",
18  .....   "usuarioCreacion": "18db3141-3d1f-11ed-b973-025041000001",
19  .....   "fechaModificacion": null,
20  .....   "usuarioModificacion": null,
21  .....   "id": "919bb976-4f5c-11ed-89b3-025041000001",
22  .....   "nombre": "ROLE_USER"
23  ..... }
24  .... ]
25 ]
```

Fuente: Creación Propia

✓ Pruebas funcionales

Para la prueba de la funcionalidad se debe llenar todos los campos como el nombre, apellido, correo, username y la contraseña

Figura 28 Prueba funcional de registrar usuario

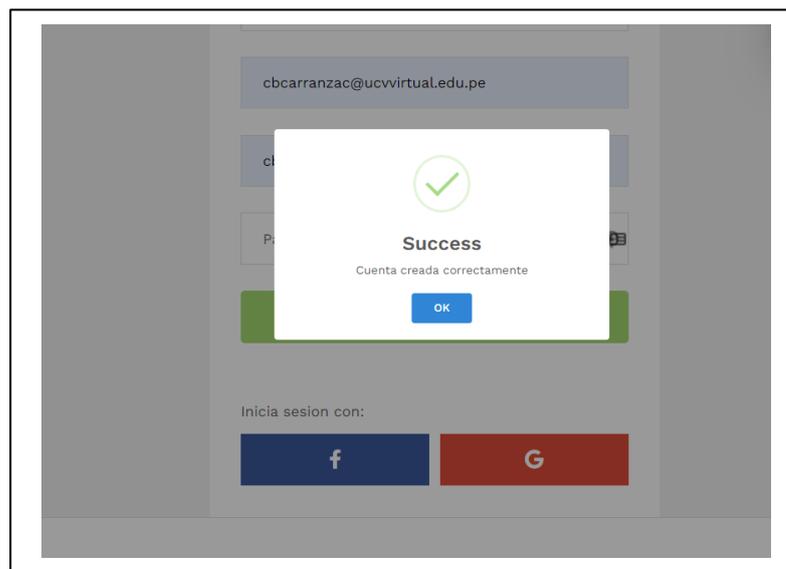


The screenshot shows a registration form titled "Login Registrase". Under the "Crea una cuenta" section, there are five input fields: a name field containing "Christian", a last name field containing "Carranza", an email field containing "cbcarranzac@ucvvirtual.edu.pe", a username field containing "cbcarran", and a password field with masked characters ".....". Below these fields is a green "Registro" button. At the bottom, there is a section "Inicia sesion con:" with two buttons: a blue Facebook "f" button and a red Google "G" button.

Fuente: Creación Propia

Cuando se registre correctamente se mostrará el siguiente mensaje de confirmación

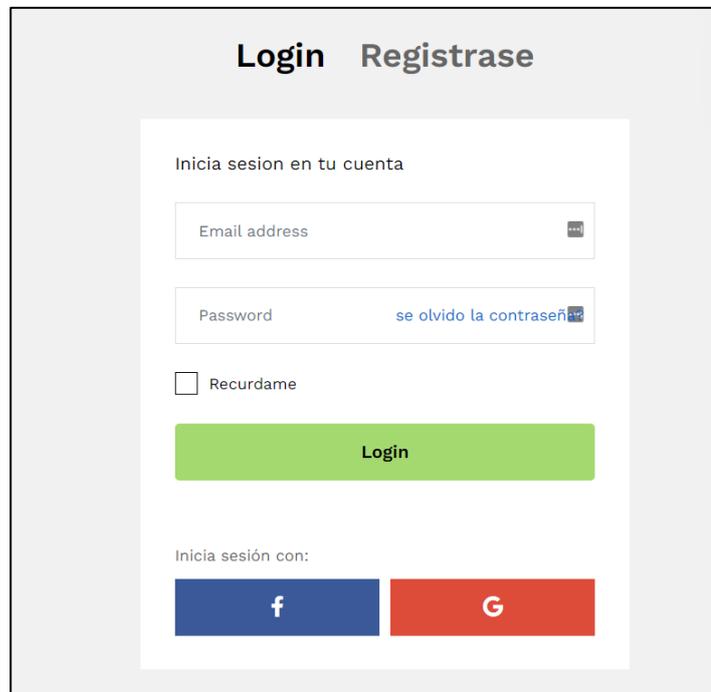
Figura 29 Mensaje de confirmación de registro usuario



Fuente: Creación Propia

Luego de que termine la confirmación se redirigirá a la opción del inicio de sesión.

Figura 30 login cliente



The image shows a login interface with the following elements:

- Header: **Login Registrase**
- Section: **Inicia sesion en tu cuenta**
- Form fields:
 - Email address (with a clear icon)
 - Password (with a link: [se olvido la contraseña](#))
- Checkbox: **Recurdame**
- Button: **Login** (green)
- Section: **Inicia sesión con:**
- Buttons: **f** (Facebook) and **G** (Google)

Fuente: Creación Propia

2) REGISTRO DE CATEGORÍA DE PRODUCTOS

✓ Desarrollo UI

Esta opción solo estará disponible para el administrador.

Figura 31 UI nueva categoría

Fuente: Creación Propia

✓ Desarrollo del API

A continuación, se muestra el desarrollo del API para el registro de categorías del producto.

Controller

Figura 32 Controller API registrar categoría

```
@PostMapping
public ResponseEntity<CustomResponse> save(@Valid @RequestBody CategoriaDTO categoriaDTO){
    CategoriaDTO categoria = this.categoriaService.save(categoriaDTO);
    CustomResponse r = new CustomResponse(String.valueOf(CodeEnum.SUCCESS), categoria,
        message: "Categoria registrado correctamente.");
    return new ResponseEntity<>(r, HttpStatus.CREATED);
}
```

Fuente: Creación Propia

ServiceImpl

Figura 33 Service API registrar categoría

```
4 usages
@Override
public CategoriaDTO save(CategoriaDTO categoriaDTO) {
    var categoriaEntity : CategoriaEntity = categoriaMapper.postDtoToEntity(categoriaDTO);
    categoriaEntity.setSubCategoriaTitulo(null);
    var save : CategoriaEntity = this.categoriaRepository.save(categoriaEntity);
    categoriaDTO.getSubCategoriaTitulo().forEach(subCategoriaTituloDTO -> {
        CategoriaDTO categoria = new CategoriaDTO();
        categoria.setId(save.getId());
        subCategoriaTituloDTO.setCategoria(categoria);
    });
    this.subCategoriaTituloRepository.saveAll(categoriaMapper.postDtoToEntity(categoriaDTO).getSubCategoriaTitulo());

    return categoriaMapper.entityToGetDto(save);
}
```

Fuente: Creación Propia

URL: {{msvc-gateway}}/msvc-optica-gestion-productos/v1/categorias

Descripción: Permite la creación de categorías del producto

Figura 34 Prueba del endpoint registrar categoría

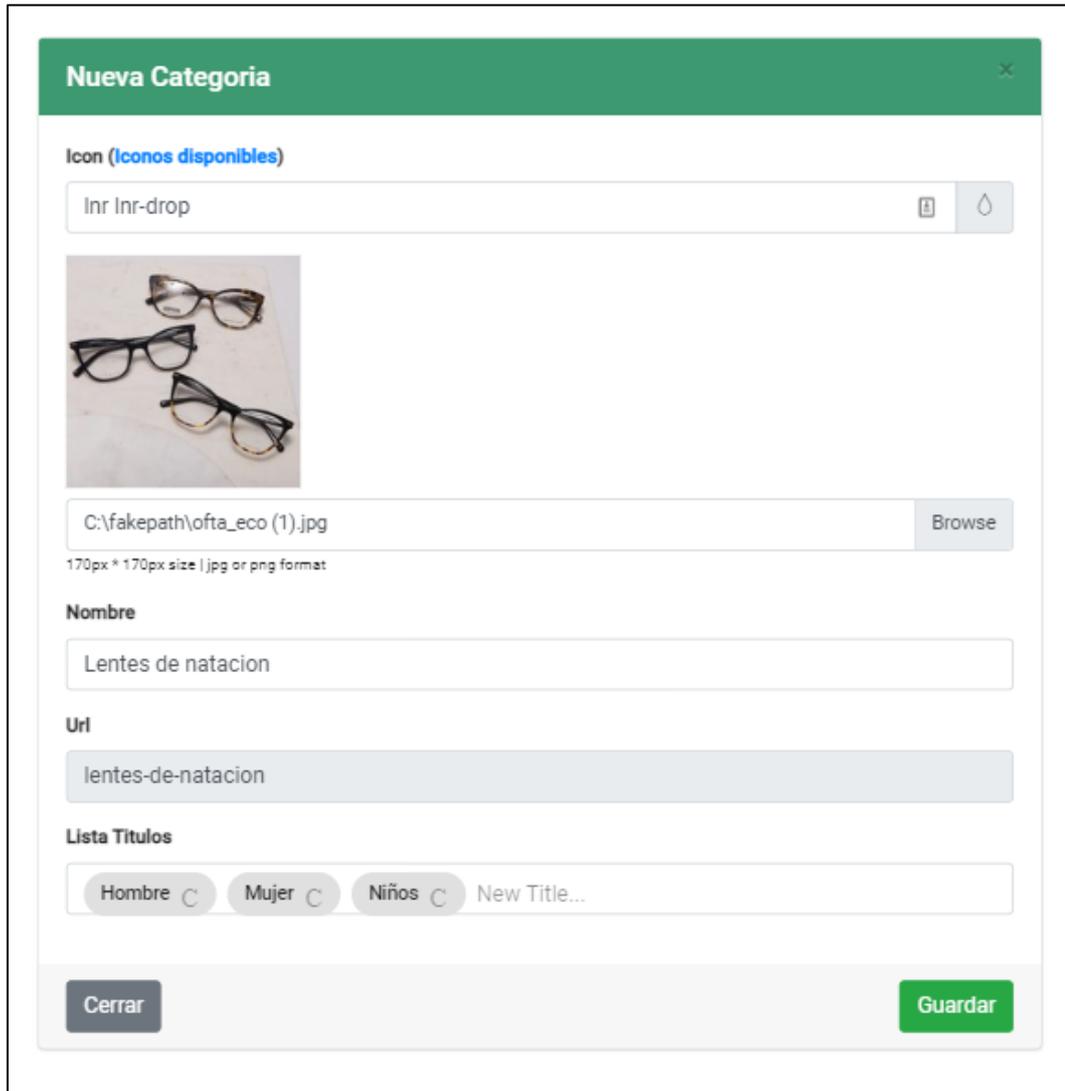
```
POST http://localhost:8887/msvc-optica-gestion-productos/v1/categorias
Body
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
[
  {
    "estado": 1,
    "fechaCreacion": "2022-11-10 23:56:52",
    "usuarioCreacion": "18db3141-3d1f-11ed-b973-025041000001",
    "fechaModificacion": "2022-11-11 18:19:06",
    "usuarioModificacion": null,
    "id": "e07e1b7f-6174-11ed-b9ef-025041000001",
    "nombre": "Accesorios",
    "url": "accesorios",
    "imagen": "v1668208747/categories/accesorios.jpg",
    "icono": "icon-desktop",
    "vista": 45,
    "subCategoriaTitulo": [
      {
        "estado": 1,
        "fechaCreacion": "2022-11-11 00:27:21",
        "usuarioCreacion": "18db3141-3d1f-11ed-b973-025041000001",
        "fechaModificacion": null,
        "usuarioModificacion": null,
        "id": "22869ec0-6179-11ed-b9ef-025041000001",
        "categoria": null,
        "descripcion": "Accesorios"
      }
    ]
  }
]
```

Fuente: Creación Propia

✓ **Pruebas funcionales**

Para la prueba de la funcionalidad se debe llenar todos los campos obligatorios

Figura 35 Prueba funcional de registro de categoría



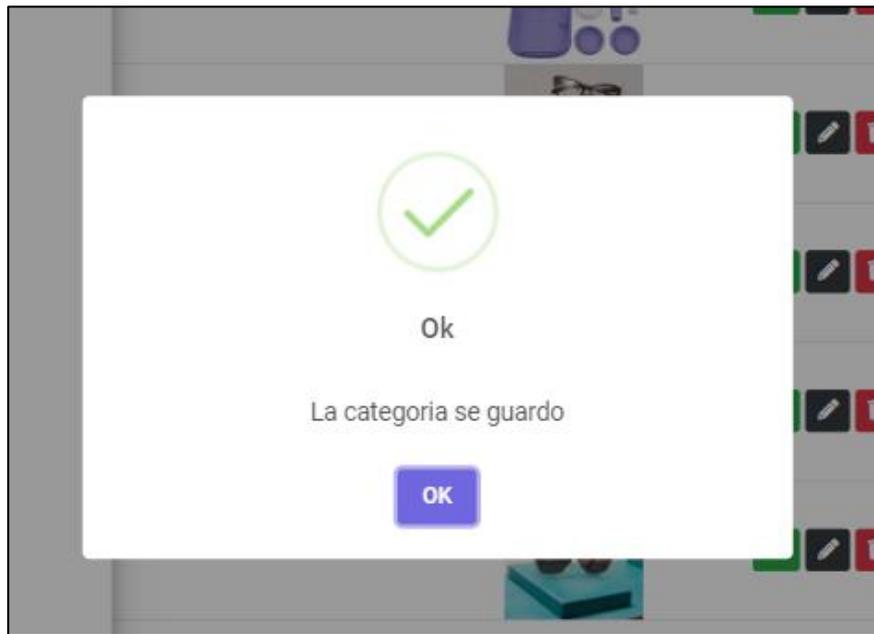
The image shows a web form titled "Nueva Categoría" (New Category) with a green header. The form contains the following fields and elements:

- Icon (Iconos disponibles):** A text input field containing "Inr Inr-drop" with a trash icon and a refresh icon to its right.
- Image:** A square image showing three pairs of eyeglasses.
- Image Path:** A text input field containing "C:\fakepath\ofta_eco (1).jpg" with a "Browse" button to its right.
- Dimensions:** Text below the image path indicating "170px * 170px size | jpg or png format".
- Nombre:** A text input field containing "Lentes de natacion".
- Url:** A text input field containing "lentes-de-natacion".
- Lista Titulos:** A horizontal list of buttons: "Hombre C", "Mujer C", "Niños C", and "New Title...".
- Buttons:** At the bottom, there are two buttons: "Cerrar" (Close) on the left and "Guardar" (Save) on the right.

Fuente: Creación Propia

Cuando se registre correctamente se mostrará el siguiente mensaje de confirmación

Figura 36 Mensaje de confirmación de registro de categorías



Fuente: Creación Propia

Luego de que termine la confirmación se redirigirá a la lista de categorías

Figura 37 Listado de categorías

The screenshot shows a web page titled "Categorías". At the top left, there is a green button labeled "Nueva Categoría". At the top right, there is a search bar with the placeholder text "Buscar". Below these elements is a table with the following columns: "No. ↑", "Nombre", "Url", "Imagen", and "Acciones". The table contains three rows of data:

No. ↑	Nombre	Url	Imagen	Acciones
6	Monturas o Gafas	monturas-o-gafas		
7	Lentes de Contacto	lentes-de-contacto		
8	lentes de natacion	lentes-de-natacion		

At the bottom right of the page, there is a pagination control showing "Items per page: 5" and "6 - 8 of 8" with navigation arrows.

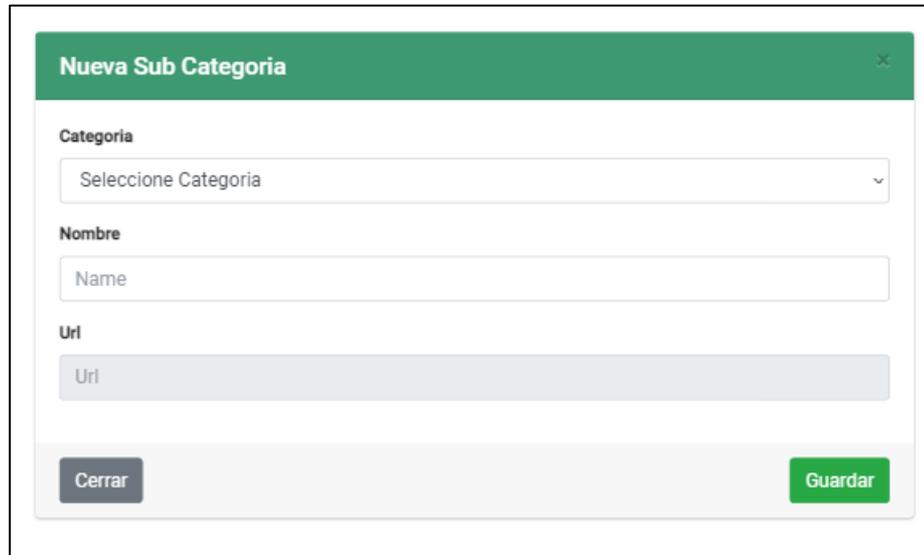
Fuente: Creación Propia

3) REGISTRO DE SUBCATEGORÍAS

✓ Desarrollo UI

Esta opción solo estará disponible para el cliente.

Figura 38 UI subcategoría



Fuente: Creación Propia

✓ Desarrollo del API

A continuación, se muestra el desarrollo del API para el registro de categorías del producto.

Controller

Figura 39 Controller API registrar subcategoría

```
@PostMapping
public ResponseEntity<CustomResponse> save(@Valid @RequestBody SubCategoriaDTO subCategoriaDTO){
    SubCategoriaDTO categoria = this.subCategoriaService.save(subCategoriaDTO);
    CustomResponse r = new CustomResponse(String.valueOf(CodeEnum.SUCCESS), categoria,
        message: "Categoria registrado correctamente.");
    return new ResponseEntity<>(r, HttpStatus.CREATED);
}
```

Fuente: Creación Propia

ServiceImpl

Figura 40 Service API registrar subcategoría

```
4 usages
@Override
public SubCategoriaDTO save(SubCategoriaDTO subCategoriaDTO) {
    SubCategoriaEntity subCategoriaEntity = subCategoriaMapper.postDtoToEntity(subCategoriaDTO);
    var save : SubCategoriaEntity = this.subCategoriaRepository.save(subCategoriaEntity);
    return subCategoriaMapper.entityToGetDto(save);
}
```

Fuente: Creación Propia

URL: `{{msvc-gateway}}/msvc-optica-gestion-productos/v1/sub-categorias`

Descripción: Permite la creación de las subcategorías del producto

Figura 41 Prueba del endpoint registrar subcategoría

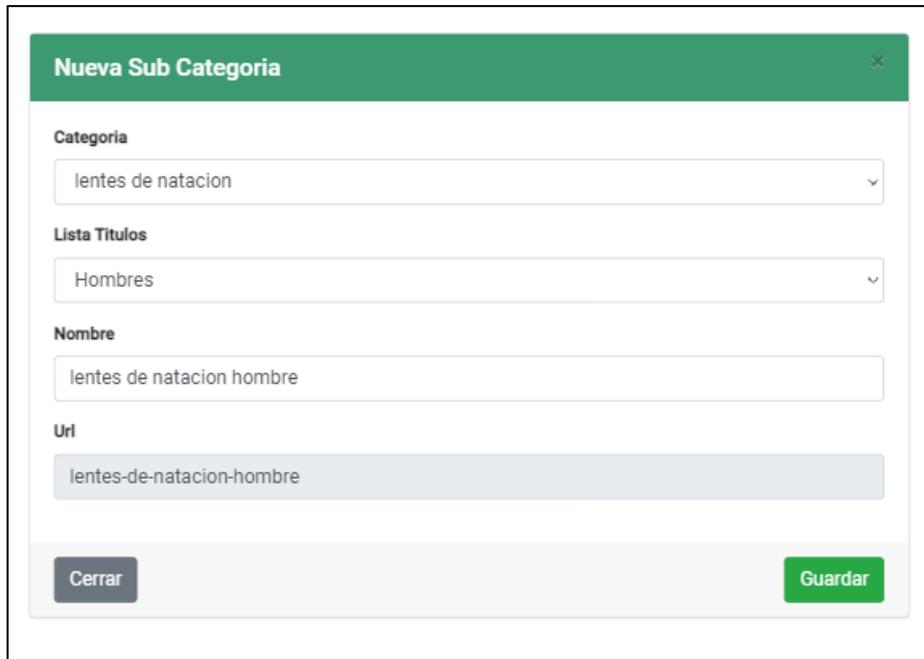
```
POST {{msvc-gateway}}/msvc-optica-gestion-productos/v1/sub-categorias
Body
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 [
2   {
3     "estado": 1,
4     "fechaCreacion": "2022-11-11 19:58:05",
5     "usuarioCreacion": "18db3141-3d1f-11ed-b973-025041000001",
6     "fechaModificacion": null,
7     "usuarioModificacion": null,
8     "id": "00cc3e30-cde2-43c2-acb1-b9454130d228",
9     "subCategoriaTitulos": {
10      "estado": 1,
11      "fechaCreacion": "2022-11-11 00:26:59",
12      "usuarioCreacion": "18db3141-3d1f-11ed-b973-025041000001",
13      "fechaModificacion": null,
14      "usuarioModificacion": null,
15      "id": "151b781a-6179-11ed-b9ef-025041000001",
16      "categoria": {
17        "estado": 1,
18        "fechaCreacion": "2022-11-10 23:56:36",
19        "usuarioCreacion": "18db3141-3d1f-11ed-b973-025041000001",
20        "fechaModificacion": "2022-11-11 18:50:26",
21        "usuarioModificacion": null,
22        "id": "d67f964d-6174-11ed-b9ef-025041000001",
23        "nombre": "Lentes de Contacto",
24        "url": "lentes-de-contacto",
25        "imagen": "v1668210627/categories/lentes-de-contacto.jpg",
26        "icono": "icon-laundry",
27        "vista": 2,
28        "subCategoriaTitulo": []
29      },
30      "descripcion": "Tipos"
31    },
32    "nombre": "Terapéuticos",
33    "url": "terapeuticos",
34    "imagen": "terapeuticos",
35    "inventario": "0",
36    "vista": 0
37  }
38 ]
```

Fuente: Creación Propia

✓ Pruebas funcionales

Para la prueba de la funcionalidad se debe llenar todos los campos obligatorios.

Figura 42 Prueba funcional registro subcategoría



El formulario 'Nueva Sub Categoría' contiene los siguientes campos:

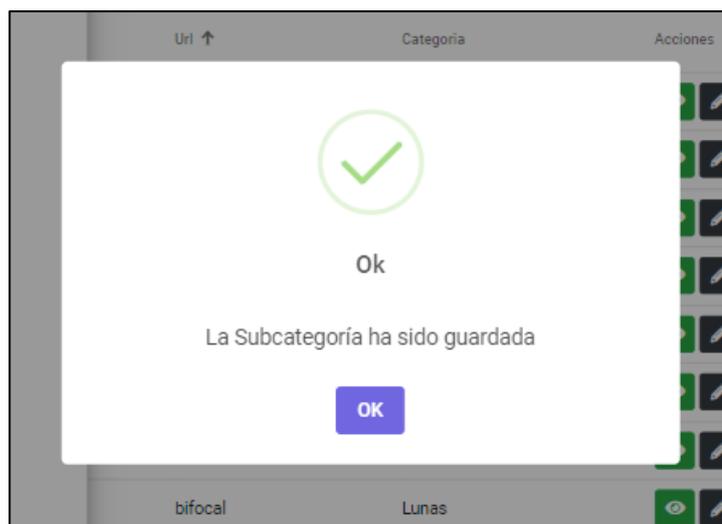
- Categoría:** Dropdown con 'lentes de natacion' seleccionado.
- Lista Titulos:** Dropdown con 'Hombres' seleccionado.
- Nombre:** Campo de texto con 'lentes de natacion hombre'.
- Uri:** Campo de texto con 'lentes-de-natacion-hombre'.

Botones: 'Cerrar' (gris) y 'Guardar' (verde).

Fuente: Creación Propia

Cuando se registre correctamente se mostrará el siguiente mensaje de confirmación

Figura 43 mensaje de confirmación de registro de subcategoría



Fuente: Creación Propia

Luego de que termine la confirmación se redirigirá a la lista de categorías

Figura 44 listado de subcategorías

Fuente: Creación Propia

○ **Revisión del sprint**

Tabla 35 Revisión del sprint

N°	ACTIVIDADES	DURACIÓN (#DÍAS)	ESTADO	OBSERVACIONES	DURACIÓN FINAL (#DÍAS)
1	Reunión de planificación	1	Completado	Ninguna	1
2	Desarrollar UI	2	Completado	Ninguna	2
3	Desarrollo del API	1	Completado	Ninguna	1
4	Pruebas funcionales	1	Completado	Ninguna	1
5	Reunión de planificación	1	Completado	Ninguna	1
6	Desarrollar UI	2	Completado	Ninguna	2
7	Desarrollo del API	1	Completado	Ninguna	1
8	Pruebas Unitarias	1	Completado	Ninguna	1
9	Pruebas funcionales	1	Completado	Ninguna	1
10	Reunión de planificación	1	Completado	Ninguna	1
11	Desarrollar UI	2	Completado	Ninguna	2
12	Desarrollo del API	1	Completado	Ninguna	1
13	Pruebas Unitarias	1	Completado	Ninguna	1
14	Pruebas funcionales	1	Completado	Ninguna	1

Fuente: Creación Propia

Sprint 2

Tabla 36 Sprint 2

N° SPRINT	HISTORIA DE USUARIO	ACTIVIDADES	ESTIMACIÓN (#DÍAS)	TOTAL (#DÍAS)
2		Reunión de planificación	1	12
	Visualizar categoría de productos	Desarrollar UI	2	
		Desarrollo del API	1	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
		Reunión de planificación	1	
	Visualizar productos	Desarrollar UI	2	
		Desarrollo del API	1	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
		Reunión de planificación	1	
	Visualizar precios	Desarrollar UI	2	
		Desarrollo del API	1	
		Pruebas Unitarias	1	
		Pruebas funcionales	1	
		Reunión de planificación	1	
Registrar visitas al sistema	Desarrollar UI	2		
	Desarrollo del API	1		
	Pruebas Unitarias	1		
	Pruebas funcionales	1		

Fuente: Creación Propia

- **Reunión de planificación**

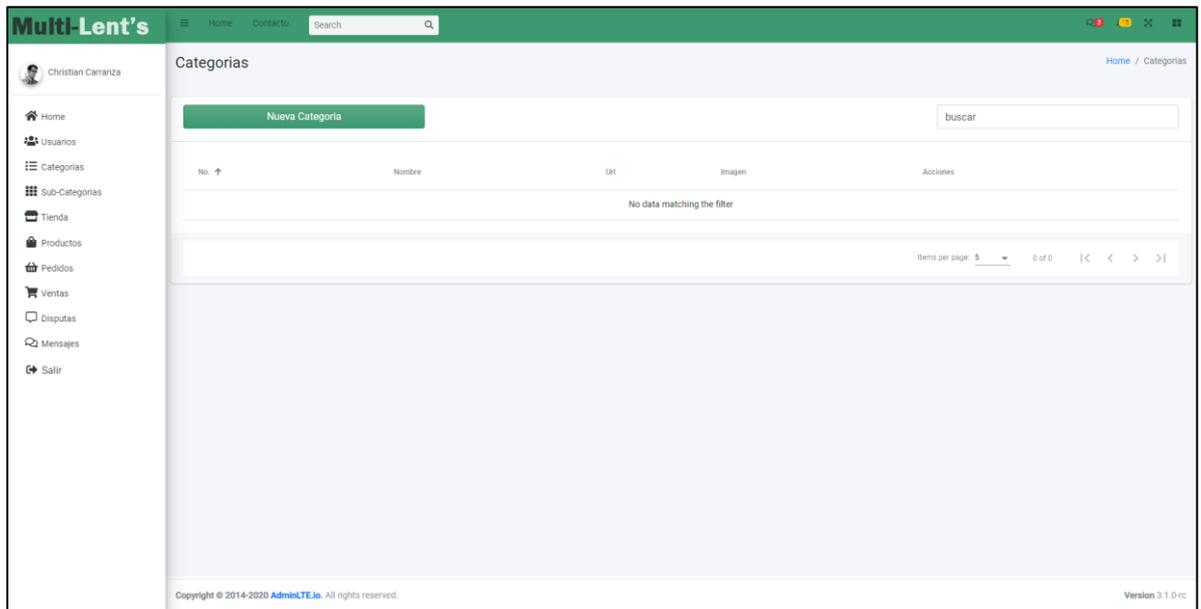
Se llevo a cabo la reunión con todo el equipo para el desarrollo del Sprint 2, a fin de planificar todas las actividades para completar los entregables.

4) VISUALIZAR CATEGORÍA DE PRODUCTOS

- ✓ **Desarrollo UI**

La vista dependerá según la página que se visite ya que existe un listado de categorías para el administrador y otro para los clientes.

Figura 45 UI Visualizar categoría de productos



Fuente: Creación Propia

✓ Desarrollo del API

A continuación, se muestra el desarrollo del API para el listado de categorías del producto.

Controller

Figura 46 Controller API listado de categorías

```
@Slf4j
@RestController
@RequestMapping(ConstantsUtil.API_CATEGORIA)
public class CategoriaController {
    @ usages
    private final CategoriaService categoriaService;

    public CategoriaController(CategoriaService categoriaService) { this.categoriaService = categoriaService; }

    @GetMapping("/{findAllPage}")
    @ResponseBody
    public ResponseEntity<CustomResponse> findAllPage(Pageable paginador){
        Page<CategoriaDTO> lstCategoriaDTO= this.categoriaService.findAllPage(paginador);
        if (lstCategoriaDTO.isEmpty()) {
            return new ResponseEntity<>(HttpStatus.NO_CONTENT);
        }
        CustomResponse rpta = new CustomResponse(String.valueOf(CodeEnum.SUCCESS),
            lstCategoriaDTO, message: "Información encontrada");
        return new ResponseEntity<>(rpta, HttpStatus.OK);
    }
}
```

Fuente: Creación Propia

Repository

Figura 47 Repository API listado de categorías

```
3 usages
@Repository
public interface CategoriaRepository extends JpaRepository<CategoriaEntity, UUID> {

    @Query("select c from CategoriaEntity c where UPPER(c.nombre) like UPPER(:nombre) ORDER BY c.nombre")
    Optional<CategoriaEntity> findByLikeNombre(@Param("nombre") String nombre);

    @Query("select c from CategoriaEntity c where c.estado=1")
    Page<CategoriaEntity> findAllPage(Pageable pageable);
}
```

Fuente: Creación Propia

ServiceImpl

Figura 48 Service API listado de categorías

```
@Override
public Page<CategoriaDTO> findAllPage(Pageable paginador) {
    Page<CategoriaEntity> lstCategoriaEntity = categoriaRepository.findAllPage(paginador);
    List<CategoriaDTO> result = lstCategoriaEntity.stream().map(categoriaMapper::entityToGetDto).collect(Collectors.toList());

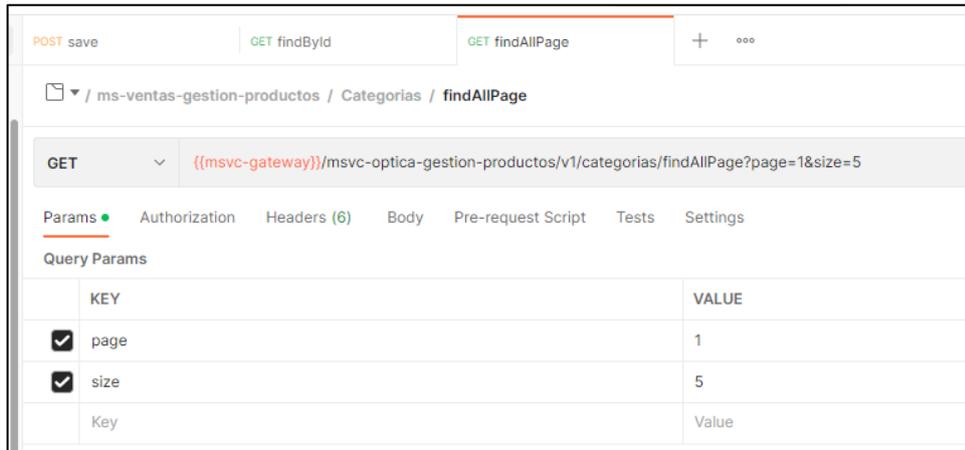
    return (Page<CategoriaDTO>) FechasUtil.paginate(result, paginador, lstCategoriaEntity.getTotalElements());
}
```

Fuente: Creación Propia

URL: `{{msvc-gateway}}/msvc-optica-gestion-productos/v1/categorias/findAllPage?page=1&size=5`

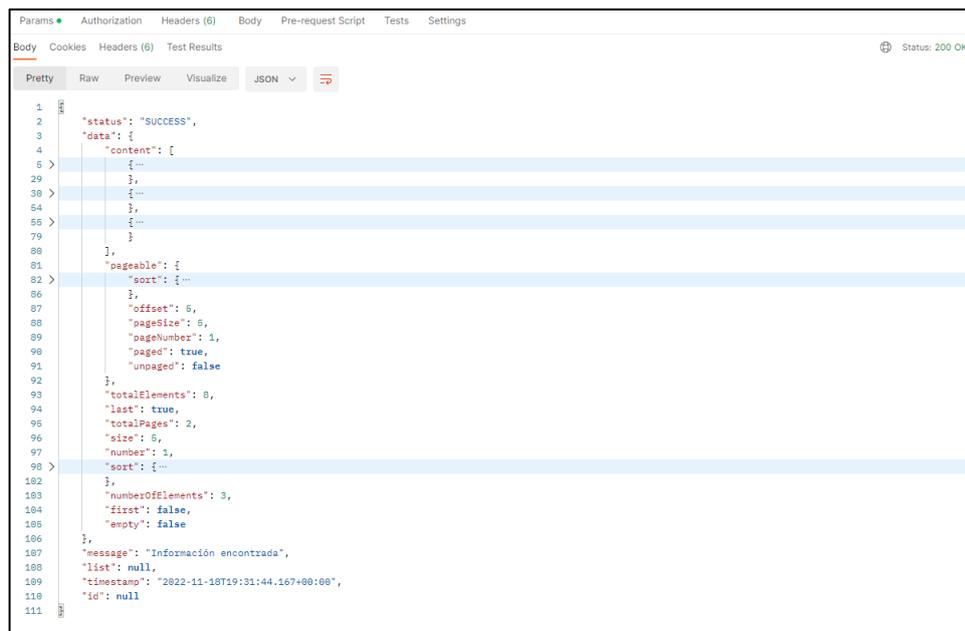
Descripción: Permite hacer un listado de categorías del producto con paginación

Figura 49 Envío de parámetros para el endpoint Visualizar categoría de productos



Fuente: Creación Propia

Figura 50 Prueba del endpoint Visualizar categoría de productos



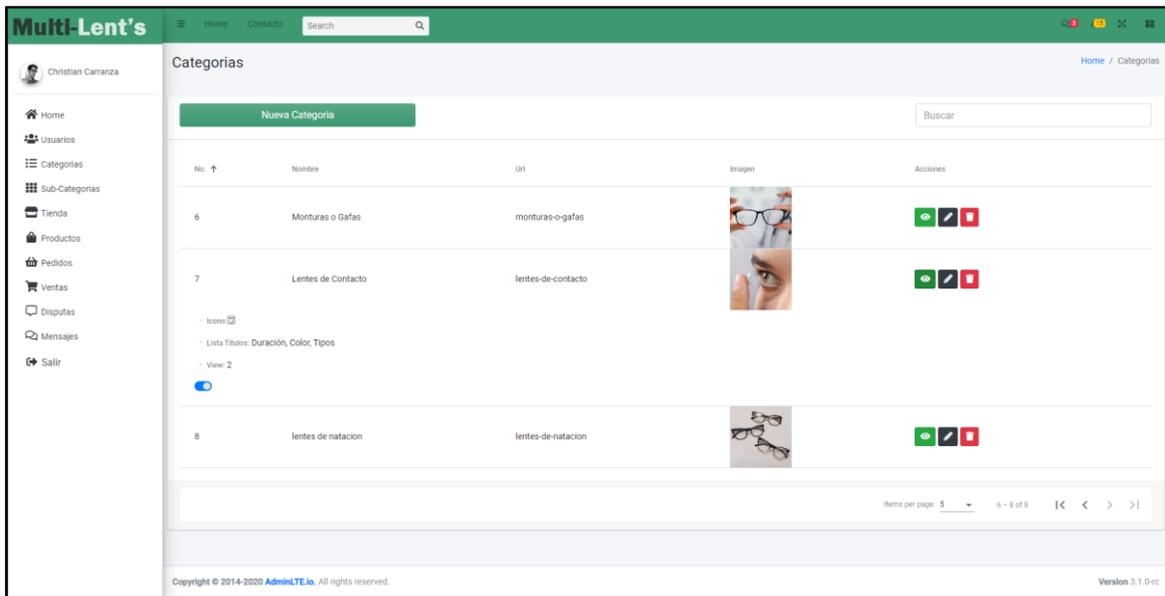
Fuente: Creación Propia

✓ Pruebas funcionales

Para la prueba de la funcionalidad se realizó un listado de todas las categorías divididas en varias páginas.

Administrador

Figura 51 Prueba funcional listado de categorías - administrador



Fuente: Creación Propia

Cliente

Figura 52 Prueba funcional listado de categorías - cliente



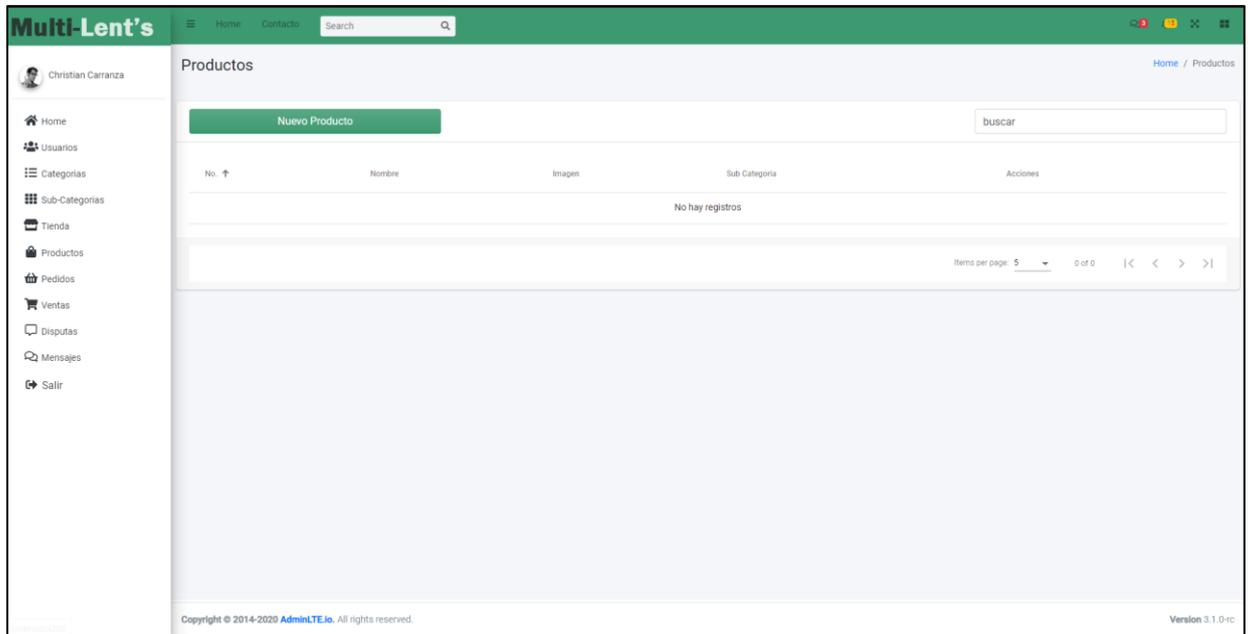
Fuente: Creación Propia

5) VISUALIZAR PRODUCTOS

✓ Desarrollo UI

La vista dependerá según la página que se visite ya que existe un listado de categorías para el administrador y otro para los clientes.

Figura 53 Prueba funcional listado de producto



Fuente: Creación Propia

✓ Desarrollo del API

A continuación, se muestra el desarrollo del API para el registro de categorías del producto.

Controller

Figura 54 Controller API listado de productos

```
@Slf4j
@RestController
@RequestMapping(ConstantsUtil.API_PRODUCTO)
public class ProductoController {
    10 usages
    private final ProductoService productoService;

    public ProductoController(ProductoService productoService) { this.productoService = productoService; }

    @GetMapping("/findAllPage")
    @ResponseBody
    public ResponseEntity<CustomResponse> findAllPage(Pageable paginador){
        Page<ProductoDTO> lstProductoDTO= this.productoService.findAllPage(paginador);
        if (lstProductoDTO.isEmpty()) {
            return new ResponseEntity<>(HttpStatus.NO_CONTENT);
        }
        CustomResponse rpta = new CustomResponse(String.valueOf(CodeEnum.SUCCESS),
            lstProductoDTO, message: "Información encontrada");
        return new ResponseEntity<>(rpta, HttpStatus.OK);
    }
}
```

Fuente: Creación Propia

ServiceImpl

Figura 55 Service API listado de productos

```
@Override
public Page<ProductoDTO> findAllPage(Pageable paginador) {
    Page<ProductoEntity> lstProductoEntity = ProductoRepository.findAllPage(paginador);
    List<ProductoDTO> result = lstProductoEntity.stream().map(productoMapper::entityToGetDto).collect(Collectors.toList());

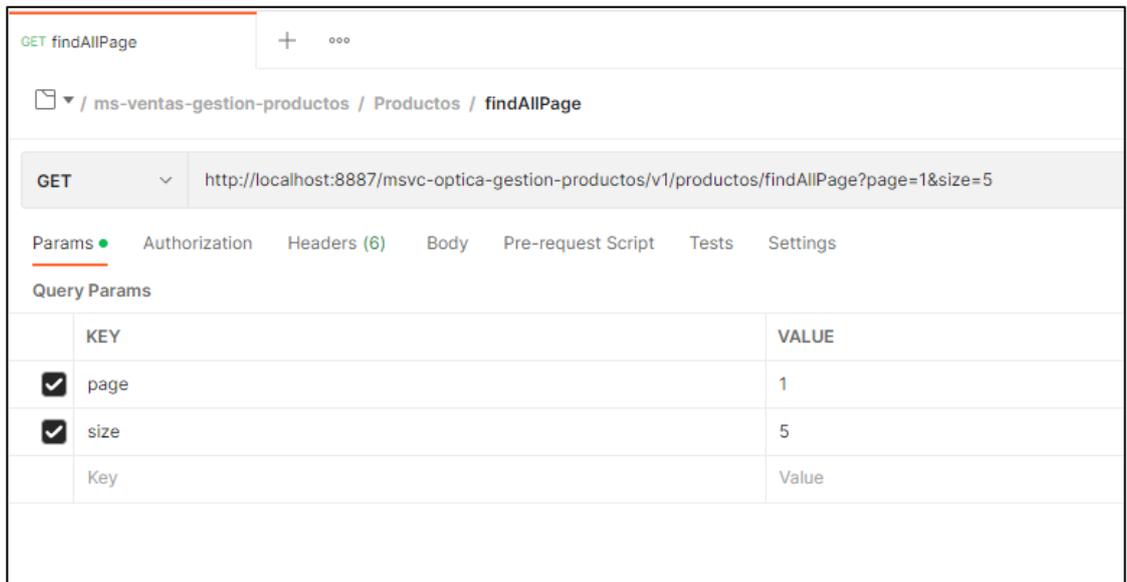
    return (Page<ProductoDTO>) FechasUtil.paginate(result, paginador, lstProductoEntity.getTotalElements());
}
```

Fuente: Creación Propia

URL: `{{msvc-gateway}}/msvc-optica-gestion-productos/v1/productos/findAllPage?page=1&size=5`

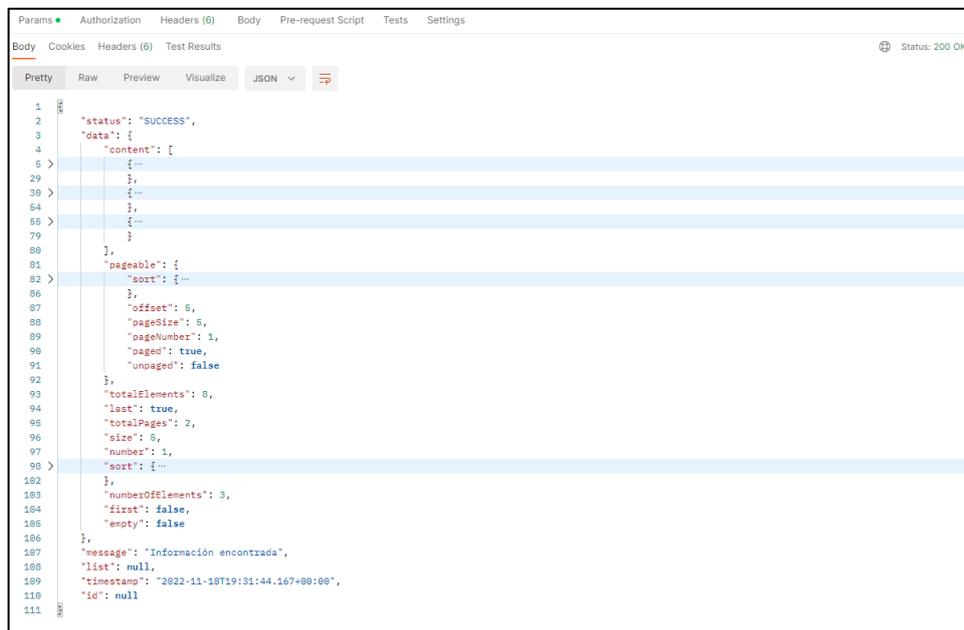
Descripción: Permite hacer un listado de productos con paginación

Figura 56 Envío de parámetros para el endpoint Visualizar productos



Fuente: Creación Propia

Figura 57 Prueba del endpoint Visualizar productos

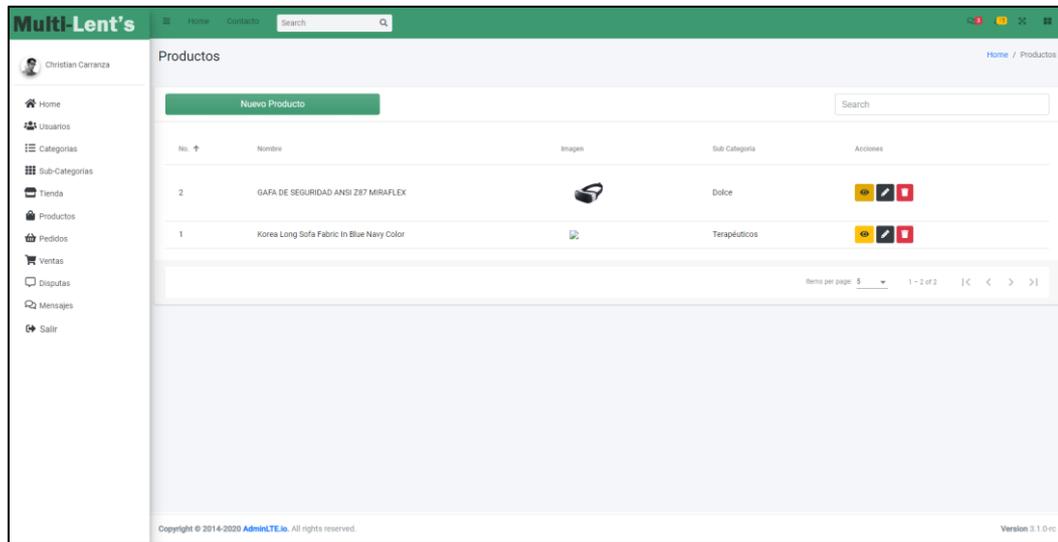


Fuente: Creación Propia

✓ **Pruebas funcionales**

Para la prueba de la funcionalidad se realizó un listado de todos los productos divididos en varias páginas.

Figura 58 Prueba funcional listado de productos



Fuente: Creación Propia

○ **Revisión del sprint**

Tabla 37 Revisión del sprint

N°	ACTIVIDADES	DURACIÓN (#DÍAS)	ESTADO	OBSERVACIONES	DURACIÓN FINAL (#DÍAS)
1	Reunión de planificación	1	Completado	Ninguna	1
2	Desarrollar UI	2	Completado	Ninguna	2
3	Desarrollo del API	1	Completado	Ninguna	1
4	Pruebas Unitarias	1	Completado	Ninguna	1
5	Pruebas funcionales	1	Completado	Ninguna	1
6	Reunión de planificación	1	Completado	Ninguna	1
7	Desarrollar UI	2	Completado	Ninguna	2
8	Desarrollo del API	1	Completado	Ninguna	1
9	Pruebas Unitarias	1	Completado	Ninguna	1
10	Pruebas funcionales	1	Completado	Ninguna	1

11	Reunión de planificación	1	Completado	Ninguna	1
12	Desarrollar UI	2	Completado	Ninguna	2
13	Desarrollo del API	1	Completado	Ninguna	1
14	Pruebas Unitarias	1	Completado	Ninguna	1
15	Pruebas funcionales	1	Completado	Ninguna	1
16	Reunión de planificación	1	Completado	Ninguna	1
17	Desarrollar UI	2	Completado	Ninguna	2
18	Desarrollo del API	1	Completado	Ninguna	1
19	Pruebas Unitarias	1	Completado	Ninguna	1
20	Pruebas funcionales	1	Completado	Ninguna	1

Fuente: Creación Propia



UNIVERSIDAD CÉSAR VALLEJO

**FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

Declaratoria de Autenticidad del Asesor

Yo, NECOCHEA CHAMORRO JORGE ISAAC, 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: "Arquitectura de microservicios para el proceso de ventas de la óptica Multi-len's, 2022.", cuyo autor es CARRANZA FELIX CHRISTIAN BERNARDO, constato que la investigación tiene un índice de similitud de 19.00%, 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, 29 de Noviembre del 2022

Apellidos y Nombres del Asesor:	Firma
NECOCHEA CHAMORRO JORGE ISAAC DNI: 18167347 ORCID: 0000-0002-3290-8975	Firmado electrónicamente por: JNECOCHEA el 29- 11-2022 16:56:22

Código documento Trilce: TRI - 0461419